Visual Grounding with Self-Supervised Vision Transformer

Hemran Akhtari (S5424801)
Kyriakos Antoniou (S5715881)

***Abstract*—Visual grounding, the task of localizing specific objects in an image based on natural language queries, has numerous real-world applications, such as human-robot interaction, augmented reality, and multimedia search. Traditional approaches rely on convolutional neural networks (CNNs) and require large amounts of labeled data. However, Vision Transformers (ViTs), pretrained on large-scale datasets, have shown superior performance in capturing global context and semantic features.**

This project aims to explore self-supervised pretrained ViTs (e.g., DINO or MAE) for visual grounding. By leveraging pretrained models and fine-tuning on visual grounding datasets (e.g., RefCOCO, RefCOCO+, Flickr30k Entities), we expect to build a system that can accurately localize objects based on text queries, without requiring extensive task-specific data for pretraining.

I. Introduction

Visual grounding refers to the task of identifying specific regions in an image corresponding to a natural language query. It has broad applications in fields like human-robot interaction, multimedia search, and augmented reality. The goal of this project is to explore how **self-supervised pretrained Vision Transformers (ViTs)** can be fine-tuned for visual grounding tasks.

While traditional methods rely heavily on CNN-based models and supervised learning, recent advances in ViTs and self-supervised learning open the door for more data-efficient models that can generalize better across tasks. By leveraging pretrained ViTs and fine-tuning them on datasets like RefCOCO and Flickr30k Entities, we aim to develop a model capable of accurate object localization based on textual descriptions.

II. Data

We will use the following datasets for fine-tuning and evaluation:

- **RefCOCO, RefCOCO+, RefCOCOg**: These datasets provide referring expressions (queries) and bounding box annotations for object localization in MS COCO images. The combined size of the datasets is approximately 140k referring expressions across 50k images.
- **Flickr30k Entities**: This dataset contains 30k images, each annotated with multiple phrases referring to different regions, along with bounding box annotations.
- **MS COCO Captions** (Optional): This dataset may be used for additional fine-tuning or contrastive pretraining if necessary.

III. Expected Results

The expected outcomes of the project are as follows:

- A fine-tuned ViT-based model capable of accurately localizing objects based on natural language queries.
- **Quantitative metrics**:
  - **Intersection over Union (IoU)**: This metric will measure the overlap between predicted and ground truth bounding boxes.
  - **Accuracy**: The accuracy of selecting the correct region will also be evaluated.
- **Qualitative results**:
  - Visualization of attention maps and bounding boxes to demonstrate model interpretability.
  - Comparative analysis of attention-based explanations and bounding box predictions.

IV. Time Plan

The project is structured into multiple phases to ensure steady progress:

Task	Duration	Milestone
Literature review	5 days	Understand ViTs, self-supervised learning, and visual grounding models.
Data preparation	3 days	Preprocess RefCOCO, Flickr30k datasets, and validate readiness.
Model setup	7 days	Implement image encoder (ViT), text encoder (BERT), and fusion module; test components.
Fine-tuning	7 days	Fine-tune the pretrained ViT on visual grounding datasets; iterate on hyperparameters.
Evaluation & Results	5 days	Evaluate the model using IoU and accuracy metrics; visualize attention maps.
Visualization & Report	3 days	Generate visualizations and compile the final report.

TABLE I
Project timeline with tasks, durations, and milestones. The project is expected to conclude on the 11th of February.

V. Expected Resources

A. Computational Resources

- **Google Colab Pro** for model fine-tuning.
- Two local machines will be used for initial code development and other tasks (one per student).

B. Tools

- **GitHub**: For efficient code collaboration and synchronization between the two local machines.
- **PyTorch**: For implementing the model and training.
- **Hugging Face Transformers**: For using pretrained BERT or other transformer-based text encoders.
- **torchvision**: For using pretrained ViT models and data augmentation.