4mlq5souf

December 23, 2024

1 Florence-2-large sample usage

[1]:
```
import sys
print(sys.executable)
```

```
/home/students/anaconda3/envs/florencenv/bin/python
```

[4]:
```
!pip install ollama
```

```
Collecting ollama
  Using cached ollama-0.4.4-py3-none-any.whl.metadata (4.7 kB)
Requirement already satisfied: httpx<0.28.0,>=0.27.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
ollama) (0.27.0)
Collecting pydantic<3.0.0,>=2.9.0 (from ollama)
  Downloading pydantic-2.10.4-py3-none-any.whl.metadata (29 kB)
Requirement already satisfied: anyio in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
httpx<0.28.0,>=0.27.0->ollama) (4.6.2)
Requirement already satisfied: certifi in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
httpx<0.28.0,>=0.27.0->ollama) (2024.12.14)
Requirement already satisfied: httpcore==1.* in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
httpx<0.28.0,>=0.27.0->ollama) (1.0.2)
Requirement already satisfied: idna in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
httpx<0.28.0,>=0.27.0->ollama) (3.7)
Requirement already satisfied: sniffio in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
httpx<0.28.0,>=0.27.0->ollama) (1.3.0)
Requirement already satisfied: h11<0.15,>=0.13 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
httpcore==1.*->httpx<0.28.0,>=0.27.0->ollama) (0.14.0)
Collecting annotated-types>=0.6.0 (from pydantic<3.0.0,>=2.9.0->ollama)
  Using cached annotated_types-0.7.0-py3-none-any.whl.metadata (15 kB)
Collecting pydantic-core==2.27.2 (from pydantic<3.0.0,>=2.9.0->ollama)
  Downloading pydantic_core-2.27.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014
```

```
_x86_64.whl.metadata (6.6 kB)
Requirement already satisfied: typing-extensions>=4.12.2 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
pydantic<3.0.0,>=2.9.0->ollama) (4.12.2)
Requirement already satisfied: exceptiongroup>=1.0.2 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
anyio->httpx<0.28.0,>=0.27.0->ollama) (1.2.0)
Using cached ollama-0.4.4-py3-none-any.whl (13 kB)
Downloading pydantic-2.10.4-py3-none-any.whl (431 kB)
Downloading
pydantic_core-2.27.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(2.0 MB)

2.0/2.0 MB 446.0 kB/s eta 0:00:001m451.7 kB/s eta
0:00:01
Using cached annotated_types-0.7.0-py3-none-any.whl (13 kB)
Installing collected packages: pydantic-core, annotated-types, pydantic, ollama
Successfully installed annotated-types-0.7.0 ollama-0.4.4 pydantic-2.10.4
pydantic-core-2.27.2
```

[5]:
```
!pip list
```

```
Package                   Version
------------------------- -----------
annotated-types           0.7.0
anyio                     4.6.2
argon2-cffi               21.3.0
argon2-cffi-bindings      21.2.0
asttokens                 2.0.5
async-lru                 2.0.4
attrs                     24.3.0
Babel                     2.11.0
backcall                  0.2.0
beautifulsoup4            4.12.3
bleach                    6.2.0
Brotli                    1.0.9
certifi                   2024.12.14
cffi                      1.17.1
charset-normalizer        3.3.2
comm                      0.2.1
debugpy                   1.6.7
decorator                 5.1.1
defusedxml                0.7.1
exceptiongroup            1.2.0
executing                 0.8.3
fastjsonschema            2.20.0
h11                       0.14.0
httpcore                  1.0.2
```

httpx	0.27.0
idna	3.7
importlib_metadata	8.5.0
ipykernel	6.29.5
ipython	8.15.0
ipywidgets	8.1.5
jedi	0.19.2
Jinja2	3.1.4
json5	0.9.25
jsonschema	4.23.0
jsonschema-specifications	2023.7.1
jupyter	1.0.0
jupyter_client	8.6.0
jupyter-console	6.6.3
jupyter_core	5.7.2
jupyter-events	0.10.0
jupyter-lsp	2.2.0
jupyter_server	2.14.1
jupyter_server_terminals	0.4.4
jupyterlab	4.2.5
jupyterlab-pygments	0.1.2
jupyterlab_server	2.27.3
jupyterlab_widgets	3.0.13
MarkupSafe	2.1.3
matplotlib-inline	0.1.6
mistune	2.0.4
nbclient	0.8.0
nbconvert	7.16.4
nbformat	5.10.4
nest-asyncio	1.6.0
notebook	7.2.2
notebook_shim	0.2.3
ollama	0.4.4
overrides	7.4.0
packaging	24.2
pandocfilters	1.5.0
parso	0.8.4
pexpect	4.8.0
pickleshare	0.7.5
pip	24.2
platformdirs	3.10.0
ply	3.11
prometheus_client	0.21.0
prompt-toolkit	3.0.43
psutil	5.9.0
ptyprocess	0.7.0
pure-eval	0.2.2
pycparser	2.21

```
pydantic                  2.10.4
pydantic_core             2.27.2
Pygments                  2.15.1
PyQt5                     5.15.10
PyQt5-sip                 12.13.0
PySocks                   1.7.1
python-dateutil           2.9.0.post0
python-json-logger        3.2.1
pytz                      2024.1
PyYAML                    6.0.2
pyzmq                     26.2.0
qtconsole                 5.6.0
QtPy                      2.4.1
referencing               0.30.2
requests                  2.32.3
rfc3339-validator         0.1.4
rfc3986-validator         0.1.1
rpds-py                   0.10.6
Send2Trash                1.8.2
setuptools                75.1.0
sip                       6.7.12
six                       1.16.0
sniffio                   1.3.0
soupsieve                 2.5
stack-data                0.2.0
terminado                 0.17.1
tinycss2                  1.2.1
tomli                     2.0.1
tornado                   6.4.2
traitlets                 5.14.3
typing_extensions         4.12.2
urllib3                   2.2.3
wcwidth                   0.2.5
webencodings              0.5.1
websocket-client          1.8.0
wheel                     0.44.0
widgetsnbextension        4.0.13
zipp                      3.21.0
```

[7]:
```
!pip install transformers
```

```
Collecting transformers
  Using cached transformers-4.47.1-py3-none-any.whl.metadata (44 kB)
Collecting filelock (from transformers)
  Using cached filelock-3.16.1-py3-none-any.whl.metadata (2.9 kB)
Collecting huggingface-hub<1.0,>=0.24.0 (from transformers)
  Using cached huggingface_hub-0.27.0-py3-none-any.whl.metadata (13 kB)
Collecting numpy>=1.17 (from transformers)
```

```
  Using cached
numpy-2.0.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(60 kB)
Requirement already satisfied: packaging>=20.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
transformers) (6.0.2)
Collecting regex!=2019.12.17 (from transformers)
  Downloading regex-2024.11.6-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_
64.whl.metadata (40 kB)
Requirement already satisfied: requests in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
transformers) (2.32.3)
Collecting tokenizers<0.22,>=0.21 (from transformers)
  Using cached tokenizers-0.21.0-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x
86_64.whl.metadata (6.7 kB)
Collecting safetensors>=0.4.1 (from transformers)
  Downloading safetensors-0.4.5-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x8
6_64.whl.metadata (3.8 kB)
Collecting tqdm>=4.27 (from transformers)
  Using cached tqdm-4.67.1-py3-none-any.whl.metadata (57 kB)
Collecting fsspec>=2023.5.0 (from huggingface-hub<1.0,>=0.24.0->transformers)
  Using cached fsspec-2024.12.0-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
huggingface-hub<1.0,>=0.24.0->transformers) (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
requests->transformers) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
requests->transformers) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
requests->transformers) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
requests->transformers) (2024.12.14)
Using cached transformers-4.47.1-py3-none-any.whl (10.1 MB)
Using cached huggingface_hub-0.27.0-py3-none-any.whl (450 kB)
Using cached
numpy-2.0.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (19.5 MB)
Downloading
regex-2024.11.6-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (780
kB)
```

```
780.9/780.9 kB 3.7 MB/s eta 0:00:00[31m5.4 MB/s
eta 0:00:01
Downloading
safetensors-0.4.5-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (436
kB)
Using cached
tokenizers-0.21.0-cp39-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.0
MB)
Using cached tqdm-4.67.1-py3-none-any.whl (78 kB)
Using cached filelock-3.16.1-py3-none-any.whl (16 kB)
Downloading fsspec-2024.12.0-py3-none-any.whl (183 kB)
Installing collected packages: tqdm, safetensors, regex, numpy, fsspec,
filelock, huggingface-hub, tokenizers, transformers
Successfully installed filelock-3.16.1 fsspec-2024.12.0 huggingface-hub-0.27.0
numpy-2.0.2 regex-2024.11.6 safetensors-0.4.5 tokenizers-0.21.0 tqdm-4.67.1
transformers-4.47.1
```

