Ramakrishna Mission
Vivekananda Educational and Research Institute
(Deemed-to-be-University as declared by Govt. of India under Section 3 of UGC Act, 1956)
(Formerly Ramakrishna Mission Vivekananda University)
(Accredited by NAAC with A++ Grade)

CS411: Applications of Computer Vision and Deep Learning
Assignment: Multimodal Interaction

Designed by: Seshadri Mazumder[seshadrimazumder1997@gmail.com] & **ChatGPT**

Start Date: 7th April - End Date: 17th April(11:59 pm)

Note: No Extensions on any Grounds, Start Early to avoid Last moment Rush

Title: Image Retrieval using Vision Transformer (ViT) and Transformer Encoder for Text

Objective: The objective of this assignment is to implement and train a Vision Transformer (ViT) based Transformer encoder for image embeddings and a Transformer encoder for text embeddings using the MSCOCO dataset. Subsequently, the trained models will be utilized for image retrieval tasks using the Transformer architecture.

The aim of this assignment is to develop proficiency in leveraging online resources, particularly blog posts, for learning and adapting existing code to suit one's coding abilities. The goal is to efficiently train models and complete projects within tight deadlines. While utilizing blog resources and ChatGPT for coding assistance is encouraged, it is essential to thoroughly understand and explain every aspect of the code. Additionally, the assignment emphasizes the avoidance of using nn.Transformer module in any capacity.

Assignment Description:

Task 1: Implementation of Vision Transformer (ViT) and Transformer Encoder for Text (40 marks)

1. Implement the Vision Transformer (ViT) architecture for image feature extraction.
2. Implement the Transformer encoder architecture for text feature extraction.
3. Train the ViT and Transformer encoder on the MSCOCO dataset for contrastive learning using a CLIP-style training objective.

Task 2: Image Retrieval using Transformer (30 marks)

1. Utilize the trained ViT and Transformer encoder models to encode images and text descriptions, respectively.
2. Implement an image retrieval mechanism using the encoded image and text embeddings.
3. Evaluate the image retrieval performance using relevant metrics such as retrieval accuracy or mean average precision (mAP).

Task 3: Analysis and Interpretation (30 marks)

1. Analyze the performance of the image retrieval system based on Transformer embeddings.
2. Discuss the impact of different architectural choices and training strategies on the retrieval performance.
3. Compare the performance of the Transformer-based image retrieval system with other baseline methods.
4. Provide insights into the strengths and weaknesses of utilizing Transformers for image retrieval tasks.

Submission Guidelines:

1. Submit well-documented code implementing the ViT, Transformer encoder, and image retrieval mechanism.
2. Provide a detailed report including implementation details, training procedures, evaluation results, and analysis.
3. Include any additional resources or references used in your implementation.
4. Prepare a 10 minutes Presentation for your work

Grading Rubric:

- Task 1: Implementation of ViT and Transformer Encoder - 40 marks
- Task 2: Image Retrieval using Transformer - 30 marks
- Task 3: Analysis and Interpretation - 30 marks
- Task 4: Report - 30 Marks [Overleaf Document, Properly structured in a single column format]
- Task 5: Presentation - 30 Marks [Presentation Created, Delivery Style, Timely Completion with a link to show code demo, within the last two minutes, Unique Addition Tried or Optimised, providing Clear References]

Total: 100 marks

Reference Papers:

