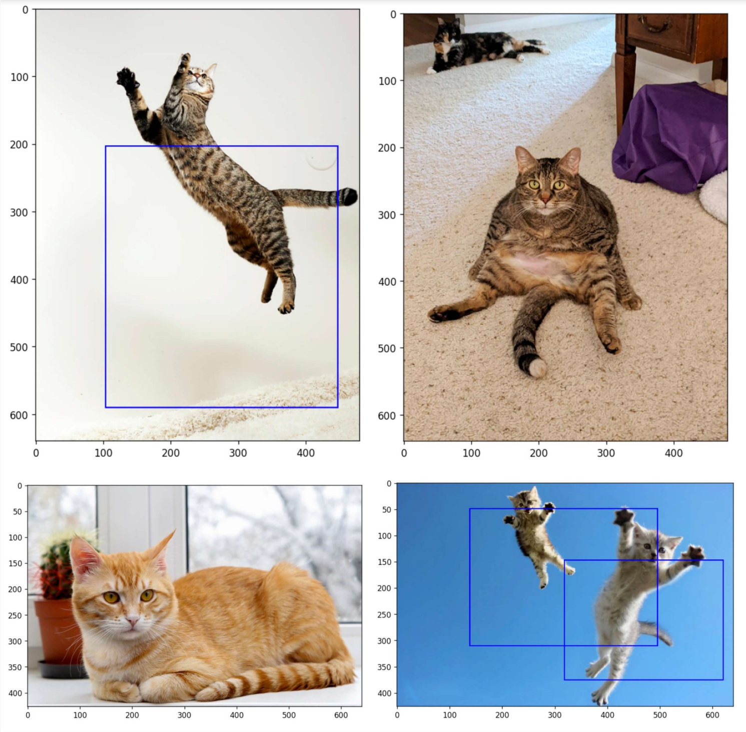
Zero-Shot Object Detection

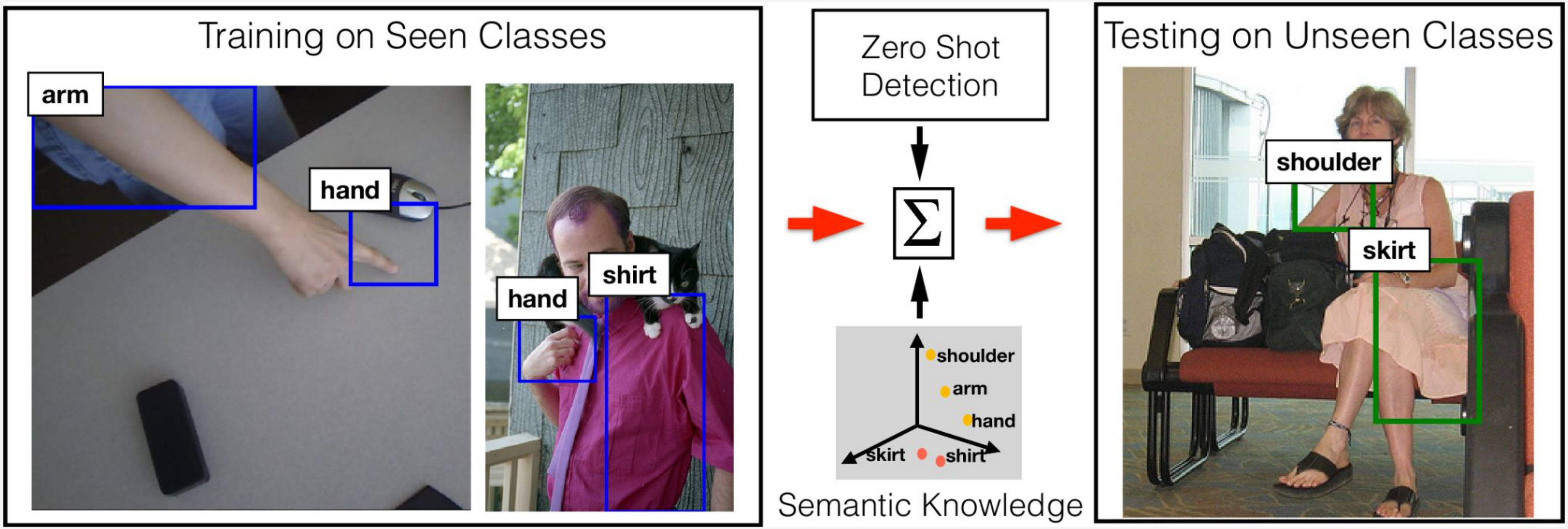
Team Members
Ananya Alekar, Tiya Gupta, Arya Gawde



What is Zero Shot Object Detection?