[9]:
```
!pip install Pillow
```

```
Collecting Pillow
  Using cached pillow-11.0.0-cp39-cp39-manylinux_2_28_x86_64.whl.metadata (9.1
kB)
Using cached pillow-11.0.0-cp39-cp39-manylinux_2_28_x86_64.whl (4.4 MB)
Installing collected packages: Pillow
Successfully installed Pillow-11.0.0
```

[11]:
```
!pip install torch
```

```
Collecting torch
  Downloading torch-2.5.1-cp39-cp39-manylinux1_x86_64.whl.metadata (28 kB)
Requirement already satisfied: filelock in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (4.12.2)
Collecting networkx (from torch)
  Using cached networkx-3.2.1-py3-none-any.whl.metadata (5.2 kB)
Requirement already satisfied: jinja2 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (3.1.4)
Requirement already satisfied: fsspec in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (2024.12.0)
Collecting nvidia-cuda-nvrtc-cu12==12.4.127 (from torch)
  Using cached nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-runtime-cu12==12.4.127 (from torch)
```

```
  Using cached nvidia_cuda_runtime_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-cupti-cu12==12.4.127 (from torch)
  Using cached nvidia_cuda_cupti_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cudnn-cu12==9.1.0.70 (from torch)
  Using cached nvidia_cudnn_cu12-9.1.0.70-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cublas-cu12==12.4.5.8 (from torch)
  Using cached nvidia_cublas_cu12-12.4.5.8-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cufft-cu12==11.2.1.3 (from torch)
  Using cached nvidia_cufft_cu12-11.2.1.3-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-curand-cu12==10.3.5.147 (from torch)
  Using cached nvidia_curand_cu12-10.3.5.147-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cusolver-cu12==11.6.1.9 (from torch)
  Using cached nvidia_cusolver_cu12-11.6.1.9-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cusparse-cu12==12.3.1.170 (from torch)
  Using cached nvidia_cusparse_cu12-12.3.1.170-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-nccl-cu12==2.21.5 (from torch)
  Using cached nvidia_nccl_cu12-2.21.5-py3-none-
manylinux2014_x86_64.whl.metadata (1.8 kB)
Collecting nvidia-nvtx-cu12==12.4.127 (from torch)
  Using cached nvidia_nvtx_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.7 kB)
Collecting nvidia-nvjitlink-cu12==12.4.127 (from torch)
  Using cached nvidia_nvjitlink_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting triton==3.1.0 (from torch)
  Downloading
triton-3.1.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(1.3 kB)
Collecting sympy==1.13.1 (from torch)
  Using cached sympy-1.13.1-py3-none-any.whl.metadata (12 kB)
Collecting mpmath<1.4,>=1.1.0 (from sympy==1.13.1->torch)
  Using cached mpmath-1.3.0-py3-none-any.whl.metadata (8.6 kB)
Requirement already satisfied: MarkupSafe>=2.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
jinja2->torch) (2.1.3)
Downloading torch-2.5.1-cp39-cp39-manylinux1_x86_64.whl (906.5 MB)

906.5/906.5 MB 2.6 MB/s eta 0:00:00m eta
0:00:01[36m0:00:10
Using cached nvidia_cublas_cu12-12.4.5.8-py3-none-manylinux2014_x86_64.whl
```

```
(363.4 MB)
Using cached nvidia_cuda_cupti_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl
(13.8 MB)
Using cached nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl
(24.6 MB)
Using cached nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl
(883 kB)
Using cached nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl (664.8
MB)
Using cached nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl (211.5
MB)
Using cached nvidia_curand_cu12-10.3.5.147-py3-none-manylinux2014_x86_64.whl
(56.3 MB)
Using cached nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.whl
(127.9 MB)
Using cached nvidia_cusparse_cu12-12.3.1.170-py3-none-manylinux2014_x86_64.whl
(207.5 MB)
Using cached nvidia_nccl_cu12-2.21.5-py3-none-manylinux2014_x86_64.whl (188.7
MB)
Using cached nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl
(21.1 MB)
Using cached nvidia_nvtx_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (99 kB)
Using cached sympy-1.13.1-py3-none-any.whl (6.2 MB)
Downloading
triton-3.1.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (209.5 MB)

209.5/209.5 MB 2.7 MB/s eta 0:00:00m eta
0:00:01[36m0:00:03
Using cached networkx-3.2.1-py3-none-any.whl (1.6 MB)
Using cached mpmath-1.3.0-py3-none-any.whl (536 kB)
Installing collected packages: mpmath, triton, sympy, nvidia-nvtx-cu12, nvidia-
nvjitlink-cu12, nvidia-nccl-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-
cuda-runtime-cu12, nvidia-cuda-nvrtc-cu12, nvidia-cuda-cupti-cu12, nvidia-
cublas-cu12, networkx, nvidia-cusparse-cu12, nvidia-cudnn-cu12, nvidia-cusolver-
cu12, torch
Successfully installed mpmath-1.3.0 networkx-3.2.1 nvidia-cublas-cu12-12.4.5.8
nvidia-cuda-cupti-cu12-12.4.127 nvidia-cuda-nvrtc-cu12-12.4.127 nvidia-cuda-
runtime-cu12-12.4.127 nvidia-cudnn-cu12-9.1.0.70 nvidia-cufft-cu12-11.2.1.3
nvidia-curand-cu12-10.3.5.147 nvidia-cusolver-cu12-11.6.1.9 nvidia-cusparse-
cu12-12.3.1.170 nvidia-nccl-cu12-2.21.5 nvidia-nvjitlink-cu12-12.4.127 nvidia-
nvtx-cu12-12.4.127 sympy-1.13.1 torch-2.5.1 triton-3.1.0
```

[13]:
```
!pip install matplotlib
```

```
Collecting matplotlib
  Downloading matplotlib-3.9.4-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86
_64.whl.metadata (11 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
```

```
  Using cached contourpy-1.3.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86
_64.whl.metadata (5.4 kB)
Collecting cycler>=0.10 (from matplotlib)
  Using cached cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.55.3-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86
_64.whl.metadata (165 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Using cached kiwisolver-1.4.7-cp39-cp39-manylinux_2_12_x86_64.manylinux2010_x8
6_64.whl.metadata (6.3 kB)
Requirement already satisfied: numpy>=1.23 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
matplotlib) (2.0.2)
Requirement already satisfied: packaging>=20.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
matplotlib) (11.0.0)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Using cached pyparsing-3.2.0-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: python-dateutil>=2.7 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
matplotlib) (2.9.0.post0)
Collecting importlib-resources>=3.2.0 (from matplotlib)
  Using cached importlib_resources-6.4.5-py3-none-any.whl.metadata (4.0 kB)
Requirement already satisfied: zipp>=3.1.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
importlib-resources>=3.2.0->matplotlib) (3.21.0)
Requirement already satisfied: six>=1.5 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
python-dateutil>=2.7->matplotlib) (1.16.0)
Downloading
matplotlib-3.9.4-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (8.3
MB)

8.3/8.3 MB 356.4 kB/s eta 0:00:00[36m0:00:01m eta
0:00:02
Using cached
contourpy-1.3.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (321
kB)
Using cached cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading
fonttools-4.55.3-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.6
MB)

4.6/4.6 MB 374.8 kB/s eta 0:00:001m376.8 kB/s eta
0:00:01
```

```
Using cached importlib_resources-6.4.5-py3-none-any.whl (36 kB)
Using cached
kiwisolver-1.4.7-cp39-cp39-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (1.6
MB)
Using cached pyparsing-3.2.0-py3-none-any.whl (106 kB)
Installing collected packages: pyparsing, kiwisolver, importlib-resources,
fonttools, cycler, contourpy, matplotlib
Successfully installed contourpy-1.3.0 cycler-0.12.1 fonttools-4.55.3 importlib-
resources-6.4.5 kiwisolver-1.4.7 matplotlib-3.9.4 pyparsing-3.2.0
```

