CSE344: Computer Vision

Assignment-3

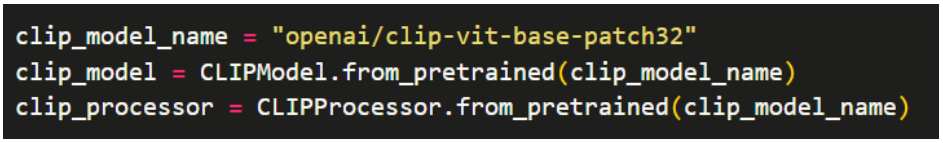
Himanshu Raj (2022216)
April 20, 2025

1. CLIP

1. Using Transformers library for CLIP dependencies

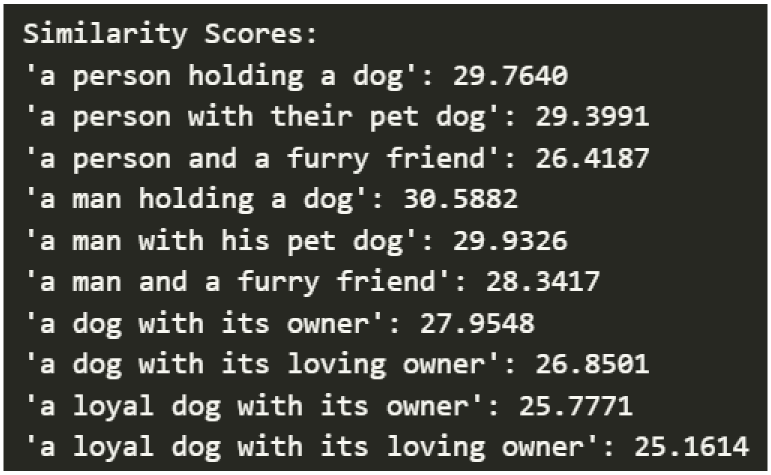
```
from transformers import CLIPProcessor, CLIPModel
from PIL import Image
import torch
```

2. Loading pretrained CLIP model from Hugging Face



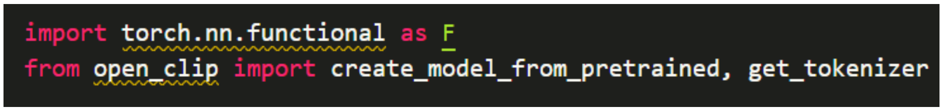
```
clip_model_name = "openai/clip-vit-base-patch32"
clip_model = CLIPModel.from_pretrained(clip_model_name)
clip_processor = CLIPProcessor.from_pretrained(clip_model_name)
```

3. Sentence similarity scores with CLIP



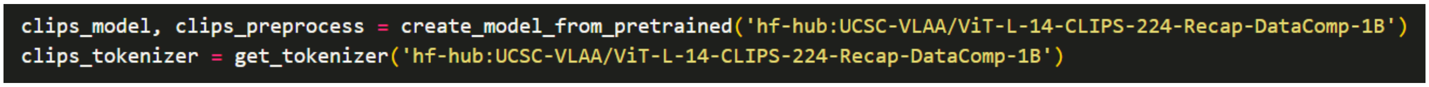
```
Similarity Scores:
'a person holding a dog': 29.7640
'a person with their pet dog': 29.3991
'a person and a furry friend': 26.4187
'a man holding a dog': 30.5882
'a man with his pet dog': 29.9326
'a man and a furry friend': 28.3417
'a dog with its owner': 27.9548
'a dog with its loving owner': 26.8501
'a loyal dog with its owner': 25.7771
'a loyal dog with its loving owner': 25.1614
```