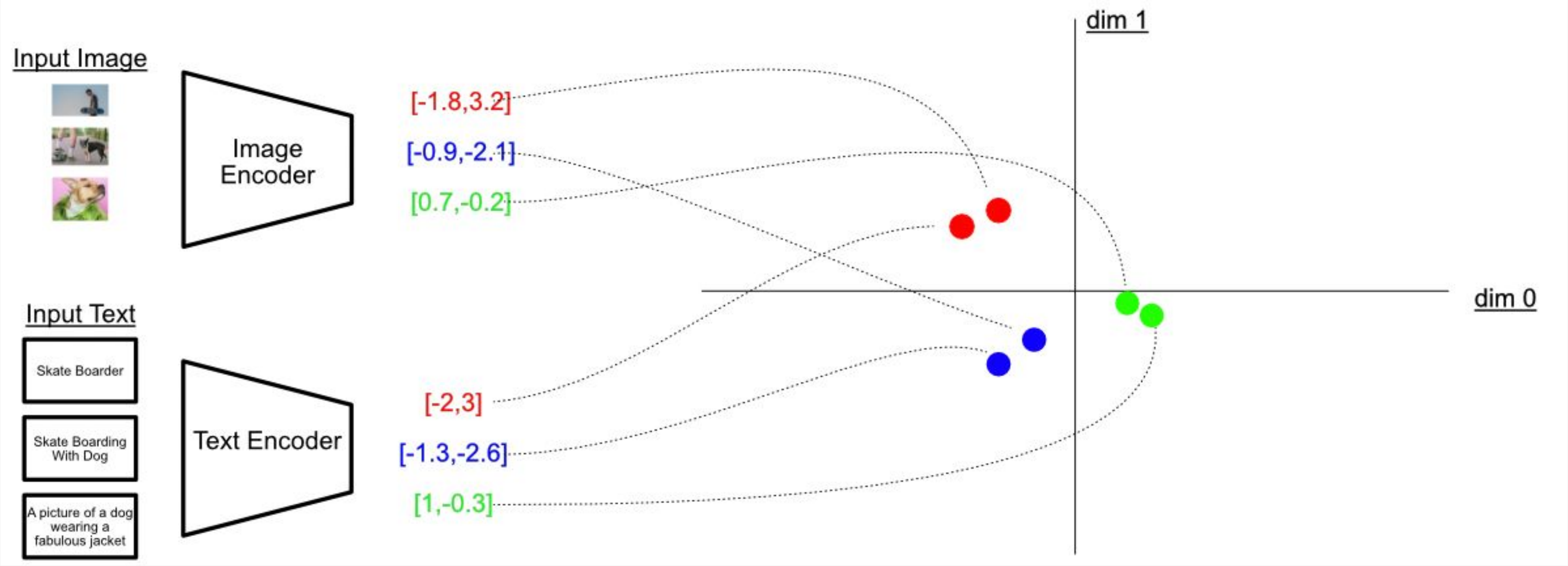
- **Zero-shot object detection**
  Localizes and classifies unseen objects by learning semantic embeddings
- The detector is not trained on the labeled images for new classes but it is able to perform the detection of any classes described in the text prompt list.
- Zero shot learning is considered a subset of transfer learning since a pre-trained model is used to predict unseen objects.
- This makes it easy to implement but it has few underlying issues.

Zero Shot Object Detection



What’s the issue?

- Common problems with this approach
    - Categorizing unseen objects as the background
    - Biasness problem
    - Hubness problem
- Our transformer based approach, a combination of DERT and CLIP aims to tackle these problems.
- Both of which are fairly recent transformer models which have proved to perform better than traditional ZSD methods.



Input Image
Image Encoder
[-1.8,3.2]
[-0.9,-2.1]
[0.7,-0.2]
Input Text
Skate Boarder
Skate Boarding With Dog
A picture of a dog wearing a fabulous jacket
Text Encoder
[-2,3]
[-1.3,-2.6]
[1,-0.3]
dim 1
dim 0

Our Model

- DETR (DEtection TRansformer): Detects objects in an image and classifies objects into predefined categories (e.g., COCO dataset classes).
- CLIP (Contrastive Language–Image Pretraining): Maps both images and text descriptions into a shared embedding space, enabling similarity comparisons between images and text in a zero-shot manner.
- The combination of these two models allows:
  - Object detection using DETR.
  - Matching detected objects to user-provided text descriptions using CLIP.