[5]:
```
!pip install timm einops
```

```
Requirement already satisfied: timm in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (1.0.12)
Requirement already satisfied: einops in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (0.8.0)
Requirement already satisfied: torch in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from timm)
(2.5.1)
Requirement already satisfied: torchvision in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from timm)
(0.20.1)
Requirement already satisfied: pyyaml in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from timm)
(6.0.2)
Requirement already satisfied: huggingface_hub in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from timm)
(0.27.0)
Requirement already satisfied: safetensors in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from timm)
(0.4.5)
Requirement already satisfied: filelock in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
huggingface_hub->timm) (3.16.1)
Requirement already satisfied: fsspec>=2023.5.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
huggingface_hub->timm) (2024.12.0)
Requirement already satisfied: packaging>=20.9 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
huggingface_hub->timm) (24.2)
Requirement already satisfied: requests in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
huggingface_hub->timm) (2.32.3)
Requirement already satisfied: tqdm>=4.42.1 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
huggingface_hub->timm) (4.67.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
```

```
huggingface_hub->timm) (4.12.2)
Requirement already satisfied: networkx in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (3.2.1)
Requirement already satisfied: jinja2 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (3.1.4)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (12.4.127)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (12.4.127)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.4.127 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (12.4.127)
Requirement already satisfied: nvidia-cudnn-cu12==9.1.0.70 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (9.1.0.70)
Requirement already satisfied: nvidia-cublas-cu12==12.4.5.8 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (12.4.5.8)
Requirement already satisfied: nvidia-cufft-cu12==11.2.1.3 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (11.2.1.3)
Requirement already satisfied: nvidia-curand-cu12==10.3.5.147 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (10.3.5.147)
Requirement already satisfied: nvidia-cusolver-cu12==11.6.1.9 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (11.6.1.9)
Requirement already satisfied: nvidia-cusparse-cu12==12.3.1.170 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (12.3.1.170)
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (2.21.5)
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (12.4.127)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.4.127 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (12.4.127)
Requirement already satisfied: triton==3.1.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch->timm) (3.1.0)
Requirement already satisfied: sympy==1.13.1 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
```

```
torch->timm) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
sympy==1.13.1->torch->timm) (1.3.0)
Requirement already satisfied: numpy in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torchvision->timm) (2.0.2)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torchvision->timm) (11.0.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
jinja2->torch->timm) (2.1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
requests->huggingface_hub->timm) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
requests->huggingface_hub->timm) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
requests->huggingface_hub->timm) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
requests->huggingface_hub->timm) (2024.12.14)
```

[2]:
```
from transformers import AutoProcessor, AutoModelForCausalLM
from PIL import Image
import requests
import copy
import torch
%matplotlib inline
```

[21]:
```
!pip install torch torchvision torchaudio
```

```
Requirement already satisfied: torch in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (2.5.1)
Requirement already satisfied: torchvision in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (0.20.1)
Requirement already satisfied: torchaudio in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (2.5.1)
Requirement already satisfied: filelock in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (4.12.2)
Requirement already satisfied: networkx in
```

```
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (3.2.1)
Requirement already satisfied: jinja2 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (3.1.4)
Requirement already satisfied: fsspec in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (2024.12.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (12.4.127)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (12.4.127)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.4.127 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (12.4.127)
Requirement already satisfied: nvidia-cudnn-cu12==9.1.0.70 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (9.1.0.70)
Requirement already satisfied: nvidia-cublas-cu12==12.4.5.8 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (12.4.5.8)
Requirement already satisfied: nvidia-cufft-cu12==11.2.1.3 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (11.2.1.3)
Requirement already satisfied: nvidia-curand-cu12==10.3.5.147 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (10.3.5.147)
Requirement already satisfied: nvidia-cusolver-cu12==11.6.1.9 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (11.6.1.9)
Requirement already satisfied: nvidia-cusparse-cu12==12.3.1.170 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (12.3.1.170)
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (2.21.5)
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (12.4.127)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.4.127 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (12.4.127)
Requirement already satisfied: triton==3.1.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (3.1.0)
Requirement already satisfied: sympy==1.13.1 in
```

```
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torch) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
sympy==1.13.1->torch) (1.3.0)
Requirement already satisfied: numpy in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torchvision) (2.0.2)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
torchvision) (11.0.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/home/students/anaconda3/envs/florencenv/lib/python3.9/site-packages (from
jinja2->torch) (2.1.3)
```

[48]:
```
# Delete the model and processor
del model
del processor

# Clear PyTorch's cache (if you're using CUDA)
torch.cuda.empty_cache()

# Optionally, you can also clear the image object if you no longer need it
#del image
```

[49]:
```
model_id = 'microsoft/Florence-2-base'
model = AutoModelForCausalLM.from_pretrained(model_id, trust_remote_code=True,␣
↪torch_dtype='auto').eval().cuda()
processor = AutoProcessor.from_pretrained(model_id, trust_remote_code=True)
```

```
config.json:   0%|          | 0.00/2.43k [00:00<?, ?B/s]

configuration_florence2.py:   0%|          | 0.00/15.1k [00:00<?, ?B/s]

A new version of the following files was downloaded from
https://huggingface.co/microsoft/Florence-2-base:
- configuration_florence2.py
. Make sure to double-check they do not contain any added malicious code. To
avoid downloading new versions of the code file, you can pin a revision.

modeling_florence2.py:   0%|          | 0.00/127k [00:00<?, ?B/s]

A new version of the following files was downloaded from
https://huggingface.co/microsoft/Florence-2-base:
- modeling_florence2.py
. Make sure to double-check they do not contain any added malicious code. To
avoid downloading new versions of the code file, you can pin a revision.

pytorch_model.bin:   0%|          | 0.00/464M [00:00<?, ?B/s]
```

```
huggingface/tokenizers: The current process just got forked, after parallelism
has already been used. Disabling parallelism to avoid deadlocks…
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true |
false)
```

```
preprocessor_config.json:   0%|          | 0.00/806 [00:00<?, ?B/s]
```

```
processing_florence2.py:   0%|          | 0.00/46.4k [00:00<?, ?B/s]
```

```
A new version of the following files was downloaded from
https://huggingface.co/microsoft/Florence-2-base:
- processing_florence2.py
. Make sure to double-check they do not contain any added malicious code. To
avoid downloading new versions of the code file, you can pin a revision.
```

```
tokenizer_config.json:   0%|          | 0.00/34.0 [00:00<?, ?B/s]
```

```
vocab.json:   0%|          | 0.00/1.10M [00:00<?, ?B/s]
```

```
tokenizer.json:   0%|          | 0.00/1.36M [00:00<?, ?B/s]
```

1.1 define the prediction function

[50]:
```python
def run_example(task_prompt, text_input=None):
    if text_input is None:
        prompt = task_prompt
    else:
        prompt = task_prompt + text_input
    inputs = processor(text=prompt, images=image, return_tensors="pt").
 ↪to('cuda', torch.float16)
    generated_ids = model.generate(
      input_ids=inputs["input_ids"].cuda(),
      pixel_values=inputs["pixel_values"].cuda(),
      max_new_tokens=1024,
      early_stopping=False,
      do_sample=False,
      num_beams=3,
    )
    generated_text = processor.batch_decode(generated_ids,␣
 ↪skip_special_tokens=False)[0]
    parsed_answer = processor.post_process_generation(
        generated_text,
        task=task_prompt,
        image_size=(image.width, image.height)
    )

    return parsed_answer
```

1.2 init image

[51]:
```
#url = "https://huggingface.co/datasets/huggingface/documentation-images/
 ↪resolve/main/transformers/tasks/car.jpg?download=true"
#image = Image.open(requests.get(url, stream=True).raw)
from PIL import Image

# Path to the local image
image_path = "/home/students/Downloads/family.jpg"

# Open the image
image = Image.open(image_path)

# Display the image (optional)
image.show()
```

```
huggingface/tokenizers: The current process just got forked, after parallelism
has already been used. Disabling parallelism to avoid deadlocks…
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true |
false)
```

[52]:
```
image
```

[52]:

1.3 Run pre-defined tasks without additional inputs

1.3.1 Caption

[53]:
```
task_prompt = '<CAPTION>'
run_example(task_prompt)
```

[53]:
```
{'<CAPTION>': 'A man and woman holding a baby while standing next to a
fireplace.'}
```

[54]:
```
task_prompt = '<DETAILED_CAPTION>'
run_example(task_prompt)
```

[54]:
```
{'<DETAILED_CAPTION>': 'The image shows a family of four posing for a photo in
front of a fireplace. The man and woman are standing on the floor, each holding
a baby in their arms. The background of the image is a wall, and there are
candles on the mantelpiece.'}
```

[57]:
```
task_prompt = '<MORE_DETAILED_CAPTION>'
run_example(task_prompt)
```

[57]:
```
{'<MORE_DETAILED_CAPTION>': 'The image shows a family of four posing for a photo
in front of a white brick fireplace. The family consists of a man, a woman, and
two young children. The man is holding one of the children in his arms, while
the woman is holding a baby in her arms. The child on the left is wearing a red
and white striped shirt and blue jeans, and the child in the middle is wearing
jeans and a plaid shirt. All four of them are smiling and looking at the camera.
The fireplace has a white mantel and there is a candle on the mantel. The
overall mood of the image is happy and relaxed.'}
```

[59]:
```
# Refined task prompt for age prediction
task_prompt = "Please predict the age of the person in the image."

# Running the example with the age prediction task
run_example(task_prompt)
```

[59]:
```
{'Please predict the age of the person in the image.': 'the age of the
person<loc_409><loc_371><loc_590><loc_834>'}
```

1.3.2 Object detection

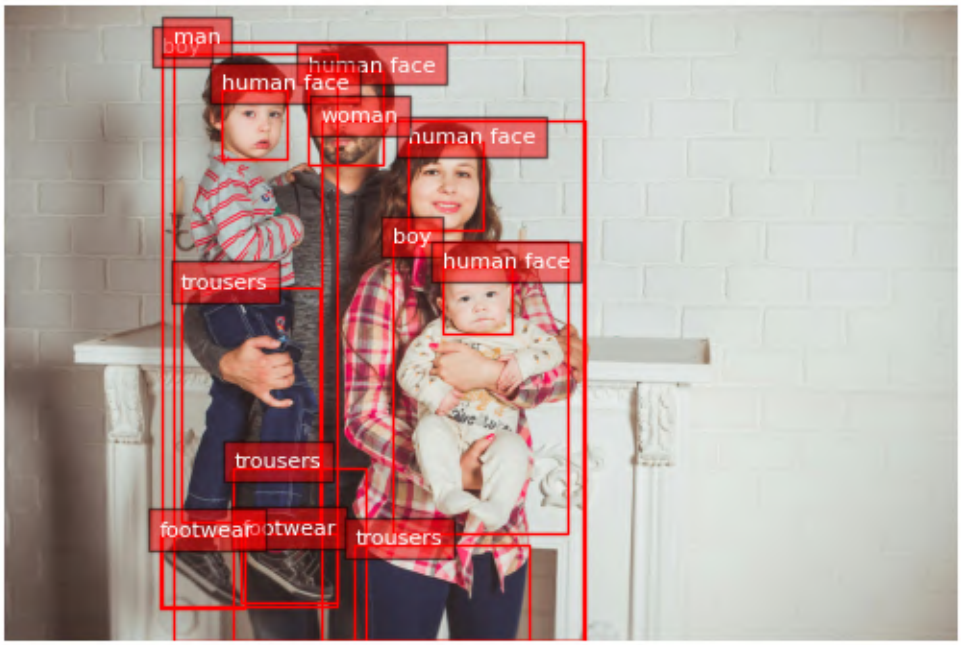
OD results format: {'<OD>': { 'bboxes': [[x1, y1, x2, y2], ...], 'labels': ['label1', 'label2', ...] } }

[60]:
```
task_prompt = '<OD>'
results = run_example(task_prompt)
print(results)
```

```
{'<OD>': {'bboxes': [[857.9520263671875, 254.01600646972656, 1811.8079833984375,
3271.10400390625], [2107.2958984375, 1283.904052734375, 3066.3359375,
2866.751953125], [1288.2239990234375, 2870.2080078125, 1811.8079833984375,
3246.912109375], [842.4000244140625, 2880.576171875, 1308.9599609375,
3278.01611328125], [1651.10400390625, 350.78399658203125, 2060.639892578125,
862.2720336914062], [2190.239990234375, 741.31201171875, 2599.776123046875,
1218.239990234375], [2382.048095703125, 1422.14404296875, 2760.47998046875,
1785.0240478515625], [1179.3599853515625, 451.0080261230469, 1537.0560302734375,
831.1680297851562], [914.9760131835938, 195.26400756835938, 3149.280029296875,
3450.816162109375], [961.6320190429688, 1532.736083984375, 1718.4959716796875,
2804.544189453125], [1246.751953125, 2510.7841796875, 1967.3280029296875,
3450.816162109375], [1905.1199951171875, 2928.9599609375, 2848.60791015625,
3450.816162109375], [1723.6800537109375, 623.8080444335938, 3154.464111328125,
3450.816162109375]], 'labels': ['boy', 'boy', 'footwear', 'footwear', 'human
face', 'human face', 'human face', 'human face', 'man', 'trousers', 'trousers',
'trousers', 'woman']}}
```

[61]:
```python
import matplotlib.pyplot as plt
import matplotlib.patches as patches
def plot_bbox(image, data):
   # Create a figure and axes
    fig, ax = plt.subplots()

    # Display the image
    ax.imshow(image)

    # Plot each bounding box
    for bbox, label in zip(data['bboxes'], data['labels']):
        # Unpack the bounding box coordinates
        x1, y1, x2, y2 = bbox
        # Create a Rectangle patch
        rect = patches.Rectangle((x1, y1), x2-x1, y2-y1, linewidth=1,␣
 ↪edgecolor='r', facecolor='none')
        # Add the rectangle to the Axes
        ax.add_patch(rect)
        # Annotate the label
        plt.text(x1, y1, label, color='white', fontsize=8,␣
 ↪bbox=dict(facecolor='red', alpha=0.5))

    # Remove the axis ticks and labels
    ax.axis('off')

    # Show the plot
    plt.show()
```

[62]:
```python
plot_bbox(image, results['<OD>'])
```



1.3.3 Dense region caption

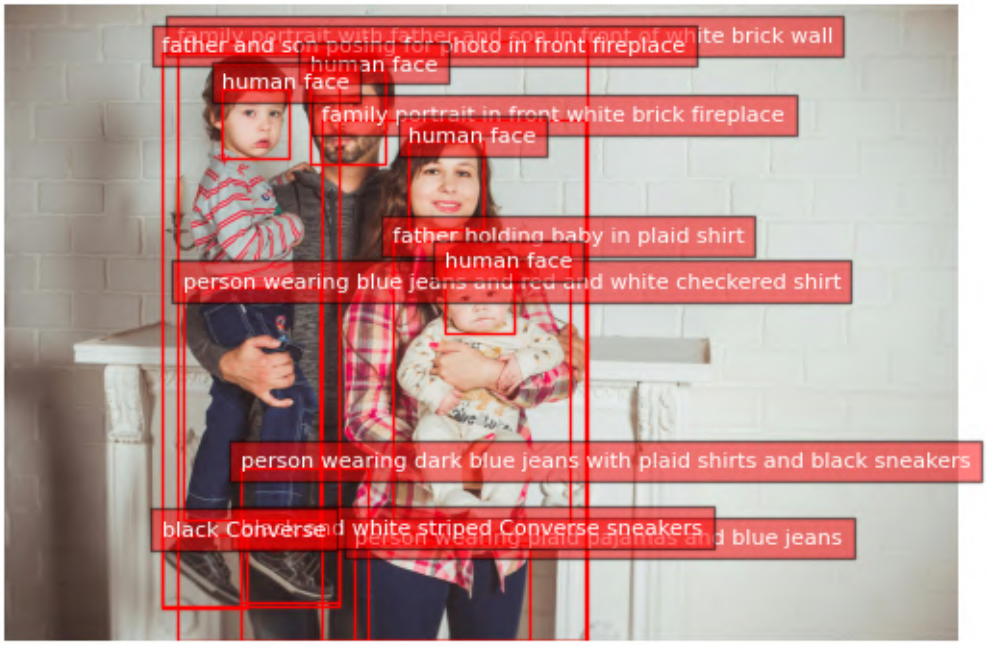
Dense region caption results format: {'<DENSE_REGION_CAPTION>': {'bboxes': [[x1, y1, x2, y2], ...], 'labels': ['label1', 'label2', ...]}}