4. Using open_clip library from Hugging Face model page of CLIPS for CLIPS dependencies



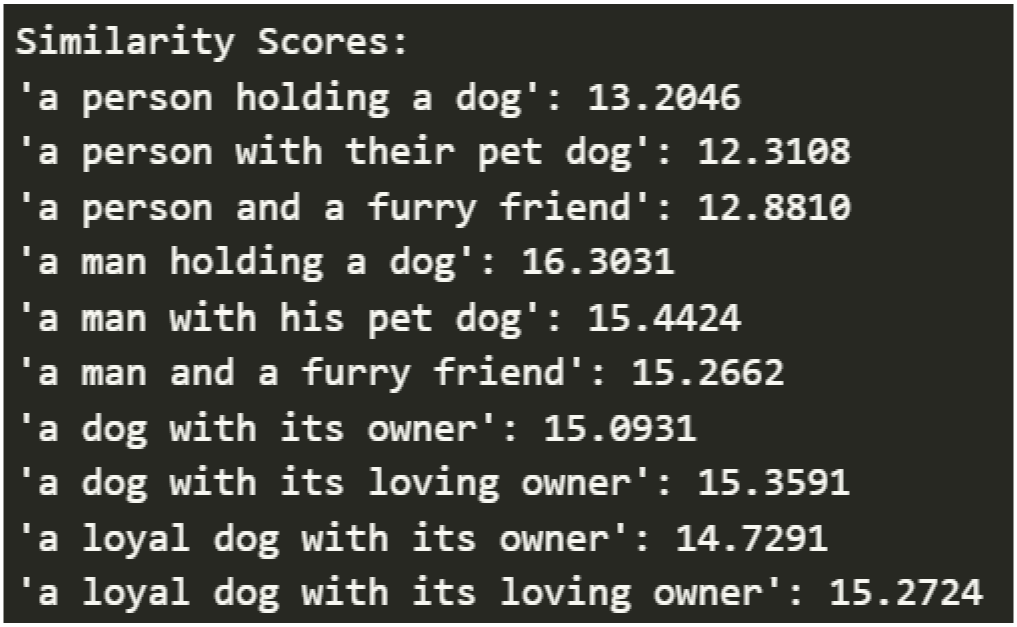
```
import torch.nn.functional as F
from open_clip import create_model_from_pretrained, get_tokenizer
```

5. Loading pretrained CLIPS model from Hugging Face



```
clips_model, clips_preprocess = create_model_from_pretrained('hf-hub:UCSC-VLAA/ViT-L-14-CLIPS-224-Recap-DataComp-1B')
clips_tokenizer = get_tokenizer('hf-hub:UCSC-VLAA/ViT-L-14-CLIPS-224-Recap-DataComp-1B')
```

6. Sentence similarity scores with CLIPS

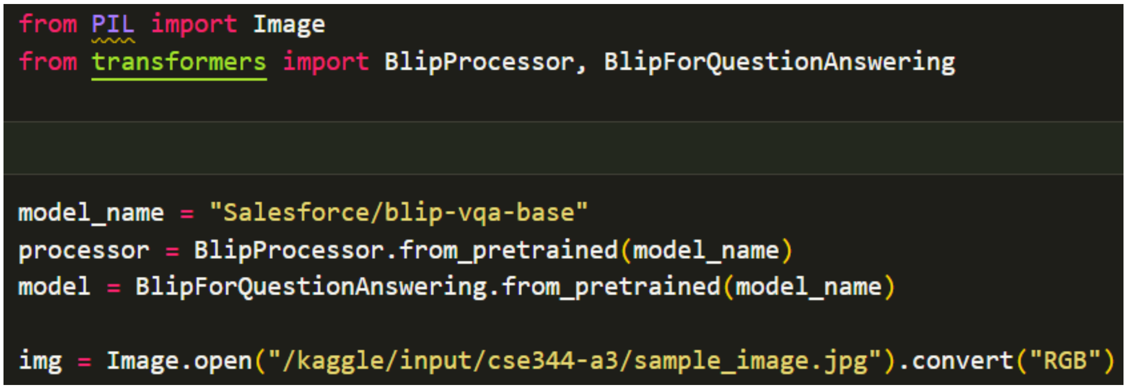


```
Similarity Scores:
'a person holding a dog': 13.2046
'a person with their pet dog': 12.3108
'a person and a furry friend': 12.8810
'a man holding a dog': 16.3031
'a man with his pet dog': 15.4424
'a man and a furry friend': 15.2662
'a dog with its owner': 15.0931
'a dog with its loving owner': 15.3591
'a loyal dog with its owner': 14.7291
'a loyal dog with its loving owner': 15.2724
```

7. CLIP overall score range shows a clear distinction between concrete descriptions and emotionally abstract ones or with extra adjectives mentioning traits outside the image, suggesting CLIP excels at grounding literal, visual elements in the text.
CLIPS scores are more tightly clustered, indicating it may be less sensitive to nuanced semantic differences and potentially more conservative in confidence across captions.
Both models give the highest score to “a man holding a dog”. The scores follow a similar relative trend, but CLIP similarity scores are higher than CLIPS similarity scores probably because of a differently normalized similarity measure.