DERT (DEtection TRansformer)

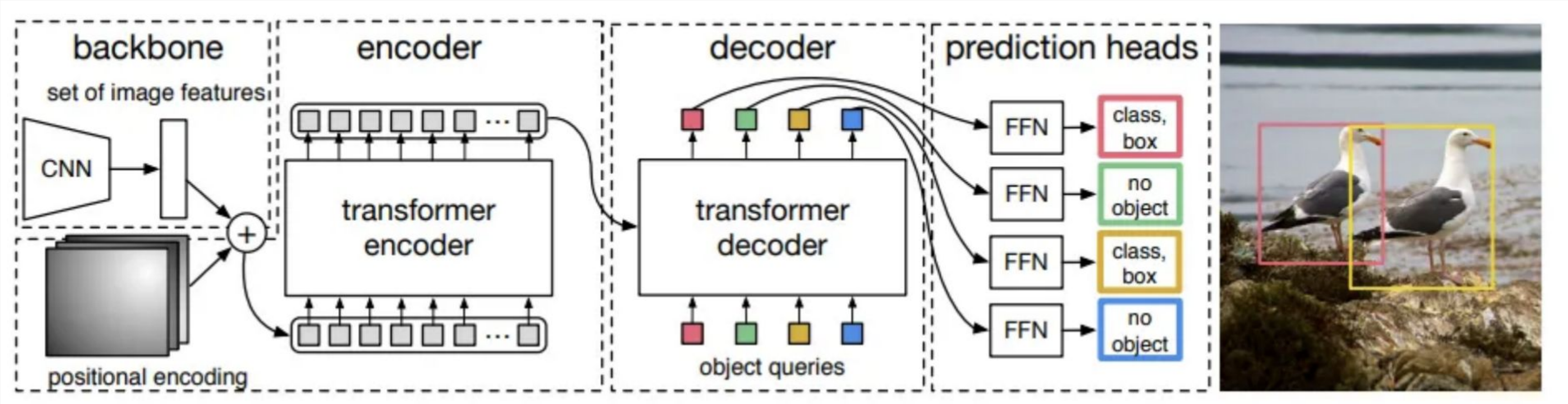


Figure: DERT Model

CLIP (Contrastive Language-Image Pretraining)