[63]:
```
task_prompt = '<DENSE_REGION_CAPTION>'
results = run_example(task_prompt)
print(results)
```

```
{'<DENSE_REGION_CAPTION>': {'bboxes': [[935.7119750976562, 195.26400756835938,
3164.83203125, 3450.816162109375], [1728.864013671875, 623.8080444335938,
3154.464111328125, 3450.816162109375], [857.9520263671875, 254.01600646972656,
1816.9920654296875, 3271.10400390625], [2107.2958984375, 1283.904052734375,
3071.52001953125, 2866.751953125], [966.8159790039062, 1532.736083984375,
1718.4959716796875, 2801.088134765625], [1283.0400390625, 2510.7841796875,
1972.511962890625, 3450.816162109375], [1905.1199951171875, 2925.504150390625,
2848.60791015625, 3450.816162109375], [1656.2879638671875, 350.78399658203125,
2065.823974609375, 862.2720336914062], [2195.424072265625, 741.31201171875,
2604.9599609375, 1218.239990234375], [1288.2239990234375, 2870.2080078125,
1816.9920654296875, 3246.912109375], [852.7680053710938, 2880.576171875,
1314.14404296875, 3274.56005859375], [2392.416015625, 1422.14404296875,
2765.6640625, 1785.0240478515625], [1184.5439453125, 451.0080261230469,
1547.4239501953125, 831.1680297851562]], 'labels': ['family portrait with father
and son in front of white brick wall', 'family portrait in front white brick
fireplace', 'father and son posing for photo in front fireplace', 'father
holding baby in plaid shirt', 'person wearing blue jeans and red and white
```

checkered shirt', 'person wearing dark blue jeans with plaid shirts and black sneakers', 'person wearing plaid pajamas and blue jeans', 'human face', 'human face', 'black and white striped Converse sneakers', 'black Converse', 'human face', 'human face']}}

[64]:
```
plot_bbox(image, results['<DENSE_REGION_CAPTION>'])
```



1.3.4 Region proposal

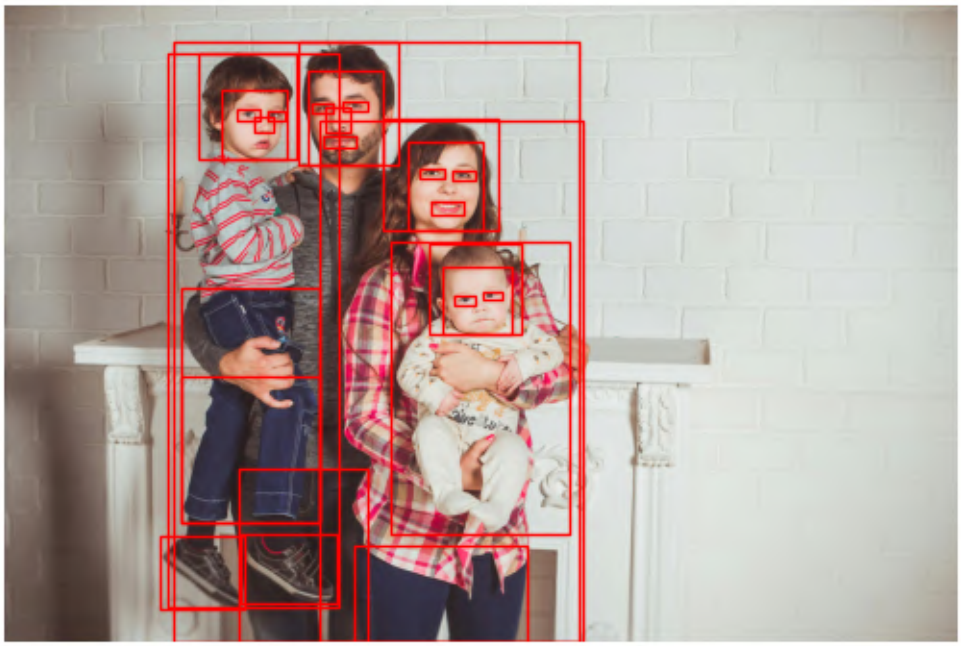
Region proposal results format: {'' : {'bboxes': [[x1, y1, x2, y2], ...], 'labels': ['',', ...]}}

[65]:
```
task_prompt = '<REGION_PROPOSAL>'
results = run_example(task_prompt)
print(results)
```

```
{'<REGION_PROPOSAL>': {'bboxes': [[920.1599731445312, 195.26400756835938,
3128.5439453125, 3450.816162109375], [1713.31201171875, 627.2640380859375,
3149.280029296875, 3450.816162109375], [883.8720092773438, 254.01600646972656,
1816.9920654296875, 3271.10400390625], [2102.112060546875, 1283.904052734375,
3071.52001953125, 2866.751953125], [961.6320190429688, 1532.736083984375,
1718.4959716796875, 2804.544189453125], [1277.85595703125, 2510.7841796875,
1972.511962890625, 3450.816162109375], [961.6320190429688, 2013.1199951171875,
1713.31201171875, 2804.544189453125], [1905.1199951171875, 2928.9599609375,
2843.424072265625, 3450.816162109375], [1599.2640380859375, 191.80799865722656,
2143.583984375, 865.72802734375], [2060.639892578125, 616.89599609375,
2682.719970703125, 1221.696044921875], [1054.9439697265625, 254.01600646972656,
```

1588.89599609375, 831.1680297851562], [2304.2880859375, 1287.3599853515625,
2807.135986328125, 1785.0240478515625], [1645.9200439453125, 350.78399658203125,
2060.639892578125, 862.2720336914062], [2190.239990234375, 741.31201171875,
2599.776123046875, 1218.239990234375], [1288.2239990234375, 2870.2080078125,
1806.6240234375, 3246.912109375], [842.4000244140625, 2880.576171875,
1308.9599609375, 3274.56005859375], [2387.23193359375, 1422.14404296875,
2760.47998046875, 1785.0240478515625], [1179.3599853515625, 451.0080261230469,
1537.0560302734375, 831.1680297851562], [1744.416015625, 554.68798828125,
1879.199951171875, 682.5599975585938], [2319.840087890625, 1059.2640380859375,
2501.280029296875, 1131.840087890625], [1739.2320556640625, 703.2960205078125,
1915.488037109375, 768.9600219726562], [1360.800048828125, 603.072021484375,
1464.47998046875, 689.4720458984375], [2252.447998046875, 879.552001953125,
2392.416015625, 938.3040161132812], [2439.072021484375, 893.3760375976562,
2563.488037109375, 948.6719970703125], [1842.9119873046875, 516.6719970703125,
1972.511962890625, 571.968017578125], [1671.8399658203125, 530.4960327148438,
1791.072021484375, 585.7919921875], [1267.488037109375, 565.0560302734375,
1386.719970703125, 620.3519897460938], [2444.256103515625, 1570.7520751953125,
2558.303955078125, 1622.592041015625], [2599.776123046875, 1550.0159912109375,
2708.639892578125, 1598.4000244140625], [1428.1920166015625, 568.5120239257812,
1521.5040283203125, 620.3519897460938]], 'labels': ['', '', '', '', '', '', '',
'', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '',
'', '', '']}}

[66]:
```
plot_bbox(image, results['<REGION_PROPOSAL>'])
```



1.4 Run pre-defined tasks that requires additional inputs

1.4.1 Phrase Grounding

Phrase grounding results format: {'<CAPTION_TO_PHRASE_GROUNDING>': {'bboxes': [[x1, y1, x2, y2], ...], 'labels': ['', '', ...]}}

[67]:
```
task_prompt = '<CAPTION_TO_PHRASE_GROUNDING>'
results = run_example(task_prompt, text_input="A green car parked in front of a␣
↪yellow building.")
print(results)
```

```
{'<CAPTION_TO_PHRASE_GROUNDING>': {'bboxes': [[2.5920000076293945,
1.7280000448226929, 5176.22412109375, 3450.816162109375]], 'labels': ['a yellow
building']}}
```

