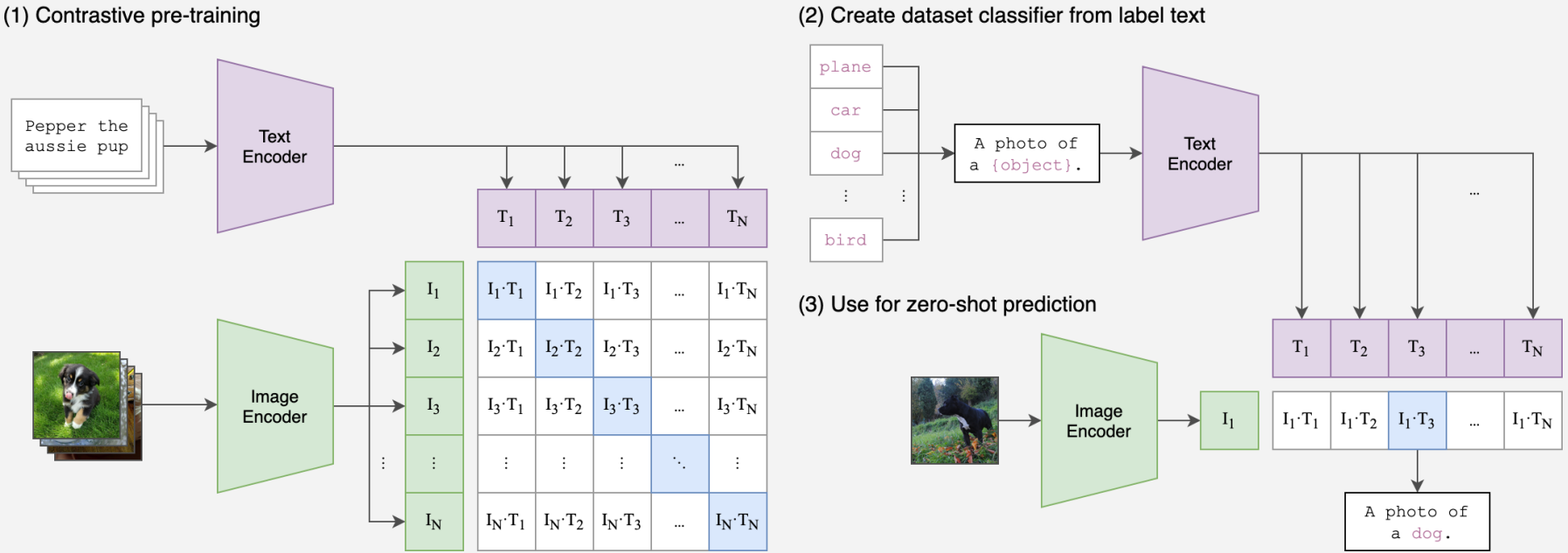
<Vision Transformers>

Practices

Vision Transformers <Task>

Objective. Create a simple video search engine. A search engine should receive a textual query and return a result list of videos, which represent a content, which satisfy query. Additionally, try to play with the a video clusterization.

CLIP: Contrastive-Language-Image-Pretraining



CLIP: Contrastive-Language-Image-Pretraining

```
import torch
import clip
from PIL import Image

device = "cuda" if torch.cuda.is_available() else "cpu"
model, preprocess = clip.load("ViT-B/32", device=device)

image = preprocess(Image.open("CLIP.png")).unsqueeze(0).to(device)
text = clip.tokenize(["a diagram", "a dog", "a cat"]).to(device)

with torch.no_grad():
    image_features = model.encode_image(image)
    text_features = model.encode_text(text)

    logits_per_image, logits_per_text = model(image, text)
    probs = logits_per_image.softmax(dim=-1).cpu().numpy()

print("Label probs:", probs)  # prints: [[0.9927937  0.00421068 0.00299572]]
```

Short description

- Environment Setup

> Setup Google Collab

> Install official CLIP implementation: https://github.com/openai/CLIP

> Ensure access to a GPU for model training and inference (Collab - T4)

Short description

- Data Preparation Setup

> We recommend just to preselect and download some videos from Youtube 8M dataset: https://research.google.com

> But you are free to use any other dataset you'd like

> To download video from Youtube: https://downloaderto.com/en/

Short description

- Model Selection

> Use official CLIP implementation and pretrains

> Make experiments on how to efficiently build video-level descriptors

> Implement video search combining text and video features

> Make some kind of ranking of results

> *Try to clusterize videos as an additional task

<Q&A>