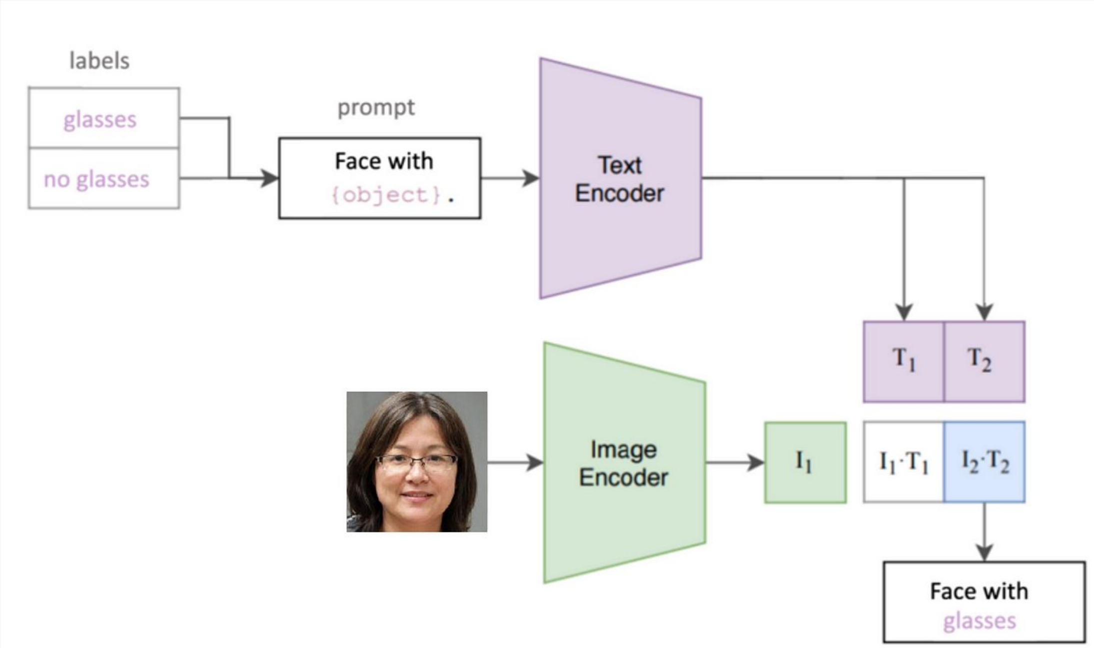


Figure: CLIP Model

Changes Made to the Existing Model

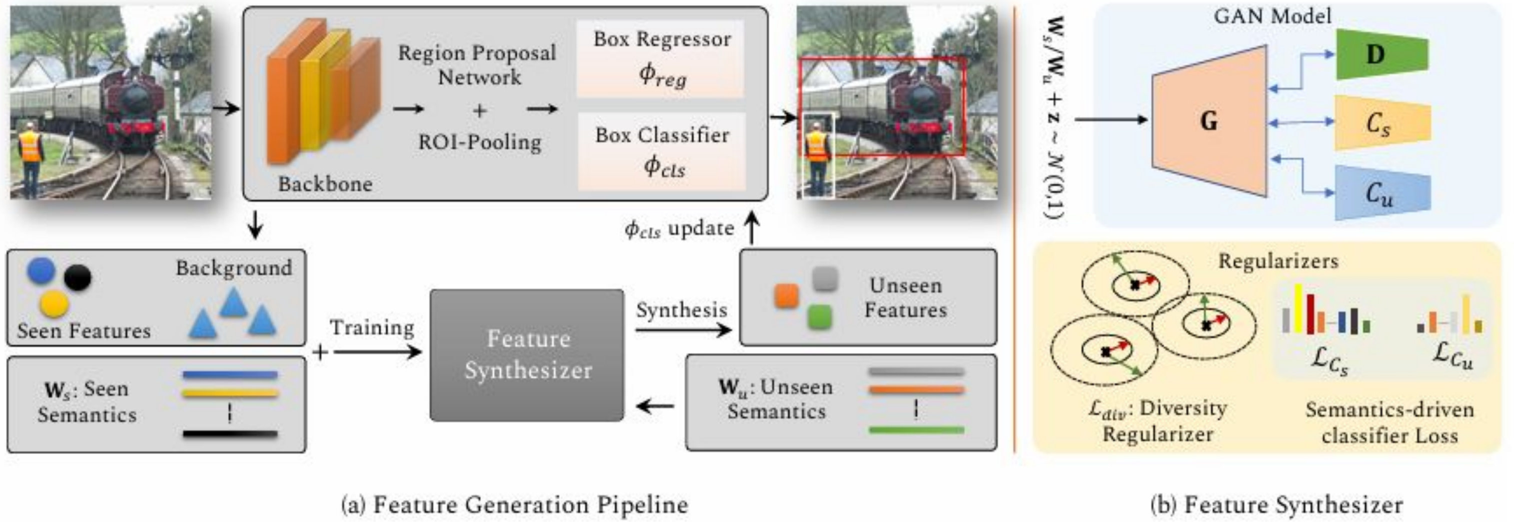


Fig. 1. Overview of proposed generative ZSD approach.

Changes Made to the Existing Model

- Unseen feature synthesis employs FasterRCNN for the prediction of bounding boxes. We instead use DERT for object detection.
- Using CLIP for combining semantic and visual embeddings instead of fixed embeddings like Word2Vec.

Results

Best Match to the given image from the description: unicorn
Confidence score of the description: 0.28559035062789917

Best Match to the given image from the description: horse
Confidence score of the description: 0.25651493668556213

Best Match to the given image from the description: wings
Confidence score of the description: 0.24009530246257782

Result depending on prompts given by the user

Challenges and Limitations

- Currently, our model is unable to detect multiple objects in an image because it relies on the logic of selecting the prediction with the highest confidence score.
- This model produces very low confidence scores for unseen classes.
- When users input words that are entirely unrelated to the image, the model defaults to predicting the background.
- CLIP is applied during the pre-training stage and is applied without fine-tuning the detection task, affecting the model's precision.

Demonstration

Q&A

Open for questions and feedback!

GitHub Link: AnyaAlekar/Unseen-feature-synthesis-for-ZSD