[68]:
```
plot_bbox(image, results['<CAPTION_TO_PHRASE_GROUNDING>'])
```

1.4.2 Referring expression segmentation

Referring expression segmentation results format: {'<REFERRING_EXPRESSION_SEGMENTATION>': {'Polygons': [[[polygon]], ...], 'labels': ['', '', ...]}}, one object is represented by a list of polygons. each polygon is [x1, y1, x2, y2, ..., xn, yn]

[70]:
```
task_prompt = '<REFERRING_EXPRESSION_SEGMENTATION>'
results = run_example(task_prompt, text_input="peoples")
print(results)
```

{'<REFERRING_EXPRESSION_SEGMENTATION>': {'polygons': [[[1127.52001953125,
364.6080017089844, 1143.072021484375, 343.87200927734375, 1168.991943359375,
319.6800231933594, 1184.5439453125, 309.31201171875, 1200.095947265625,
298.9440002441406, 1215.64794921875, 292.0320129394531, 1231.199951171875,
285.1199951171875, 1246.751953125, 278.2080078125, 1262.303955078125,
271.2960205078125, 1277.85595703125, 267.8399963378906, 1288.2239990234375,
271.2960205078125, 1303.7760009765625, 274.75201416015625, 1340.06396484375,
271.2960205078125, 1355.615966796875, 274.75201416015625, 1371.16796875,
278.2080078125, 1386.719970703125, 281.66400146484375, 1397.0880126953125,
285.1199951171875, 1407.4560546875, 288.5760192871094, 1417.823974609375,
295.4880065917969, 1428.1920166015625, 298.9440002441406, 1438.56005859375,
305.85601806640625, 1448.927978515625, 312.76800537109375, 1459.2960205078125,
319.6800231933594, 1469.6640625, 326.5920104980469, 1480.031982421875,
333.5039978027344, 1490.4000244140625, 340.416015625, 1505.9520263671875,
354.2400207519531, 1521.5040283203125, 371.52001953125, 1531.8719482421875,
385.3440246582031, 1542.239990234375, 395.7120056152344, 1547.4239501953125,
406.08001708984375, 1552.6080322265625, 416.447998046875, 1557.7919921875,
426.8160095214844, 1562.9759521484375, 437.18402099609375, 1568.1600341796875,
447.552001953125, 1573.343994140625, 457.9200134277344, 1573.343994140625,
475.20001220703125, 1568.1600341796875, 489.0240173339844, 1562.9759521484375,
499.3919982910156, 1557.7919921875, 509.760009765625, 1552.6080322265625,
520.1279907226562, 1552.6080322265625, 530.4960327148438, 1547.4239501953125,
540.864013671875, 1542.239990234375, 551.2319946289062, 1537.0560302734375,
561.6000366210938, 1531.8719482421875, 571.968017578125, 1531.8719482421875,
582.3359985351562, 1526.68798828125, 592.7040405273438, 1521.5040283203125,
603.072021484375, 1521.5040283203125, 616.89599609375, 1516.3199462890625,
627.2640380859375, 1516.3199462890625, 644.5440063476562, 1511.135986328125,
658.3680419921875, 1511.135986328125, 672.1920166015625, 1505.9520263671875,
686.0159912109375, 1505.9520263671875, 703.2960205078125, 1500.7679443359375,
717.1199951171875, 1500.7679443359375, 727.488037109375, 1495.583984375,
737.8560180664062, 1490.4000244140625, 748.2239990234375, 1485.216064453125,
758.592041015625, 1474.8480224609375, 772.416015625, 1464.47998046875,
782.7839965820312, 1454.112060546875, 793.1520385742188, 1443.7440185546875,
800.0640258789062, 1433.3759765625, 806.9760131835938, 1423.008056640625,
813.8880004882812, 1412.6400146484375, 820.800048828125, 1402.27197265625,
824.2560424804688, 1391.904052734375, 831.1680297851562, 1381.5360107421875,
838.0800170898438, 1376.35205078125, 848.447998046875, 1376.35205078125,
869.1840209960938, 1381.5360107421875, 883.0079956054688, 1386.719970703125,
893.3760375976562, 1391.904052734375, 903.7440185546875, 1397.0880126953125,
914.1119995117188, 1402.27197265625, 924.4800415039062, 1412.6400146484375,
934.8480224609375, 1423.008056640625, 945.2160034179688, 1433.3759765625,
955.5840454101562, 1443.7440185546875, 965.9520263671875, 1448.927978515625,
976.3200073242188, 1454.112060546875, 986.6880493164062, 1459.2960205078125,
997.0560302734375, 1464.47998046875, 1007.4240112304688, 1464.47998046875,
1017.7920532226562, 1469.6640625, 1028.1600341796875, 1469.6640625,
1038.528076171875, 1474.8480224609375, 1048.89599609375, 1474.8480224609375,
1059.2640380859375, 1480.031982421875, 1069.632080078125, 1485.216064453125,
1080.0, 1485.216064453125, 1090.3680419921875, 1490.4000244140625,

1100.736083984375, 1495.583984375, 1111.10400390625, 1500.7679443359375,
1121.4720458984375, 1511.135986328125, 1131.840087890625, 1521.5040283203125,
1135.2960205078125, 1531.8719482421875, 1138.7520751953125, 1547.4239501953125,
1138.7520751953125, 1562.9759521484375, 1142.2080078125, 1573.343994140625,
1145.6640625, 1583.7120361328125, 1149.1199951171875, 1594.0799560546875,
1152.5760498046875, 1604.447998046875, 1159.488037109375, 1614.8160400390625,
1169.8560791015625, 1620.0, 1180.2239990234375, 1625.1839599609375,
1187.135986328125, 1630.3680419921875, 1197.5040283203125, 1635.552001953125,
1207.8720703125, 1635.552001953125, 1256.2559814453125, 1630.3680419921875,
1270.080078125, 1625.1839599609375, 1280.447998046875, 1620.0,
1290.8160400390625, 1609.6319580078125, 1301.18408203125, 1599.2640380859375,
1308.0960693359375, 1588.89599609375, 1315.008056640625, 1578.5279541015625,
1321.9200439453125, 1573.343994140625, 1328.83203125, 1568.1600341796875,
1339.2000732421875, 1568.1600341796875, 1356.47998046875, 1568.1600341796875,
1422.14404296875, 1573.343994140625, 1442.8800048828125, 1573.343994140625,
1453.248046875, 1578.5279541015625, 1467.072021484375, 1578.5279541015625,
1501.632080078125, 1573.343994140625, 1515.4560546875, 1568.1600341796875,
1522.3680419921875, 1557.7919921875, 1529.280029296875, 1552.6080322265625,
1536.1920166015625, 1552.6080322265625, 1546.56005859375, 1557.7919921875,
1556.9281005859375, 1557.7919921875, 1570.7520751953125, 1562.9759521484375,
1581.1199951171875, 1562.9759521484375, 1591.488037109375, 1568.1600341796875,
1601.8560791015625, 1568.1600341796875, 1612.2239990234375, 1573.343994140625,
1626.048095703125, 1573.343994140625, 1691.7120361328125, 1568.1600341796875,
1705.5360107421875, 1568.1600341796875, 1729.72802734375, 1562.9759521484375,
1740.0960693359375, 1562.9759521484375, 1760.83203125, 1557.7919921875,
1771.2000732421875, 1557.7919921875, 1781.5679931640625, 1552.6080322265625,
1791.93603515625, 1552.6080322265625, 1802.3040771484375, 1557.7919921875,
1812.6719970703125, 1568.1600341796875, 1823.0400390625, 1578.5279541015625,
1829.9520263671875, 1588.89599609375, 1833.4080810546875, 1599.2640380859375,
1836.864013671875, 1609.6319580078125, 1840.320068359375, 1620.0,
1847.2320556640625, 1630.3680419921875, 1854.14404296875, 1635.552001953125,
1864.5120849609375, 1635.552001953125, 1878.3360595703125, 1630.3680419921875,
1888.7041015625, 1630.3680419921875, 1899.072021484375, 1625.1839599609375,
1909.4400634765625, 1625.1839599609375, 1919.80810546875, 1620.0,
1930.176025390625, 1614.8160400390625, 1937.0880126953125, 1609.6319580078125,
1944.0, 1604.447998046875, 1950.912109375, 1604.447998046875, 1961.280029296875,
1609.6319580078125, 1971.6480712890625, 1614.8160400390625, 1982.0159912109375,
1620.0, 1988.9281005859375, 1625.1839599609375, 1999.2960205078125,
1630.3680419921875, 2006.2080078125, 1635.552001953125, 2016.5760498046875,
1640.7359619140625, 2023.488037109375, 1645.9200439453125, 2033.8560791015625,
1651.10400390625, 2040.76806640625, 1656.2879638671875, 2051.135986328125,
1661.4720458984375, 2058.048095703125, 1666.656005859375, 2068.416015625,
1671.8399658203125, 2078.783935546875, 1677.0240478515625, 2089.152099609375,
1682.2080078125, 2096.06396484375, 1687.3919677734375, 2106.43212890625,
1687.3919677734375, 2116.800048828125, 1692.5760498046875, 2127.16796875,
1692.5760498046875, 2137.5361328125, 1697.760009765625, 2147.904052734375,
1697.760009765625, 2158.27197265625, 1702.9439697265625, 2168.64013671875,
1702.9439697265625, 2179.008056640625, 1708.1280517578125, 2189.3759765625,