2. Visual Question Answering

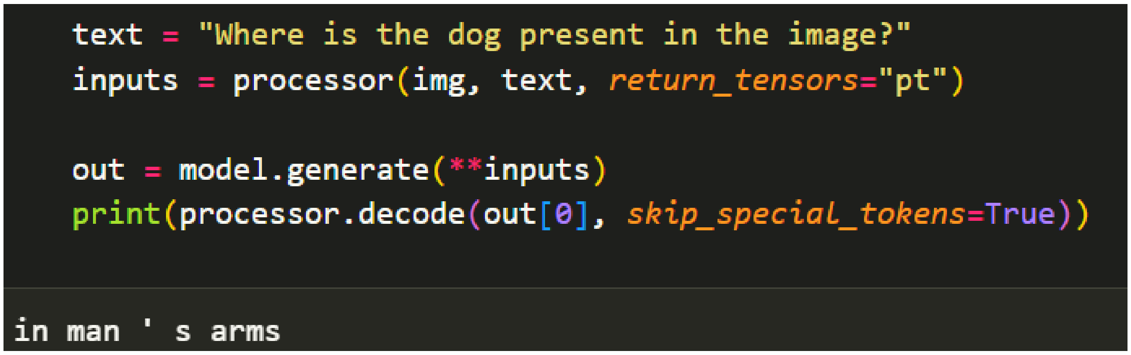
1. Using Transformers library for pretrained BLIP model and its dependencies



```python
from PIL import Image
from transformers import BlipProcessor, BlipForQuestionAnswering
```

```python
model_name = "Salesforce/blip-vqa-base"
processor = BlipProcessor.from_pretrained(model_name)
model = BlipForQuestionAnswering.from_pretrained(model_name)

img = Image.open("/kaggle/input/cse344-a3/sample_image.jpg").convert("RGB")
```

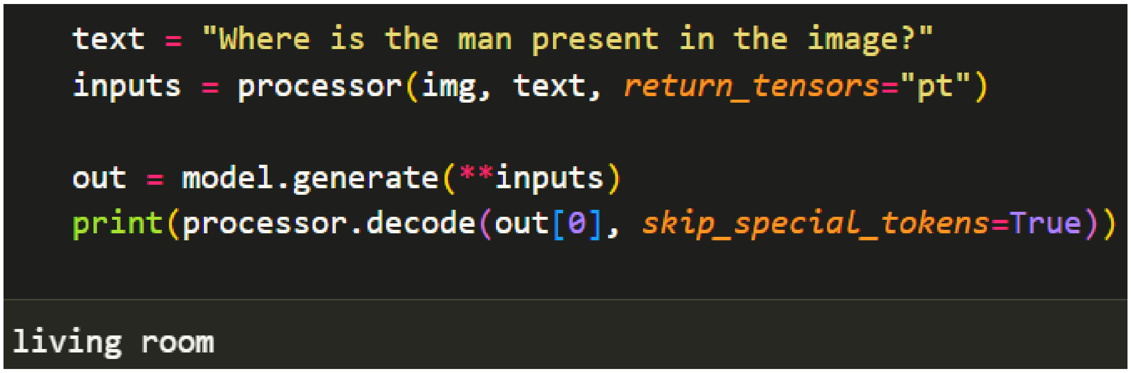
2. Question Answering task with BLIP



```python
    text = "Where is the dog present in the image?"
    inputs = processor(img, text, return_tensors="pt")

    out = model.generate(**inputs)
    print(processor.decode(out[0], skip_special_tokens=True))
```

```
in man ' s arms
```

3. Question Answering task with BLIP



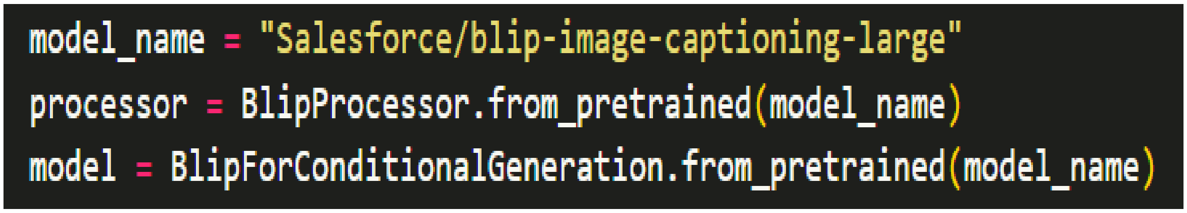
```python
    text = "Where is the man present in the image?"
    inputs = processor(img, text, return_tensors="pt")

    out = model.generate(**inputs)
    print(processor.decode(out[0], skip_special_tokens=True))
```

```
living room
```