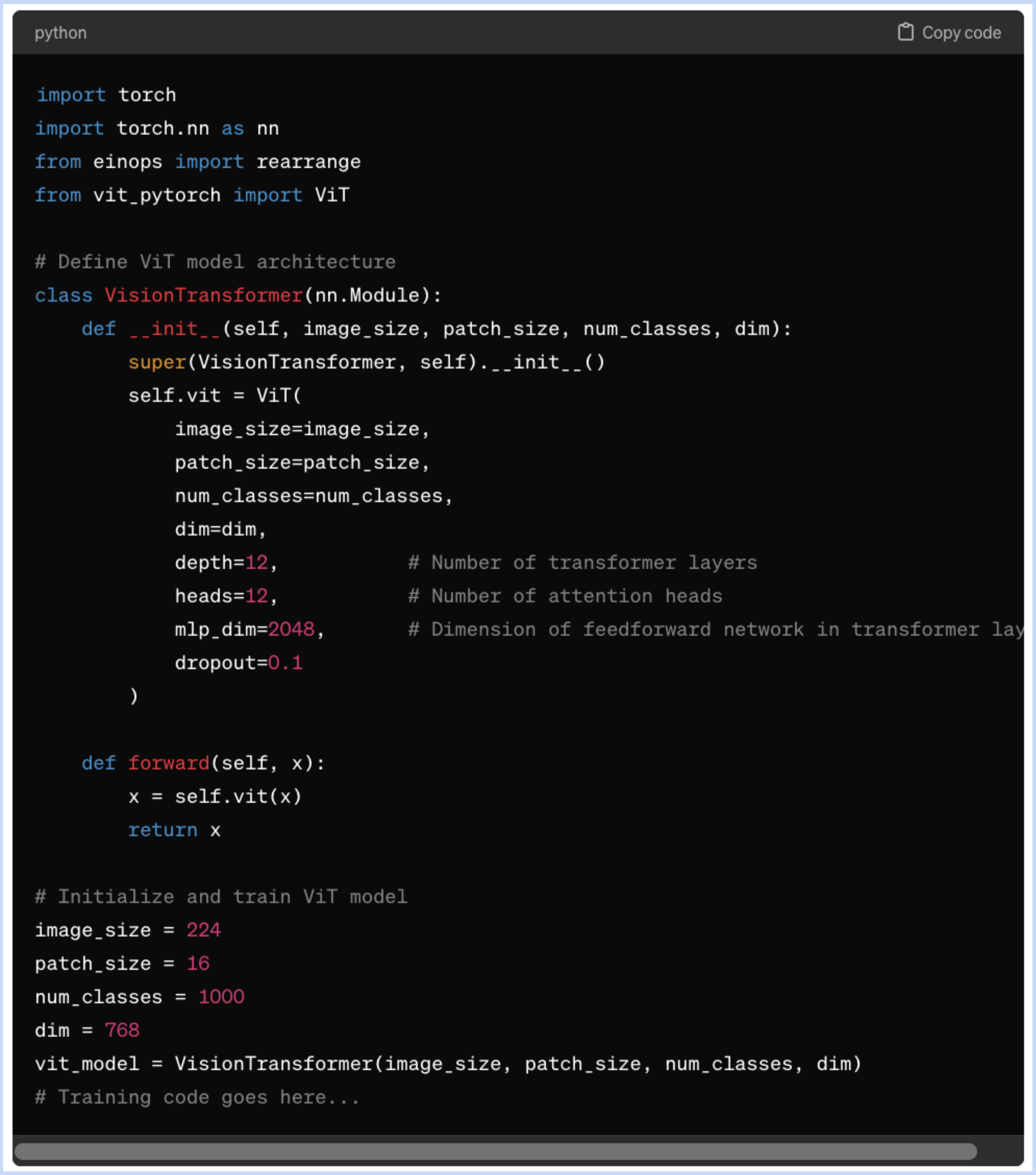
1. **Vision Transformer (ViT)**:
   - Title: "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale"
   - Authors: Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby
   - Link: arXiv:2010.11929
2. **CLIP: Contrastive Language-Image Pre-training**:
   - Title: "Learning Transferable Visual Models From Natural Language Supervision"
   - Authors: Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever
   - Link: arXiv:2103.00020
3. **Image Retrieval**:
   - Title: "DeViSE: A Deep Visual-Semantic Embedding Model"

  - Authors: Andrea Frome, Greg S. Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov
  - Link: arXiv:1310.1531

Code Blocks:

Below are code blocks to provide a starting point for implementation. These blocks are simplified and may need modification and integration into a larger codebase:

1. **Vision Transformer (ViT)**:


python

Copy code

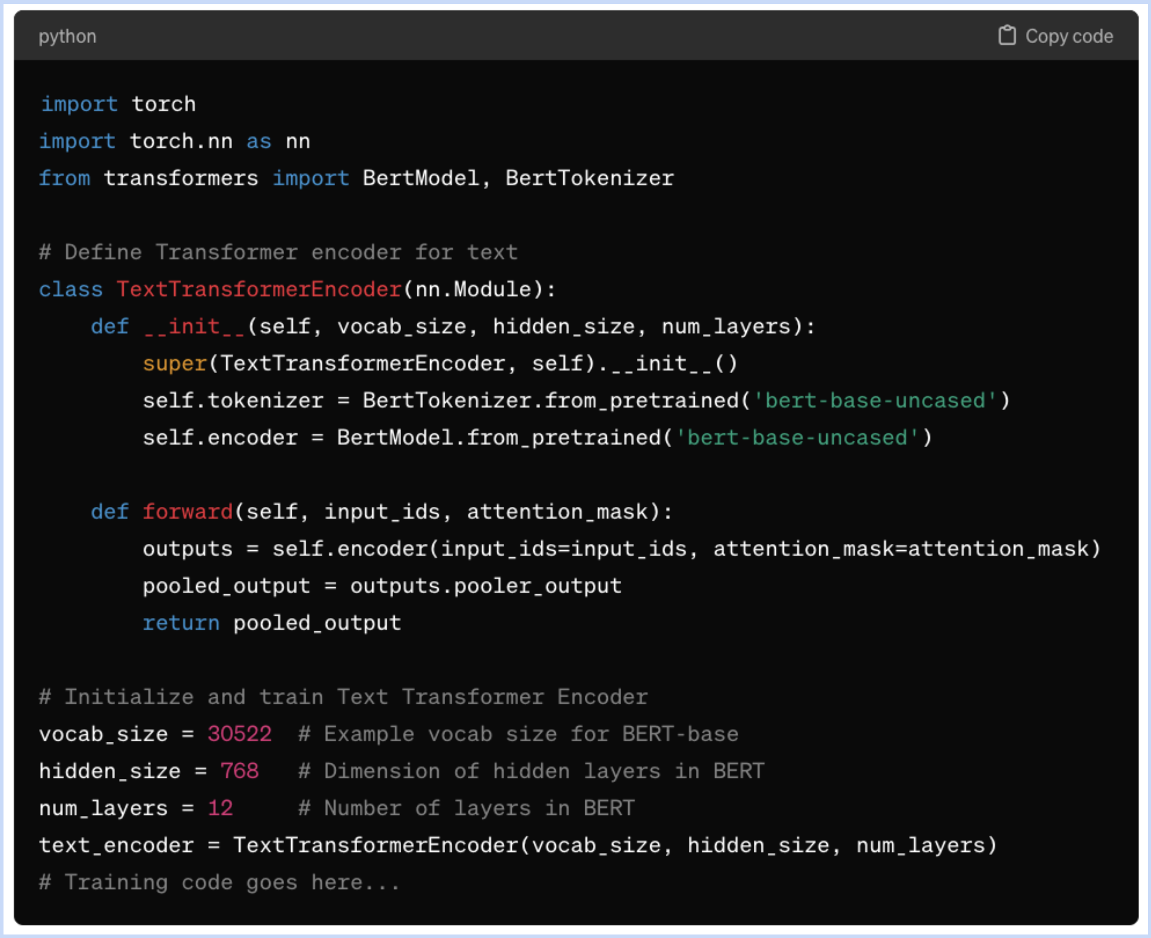
```python
import torch
import torch.nn as nn
from einops import rearrange
from vit_pytorch import ViT

# Define ViT model architecture
class VisionTransformer(nn.Module):
    def __init__(self, image_size, patch_size, num_classes, dim):
        super(VisionTransformer, self).__init__()
        self.vit = ViT(
            image_size=image_size,
            patch_size=patch_size,
            num_classes=num_classes,
            dim=dim,
            depth=12,           # Number of transformer layers
            heads=12,           # Number of attention heads
            mlp_dim=2048,       # Dimension of feedforward network in transformer lay
            dropout=0.1
        )

    def forward(self, x):
        x = self.vit(x)
        return x

# Initialize and train ViT model
image_size = 224
patch_size = 16
num_classes = 1000
dim = 768
vit_model = VisionTransformer(image_size, patch_size, num_classes, dim)
# Training code goes here...
```

2. **Text Transformer Encoder**:



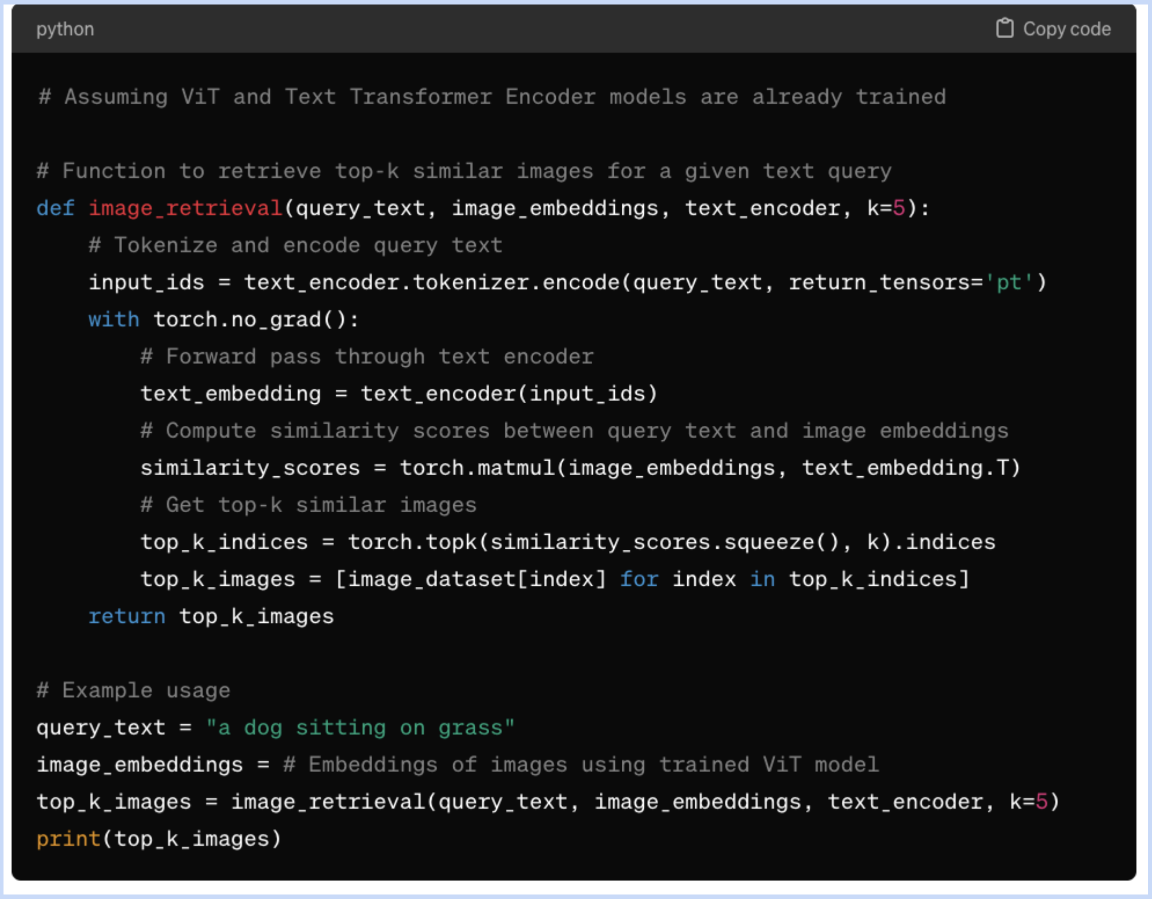
```python
import torch
import torch.nn as nn
from transformers import BertModel, BertTokenizer

# Define Transformer encoder for text
class TextTransformerEncoder(nn.Module):
    def __init__(self, vocab_size, hidden_size, num_layers):
        super(TextTransformerEncoder, self).__init__()
        self.tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
        self.encoder = BertModel.from_pretrained('bert-base-uncased')

    def forward(self, input_ids, attention_mask):
        outputs = self.encoder(input_ids=input_ids, attention_mask=attention_mask)
        pooled_output = outputs.pooler_output
        return pooled_output

# Initialize and train Text Transformer Encoder
vocab_size = 30522  # Example vocab size for BERT-base
hidden_size = 768   # Dimension of hidden layers in BERT
num_layers = 12     # Number of layers in BERT
text_encoder = TextTransformerEncoder(vocab_size, hidden_size, num_layers)
# Training code goes here...
```

3. **Image Retrieval using Transformer**:



```python
# Assuming ViT and Text Transformer Encoder models are already trained

# Function to retrieve top-k similar images for a given text query
def image_retrieval(query_text, image_embeddings, text_encoder, k=5):
    # Tokenize and encode query text
    input_ids = text_encoder.tokenizer.encode(query_text, return_tensors='pt')
    with torch.no_grad():
        # Forward pass through text encoder
        text_embedding = text_encoder(input_ids)
        # Compute similarity scores between query text and image embeddings
        similarity_scores = torch.matmul(image_embeddings, text_embedding.T)
        # Get top-k similar images
        top_k_indices = torch.topk(similarity_scores.squeeze(), k).indices
        top_k_images = [image_dataset[index] for index in top_k_indices]
    return top_k_images

# Example usage
query_text = "a dog sitting on grass"
image_embeddings = # Embeddings of images using trained ViT model
top_k_images = image_retrieval(query_text, image_embeddings, text_encoder, k=5)
print(top_k_images)
```

These code blocks provide a basic structure for implementing the Vision Transformer, Text Transformer Encoder, and image retrieval mechanism using Transformer embeddings. Additional preprocessing steps, training procedures, and evaluation metrics need to be added as required.

Best of Luck…!!

You guys are a wonderful audience…!!

7th April - 2024