1708.1280517578125, 2199.744140625, 1713.31201171875, 2210.112060546875,
1713.31201171875, 2220.47998046875, 1708.1280517578125, 2227.39208984375,
1702.9439697265625, 2234.303955078125, 1697.760009765625, 2241.216064453125,
1697.760009765625, 2251.583984375, 1692.5760498046875, 2261.9521484375,
1692.5760498046875, 2272.320068359375, 1687.3919677734375, 2282.68798828125,
1687.3919677734375, 2293.05615234375, 1682.2080078125, 2303.424072265625,
1682.2080078125, 2313.7919921875, 1677.0240478515625, 2324.16015625,
1677.0240478515625, 2334.528076171875, 1671.8399658203125, 2341.43994140625,
1671.8399658203125, 2351.80810546875, 1666.656005859375, 2362.176025390625,
1666.656005859375, 2372.5439453125, 1661.4720458984375, 2382.912109375,
1661.4720458984375, 2393.280029296875, 1656.2879638671875, 2407.10400390625,
1656.2879638671875, 2424.384033203125, 1651.10400390625, 2445.1201171875,
1651.10400390625, 2465.85595703125, 1645.9200439453125, 2476.22412109375,
1645.9200439453125, 2496.9599609375, 1640.7359619140625, 2507.328125,
1640.7359619140625, 2517.696044921875, 1635.552001953125, 2528.06396484375,
1635.552001953125, 2538.43212890625, 1630.3680419921875, 2548.800048828125,
1630.3680419921875, 2562.6240234375, 1625.1839599609375, 2572.9921875,
1625.1839599609375, 2586.816162109375, 1620.0, 2597.18408203125, 1620.0,
2607.552001953125, 1614.8160400390625, 2617.920166015625, 1614.8160400390625,
2628.2880859375, 1609.6319580078125, 2638.656005859375, 1609.6319580078125,
2649.024169921875, 1604.447998046875, 2659.39208984375, 1604.447998046875,
2669.760009765625, 1599.2640380859375, 2680.128173828125, 1599.2640380859375,
2690.49609375, 1594.0799560546875, 2700.864013671875, 1594.0799560546875,
2711.232177734375, 1588.89599609375, 2721.60009765625, 1588.89599609375,
2731.968017578125, 1583.7120361328125, 2742.336181640625, 1583.7120361328125,
2752.7041015625, 1578.5279541015625, 2763.072021484375, 1578.5279541015625,
2773.440185546875, 1583.7120361328125, 2780.35205078125, 1588.89599609375,
2787.26416015625, 1594.0799560546875, 2797.632080078125, 1594.0799560546875,
2808.0, 1599.2640380859375, 2818.3681640625, 1599.2640380859375,
2828.736083984375, 1604.447998046875, 2839.10400390625, 1604.447998046875,
2849.47216796875, 1609.6319580078125, 2859.840087890625, 1609.6319580078125,
2870.2080078125, 1614.8160400390625, 2880.576171875, 1620.0, 2887.488037109375,
1625.1839599609375, 2894.400146484375, 1630.3680419921875, 2901.31201171875,
1635.552001953125, 2908.22412109375, 1640.7359619140625, 2915.135986328125,
1645.9200439453125, 2922.048095703125, 1651.10400390625, 2928.9599609375,
1656.2879638671875, 2939.328125, 1661.4720458984375, 2946.239990234375,
1666.656005859375, 2956.608154296875, 1671.8399658203125, 2963.52001953125,
1677.0240478515625, 2973.88818359375, 1682.2080078125, 2980.800048828125,
1687.3919677734375, 2987.712158203125, 1692.5760498046875, 2998.080078125,
1697.760009765625, 3004.9921875, 1702.9439697265625, 3011.904052734375,
1708.1280517578125, 3018.816162109375, 1713.31201171875, 3029.18408203125,
1718.4959716796875, 3036.09619140625, 1723.6800537109375, 3046.464111328125,
1728.864013671875, 3053.3759765625, 1734.0479736328125, 3063.744140625,
1734.0479736328125, 3070.656005859375, 1739.2320556640625, 3081.024169921875,
1739.2320556640625, 3087.93603515625, 1744.416015625, 3098.30419921875,
1749.5999755859375, 3105.216064453125, 1754.7840576171875, 3112.128173828125,
1759.968017578125, 3119.0400390625, 1765.1519775390625, 3125.9521484375,
1770.3360595703125, 3132.864013671875, 1775.52001953125, 3139.776123046875,

1780.7039794921875, 3146.68798828125, 1785.8880615234375, 3153.60009765625,
1791.072021484375, 3160.511962890625, 1791.072021484375, 3167.424072265625,
1796.2559814453125, 3177.7919921875, 1796.2559814453125, 3198.528076171875,
1791.072021484375, 3208.89599609375, 1791.072021484375, 3215.80810546875,
1785.8880615234375, 3222.719970703125, 1780.7039794921875, 3229.632080078125,
1770.3360595703125, 3236.544189453125, 1765.1519775390625, 3240.0,
1754.7840576171875, 3243.4560546875, 1744.416015625, 3243.4560546875,
1734.0479736328125, 3246.912109375, 1723.6800537109375, 3246.912109375,
1713.31201171875, 3250.3681640625, 1708.1280517578125, 3253.823974609375,
1702.9439697265625, 3260.736083984375, 1702.9439697265625, 3267.648193359375,
1708.1280517578125, 3281.47216796875, 1708.1280517578125, 3302.2080078125,
1708.1280517578125, 3322.944091796875, 1713.31201171875, 3340.22412109375,
1713.31201171875, 3357.504150390625, 1718.4959716796875, 3378.239990234375,
1718.4959716796875, 3402.43212890625, 1723.6800537109375, 3430.080078125,
1723.6800537109375, 3454.27197265625, 1355.615966796875, 3454.27197265625,
1355.615966796875, 3447.360107421875, 1355.615966796875, 3433.5361328125,
1350.4320068359375, 3419.712158203125, 1350.4320068359375, 3409.343994140625,
1345.248046875, 3398.97607421875, 1345.248046875, 3388.608154296875,
1340.06396484375, 3378.239990234375, 1340.06396484375, 3371.328125,
1334.8800048828125, 3364.416015625, 1334.8800048828125, 3357.504150390625,
1329.696044921875, 3350.592041015625, 1329.696044921875, 3343.68017578125,
1324.511962890625, 3336.76806640625, 1324.511962890625, 3329.856201171875,
1319.3280029296875, 3322.944091796875, 1319.3280029296875, 3316.031982421875,
1314.14404296875, 3309.1201171875, 1314.14404296875, 3302.2080078125,
1308.9599609375, 3291.840087890625, 1308.9599609375, 3281.47216796875,
1303.7760009765625, 3271.10400390625, 1303.7760009765625, 3264.192138671875,
1298.592041015625, 3253.823974609375, 1298.592041015625, 3246.912109375,
1293.407958984375, 3236.544189453125, 1293.407958984375, 3229.632080078125,
1288.2239990234375, 3222.719970703125, 1288.2239990234375, 3215.80810546875,
1283.0400390625, 3208.89599609375, 1283.0400390625, 3201.984130859375,
1283.0400390625, 3195.072021484375, 1277.85595703125, 3188.16015625,
1277.85595703125, 3181.248046875, 1272.6719970703125, 3174.336181640625,
1267.488037109375, 3167.424072265625, 1267.488037109375, 3160.511962890625,
1272.6719970703125, 3153.60009765625, 1277.85595703125, 3146.68798828125,
1277.85595703125, 3139.776123046875, 1283.0400390625, 3132.864013671875,
1283.0400390625, 3125.9521484375, 1288.2239990234375, 3119.0400390625,
1288.2239990234375, 3112.128173828125, 1293.407958984375, 3105.216064453125,
1293.407958984375, 3098.30419921875, 1298.592041015625, 3087.93603515625,
1298.592041015625, 3077.568115234375, 1298.592041015625, 3070.656005859375,
1298.592041015625, 3063.744140625, 1303.7760009765625, 3056.83203125,
1303.7760009765625, 3049.920166015625, 1308.9599609375, 3043.008056640625,
1308.9599609375, 3036.09619140625, 1308.9599609375, 3029.18408203125,
1308.9599609375, 3022.27197265625, 1314.14404296875, 3015.360107421875,
1314.14404296875, 3008.447998046875, 1319.3280029296875, 3001.5361328125,
1319.3280029296875, 2994.6240234375, 1324.511962890625, 2987.712158203125,
1324.511962890625, 2980.800048828125, 1329.696044921875, 2973.88818359375,
1329.696044921875, 2966.97607421875, 1334.8800048828125, 2960.06396484375,
1334.8800048828125, 2953.152099609375, 1340.06396484375, 2946.239990234375,