4. The output of the BLIP model is very accurate and straight up to the point.

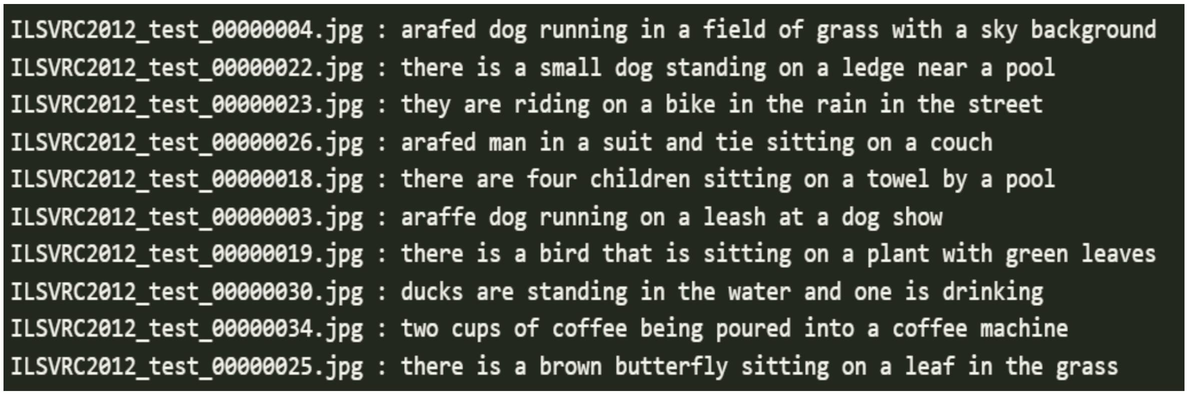
3. BLIP vs CLIP

1. Using Transformers library for pretrained BLIP model



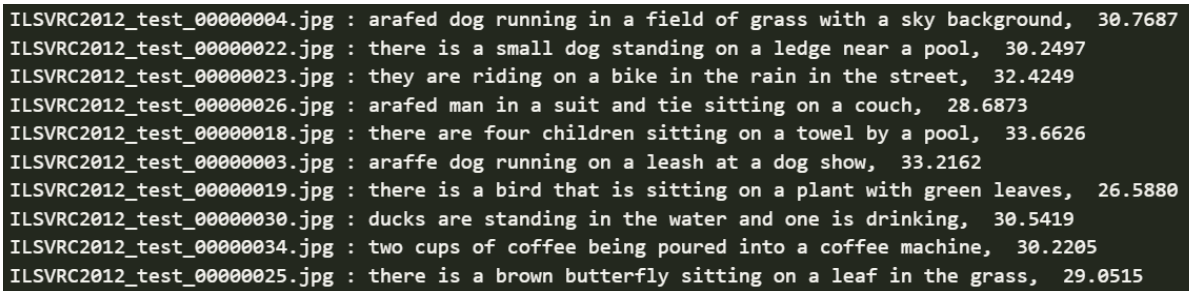
```
model_name = "Salesforce/blip-image-captioning-large"
processor = BlipProcessor.from_pretrained(model_name)
model = BlipForConditionalGeneration.from_pretrained(model_name)
```

2. Caption generation with BLIP



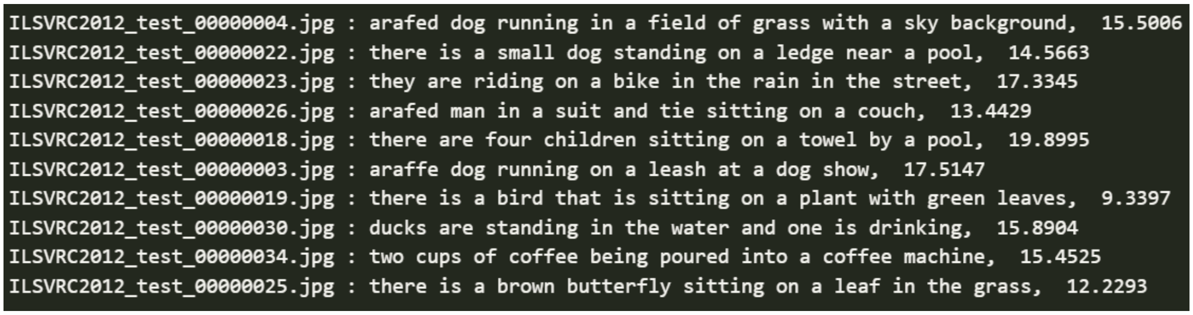
```
ILSVRC2012_test_00000004.jpg : arafed dog running in a field of grass with a sky background
ILSVRC2012_test_00000022.jpg : there is a small dog standing on a ledge near a pool
ILSVRC2012_test_00000023.jpg : they are riding on a bike in the rain in the street
ILSVRC2012_test_00000026.jpg : arafed man in a suit and tie sitting on a couch
ILSVRC2012_test_00000018.jpg : there are four children sitting on a towel by a pool
ILSVRC2012_test_00000003.jpg : araffe dog running on a leash at a dog show
ILSVRC2012_test_00000019.jpg : there is a bird that is sitting on a plant with green leaves
ILSVRC2012_test_00000030.jpg : ducks are standing in the water and one is drinking
ILSVRC2012_test_00000034.jpg : two cups of coffee being poured into a coffee machine
ILSVRC2012_test_00000025.jpg : there is a brown butterfly sitting on a leaf in the grass
```

3. Similarity scores of generated captions by CLIP
[image name : caption, similarity score]



```
ILSVRC2012_test_00000004.jpg : arafed dog running in a field of grass with a sky background,  30.7687
ILSVRC2012_test_00000022.jpg : there is a small dog standing on a ledge near a pool,  30.2497
ILSVRC2012_test_00000023.jpg : they are riding on a bike in the rain in the street,  32.4249
ILSVRC2012_test_00000026.jpg : arafed man in a suit and tie sitting on a couch,  28.6873
ILSVRC2012_test_00000018.jpg : there are four children sitting on a towel by a pool,  33.6626
ILSVRC2012_test_00000003.jpg : araffe dog running on a leash at a dog show,  33.2162
ILSVRC2012_test_00000019.jpg : there is a bird that is sitting on a plant with green leaves,  26.5880
ILSVRC2012_test_00000030.jpg : ducks are standing in the water and one is drinking,  30.5419
ILSVRC2012_test_00000034.jpg : two cups of coffee being poured into a coffee machine,  30.2205
ILSVRC2012_test_00000025.jpg : there is a brown butterfly sitting on a leaf in the grass,  29.0515
```

4. Similarity scores of generated captions by CLIPS
[image name : caption, similarity score]



```
ILSVRC2012_test_00000004.jpg : arafed dog running in a field of grass with a sky background,  15.5006
ILSVRC2012_test_00000022.jpg : there is a small dog standing on a ledge near a pool,  14.5663
ILSVRC2012_test_00000023.jpg : they are riding on a bike in the rain in the street,  17.3345
ILSVRC2012_test_00000026.jpg : arafed man in a suit and tie sitting on a couch,  13.4429
ILSVRC2012_test_00000018.jpg : there are four children sitting on a towel by a pool,  19.8995
ILSVRC2012_test_00000003.jpg : araffe dog running on a leash at a dog show,  17.5147
ILSVRC2012_test_00000019.jpg : there is a bird that is sitting on a plant with green leaves,  9.3397
ILSVRC2012_test_00000030.jpg : ducks are standing in the water and one is drinking,  15.8904
ILSVRC2012_test_00000034.jpg : two cups of coffee being poured into a coffee machine,  15.4525
ILSVRC2012_test_00000025.jpg : there is a brown butterfly sitting on a leaf in the grass,  12.2293
```

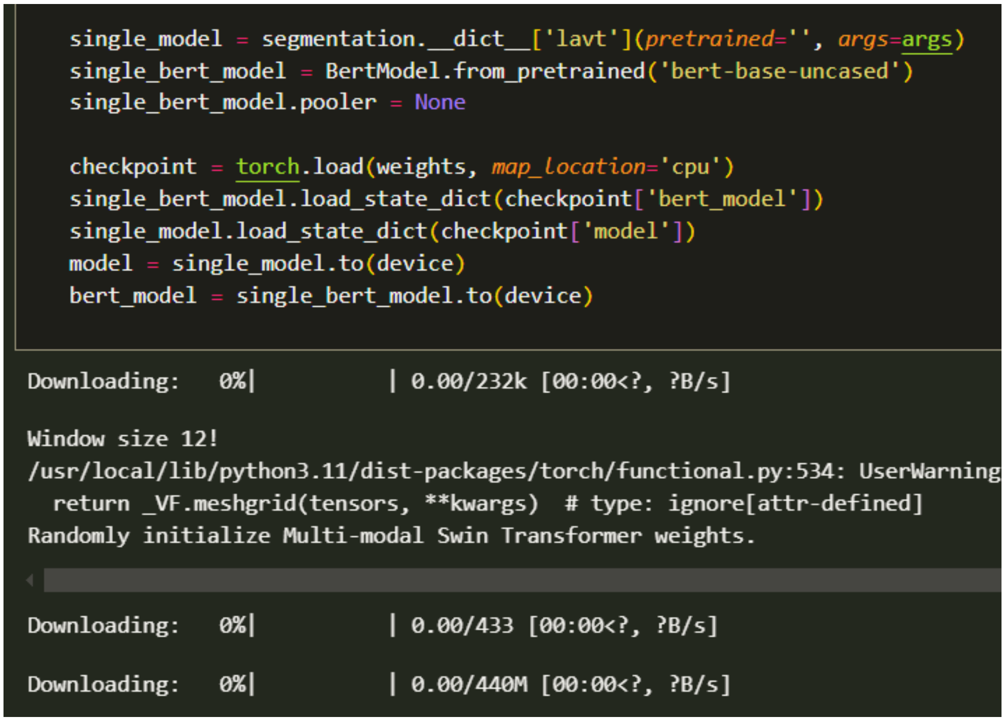
5. Metrics to quantify alignment between CLIP and BLIP outputs

CLIPScore is a metric that can be used to evaluate the quality of an automatic image captioning system. It doesn’t need reference captions. So it can be used for evaluating how well a generated caption (like from BLIP) matches the image itself, without needing reference captions.

Cosine similarity measures how semantically similar two text captions are by comparing their vector embeddings (from CLIP, BERT, etc.). It can be used for measuring semantic similarity between two text captions, usually from different models or reference candidates.

4. Referring Image Segmentation

1. Loading pretrained weights of LAVT model

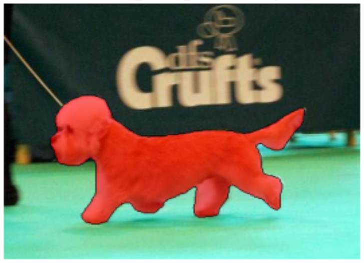


```
    single_model = segmentation.__dict__['lavt'](pretrained='', args=args)
    single_bert_model = BertModel.from_pretrained('bert-base-uncased')
    single_bert_model.pooler = None

    checkpoint = torch.load(weights, map_location='cpu')
    single_bert_model.load_state_dict(checkpoint['bert_model'])
    single_model.load_state_dict(checkpoint['model'])
    model = single_model.to(device)
    bert_model = single_bert_model.to(device)
```

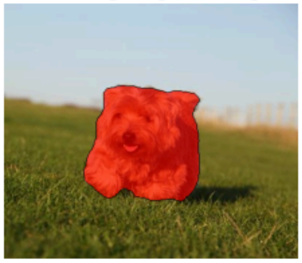
```
Downloading:   0%|          | 0.00/232k [00:00<?, ?B/s]

Window size 12!
/usr/local/lib/python3.11/dist-packages/torch/functional.py:534: UserWarning
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
Randomly initialize Multi-modal Swin Transformer weights.

Downloading:   0%|          | 0.00/433 [00:00<?, ?B/s]

Downloading:   0%|          | 0.00/440M [00:00<?, ?B/s]
```

2. Image segmentation results by LAVT model

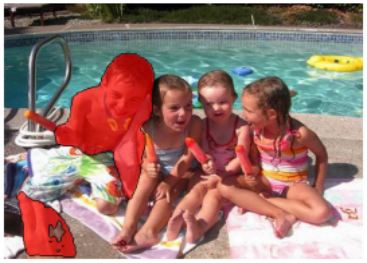
The walking dog in the picture



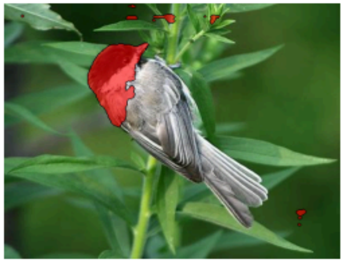
The smiling dog in the grass



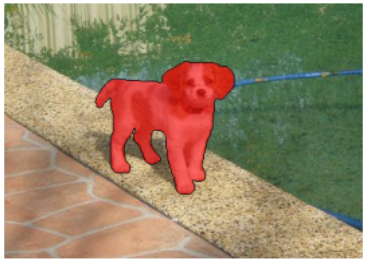
The boy on left smiling and holding icecream



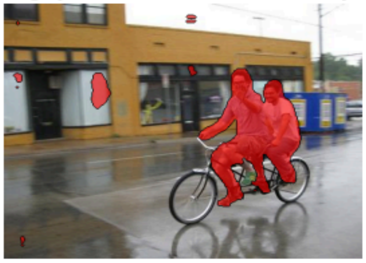
The black gray bird on in the picture



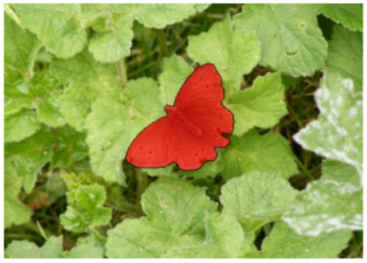
The sad dog standing beside the pool



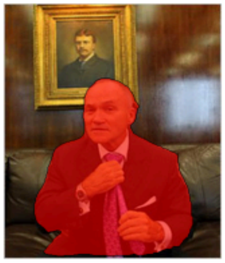
The guy in white shirt on the bicycle



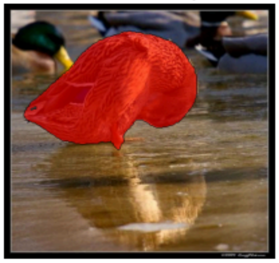
The butterfly in the picture



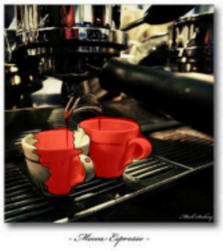
The mang wearing a suite and tie



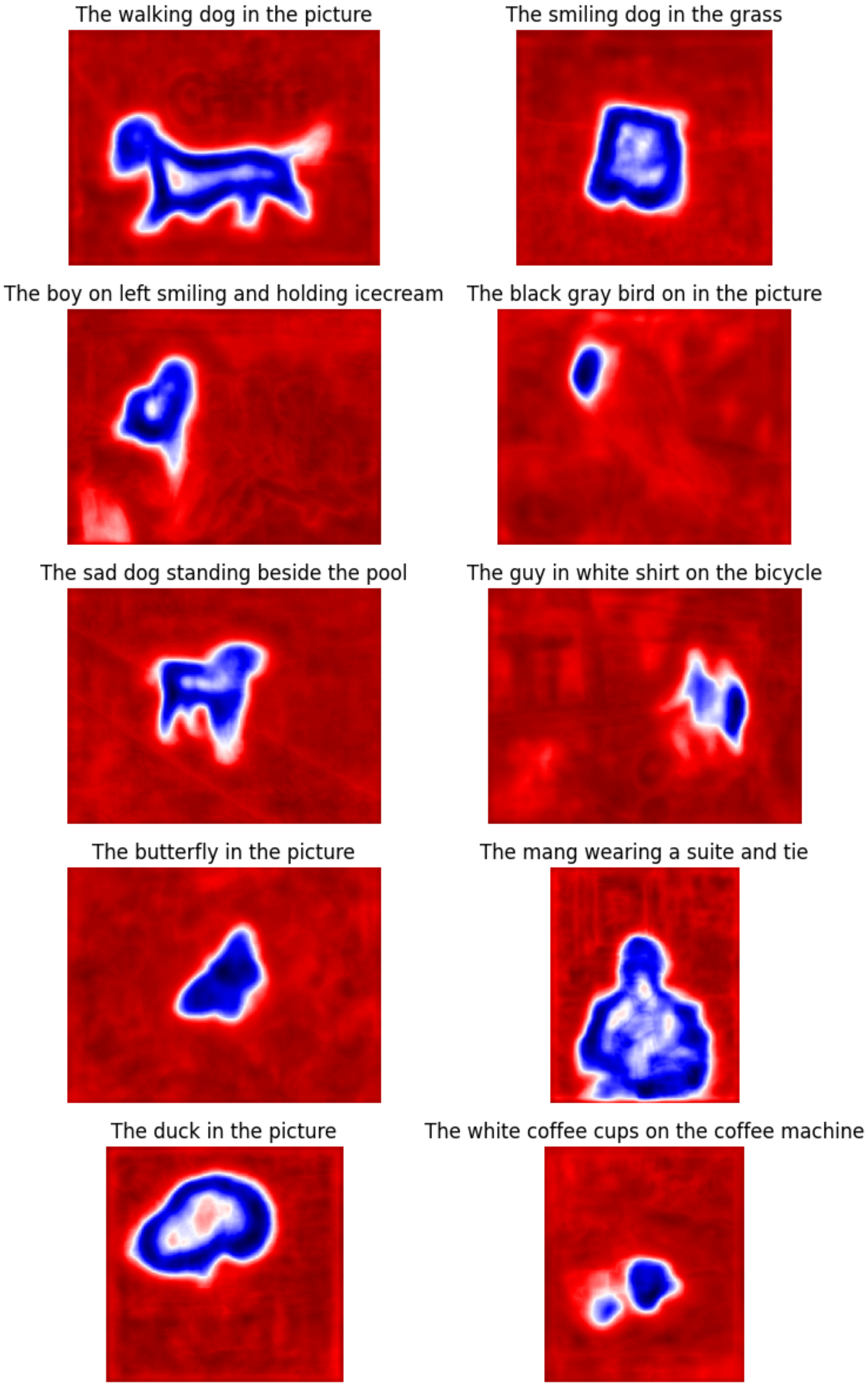
The duck in the picture



The white coffee cups on the coffee machine



3. Y1 feature maps generated in the LAVT model



The walking dog in the picture

The smiling dog in the grass

The boy on left smiling and holding icecream

The black gray bird on in the picture

The sad dog standing beside the pool

The guy in white shirt on the bicycle

The butterfly in the picture

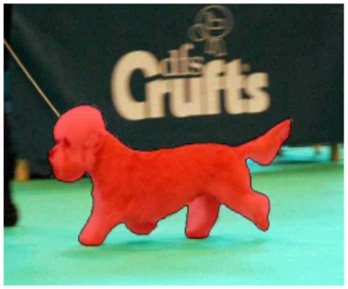
The mang wearing a suite and tie

The duck in the picture

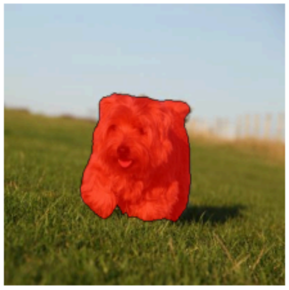
The white coffee cups on the coffee machine

4. Image segmentation results by LAVT model on custom references

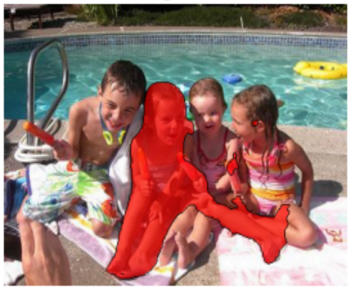
The dog on the mat



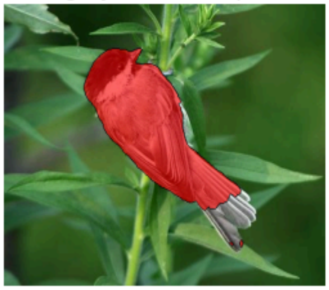
The dog on the grass



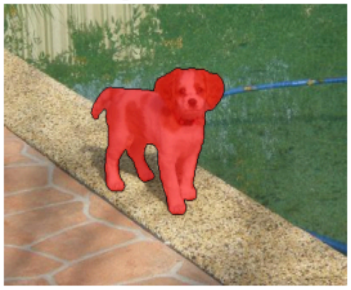
The third girl from left



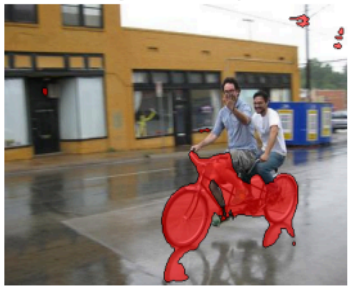
The gray bird in the picture



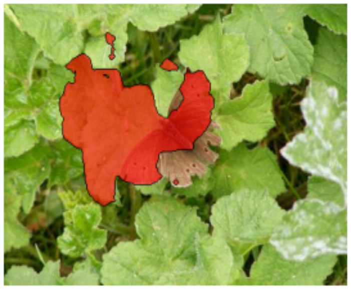
The dog beside the pool



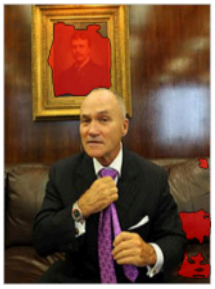
The bicycle in the picture



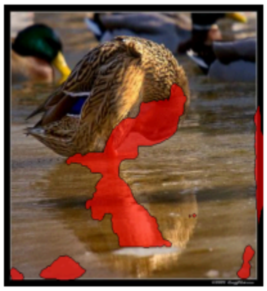
The butterfly on the leaf



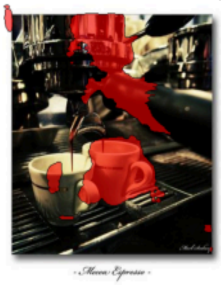
The picture on the wall



The reflection of duck

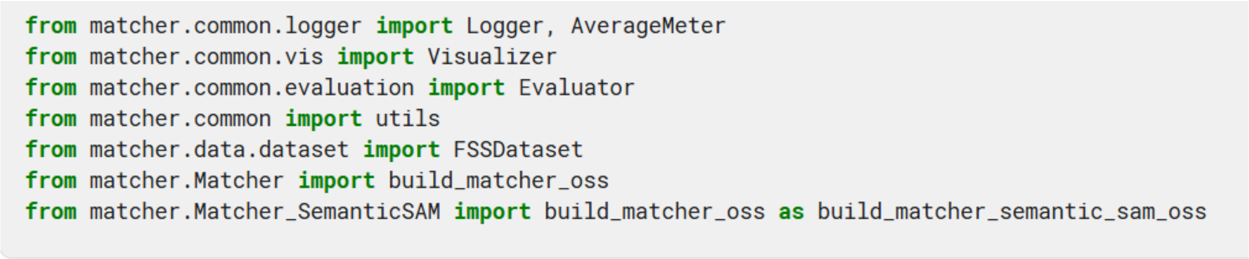


The coffee machine in the picture

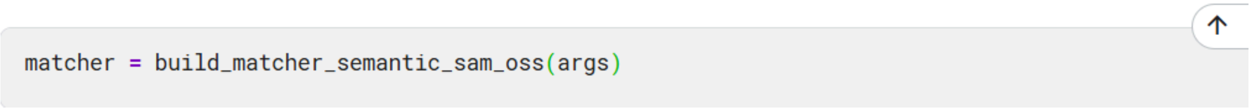


5. Image as Reference

1. Setting up Matcher dependencies and loading model



```
from matcher.common.logger import Logger, AverageMeter
from matcher.common.vis import Visualizer
from matcher.common.evaluation import Evaluator
from matcher.common import utils
from matcher.data.dataset import FSSDataset
from matcher.Matcher import build_matcher_oss
from matcher.Matcher_SemanticSAM import build_matcher_oss as build_matcher_semantic_sam_oss
```



```
matcher = build_matcher_semantic_sam_oss(args)
```