1340.06396484375, 2939.328125, 1340.06396484375, 2935.8720703125,
1345.248046875, 2928.9599609375, 1345.248046875, 2922.048095703125,
1350.4320068359375, 2915.135986328125, 1350.4320068359375, 2908.22412109375,
1350.4320068359375, 2901.31201171875, 1350.4320068359375, 2908.22412109375,
1355.615966796875, 2915.135986328125, 1355.615966796875, 2922.048095703125,
1355.615966796875, 2928.9599609375, 1360.800048828125, 2935.8720703125,
1360.800048828125, 2939.328125, 1365.9840087890625, 2946.239990234375,
1365.9840087890625, 2949.696044921875, 1371.16796875, 2956.608154296875,
1371.16796875, 2960.06396484375, 1376.35205078125, 2966.97607421875,
1381.5360107421875, 2973.88818359375, 1381.5360107421875, 2980.800048828125,
1386.719970703125, 2987.712158203125, 1386.719970703125, 2994.6240234375,
1365.9840087890625, 2953.152099609375, 1371.16796875, 2949.696044921875,
1376.35205078125, 2942.7841796875, 1381.5360107421875, 2935.8720703125,
1381.5360107421875, 2932.416015625, 1386.719970703125, 2925.504150390625,
1386.719970703125, 2922.048095703125, 1386.719970703125, 2915.135986328125,
1391.904052734375, 2908.22412109375, 1391.904052734375, 2901.31201171875,
1397.0880126953125, 2894.400146484375, 1397.0880126953125, 2887.488037109375,
1402.27197265625, 2880.576171875, 1402.27197265625, 2873.6640625,
1402.27197265625, 2866.751953125, 1402.27197265625, 2859.840087890625,
1402.27197265625, 2852.927978515625, 1397.0880126953125, 2846.01611328125,
1397.0880126953125, 2839.10400390625, 1391.904052734375, 2832.192138671875,
1391.904052734375, 2825.280029296875, 1386.719970703125, 2818.3681640625,
1386.719970703125, 2811.4560546875, 1386.719970703125, 2804.544189453125,
1381.5360107421875, 2797.632080078125, 1381.5360107421875, 2790.719970703125,
1376.35205078125, 2787.26416015625, 1371.16796875, 2780.35205078125,
1365.9840087890625, 2776.89599609375, 1360.800048828125, 2773.440185546875,
1355.615966796875, 2769.984130859375, 1350.4320068359375, 2766.528076171875,
1345.248046875, 2763.072021484375, 1340.06396484375, 2756.16015625,
1340.06396484375, 2752.7041015625, 1340.06396484375, 2745.7919921875,
1340.06396484375, 2742.336181640625, 1340.06396484375, 2735.424072265625,
1340.06396484375, 2728.511962890625, 1345.248046875, 2721.60009765625,
1345.248046875, 2714.68798828125, 1350.4320068359375, 2707.776123046875,
1350.4320068359375, 2700.864013671875, 1350.4320068359375, 2693.9521484375,
1350.4320068359375, 2690.49609375, 1350.4320068359375, 2683.583984375,
1350.4320068359375, 2676.672119140625, 1355.615966796875, 2669.760009765625,
1355.615966796875, 2662.84814453125, 1360.800048828125, 2655.93603515625,
1360.800048828125, 2652.47998046875, 1360.800048828125, 2645.568115234375,
1360.800048828125, 2638.656005859375, 1360.800048828125, 2631.744140625,
1360.800048828125, 2628.2880859375, 1360.800048828125, 2621.3759765625,
1360.800048828125, 2614.464111328125, 1360.800048828125, 2607.552001953125,
1360.800048828125, 2600.64013671875, 1360.800048828125, 2597.18408203125,
1360.800048828125, 2590.27197265625, 1360.800048828125, 2586.816162109375,
1365.9840087890625, 2579.904052734375, 1365.9840087890625, 2572.9921875,
1365.9840087890625, 2569.5361328125, 1365.9840087890625, 2562.6240234375,
1365.9840087890625, 2559.16796875, 1365.9840087890625, 2552.256103515625,
1365.9840087890625, 2548.800048828125, 1365.9840087890625, 2541.88818359375,
1365.9840087890625, 2538.43212890625, 1365.9840087890625, 2531.52001953125,
1365.9840087890625, 2528.06396484375, 1365.9840087890625, 2521.152099609375,

```
1365.9840087890625, 2517.696044921875, 1365.9840087890625, 2510.7841796875,
1365.9840087890625, 2507.328125, 1371.16796875, 2503.8720703125, 1371.16796875,
2500.416015625, 1371.16796875, 2496.9599609375, 1371.16796875,
2490.048095703125, 1371.16796875, 2483.135986328125, 1371.16796875,
2479.68017578125, 1371.16796875, 2476.22412109375, 1371.16796875,
2472.76806640625, 1371.16796875, 2469.31201171875, 1371.16796875,
2465.85595703125, 1371.16796875, 2462.400146484375, 1371.16796875,
2458.944091796875, 1371.16796875, 2455.488037109375, 1371.16796875,
2452.031982421875, 1371.16796875, 2455.488037109375, 1365.9840087890625,
2458.944091796875, 1365.9840087890625, 2462.400146484375, 1365.9840087890625,
2465.85595703125, 1365.9840087890625, 2469.31201171875, 1365.9840087890625,
2472.76806640625, 1365.9840087890625, 2476.22412109375, 1365.9840087890625,
2479.68017578125, 1365.9840087890625, 2483.135986328125, 1365.9840087890625,
2486.592041015625, 1365.9840087890625, 2490.048095703125, 1365.9840087890625,
2493.504150390625, 1365.9840087890625, 2496.9599609375, 1365.9840087890625,
2500.416015625]]], 'labels': ['']}}
```

[71]:

```
from PIL import Image, ImageDraw, ImageFont
import random
import numpy as np
colormap =␣
 ↪['blue','orange','green','purple','brown','pink','gray','olive','cyan','red',
          ␣
 ↪'lime','indigo','violet','aqua','magenta','coral','gold','tan','skyblue']
def draw_polygons(image, prediction, fill_mask=False):
    """
    Draws segmentation masks with polygons on an image.

    Parameters:
    - image_path: Path to the image file.
    - prediction: Dictionary containing 'polygons' and 'labels' keys.
                  'polygons' is a list of lists, each containing vertices of a␣
 ↪polygon.
                  'labels' is a list of labels corresponding to each polygon.
    - fill_mask: Boolean indicating whether to fill the polygons with color.
    """
    # Load the image

    draw = ImageDraw.Draw(image)


    # Set up scale factor if needed (use 1 if not scaling)
    scale = 1

    # Iterate over polygons and labels
    for polygons, label in zip(prediction['polygons'], prediction['labels']):
        color = random.choice(colormap)
```

```
        fill_color = random.choice(colormap) if fill_mask else None

        for _polygon in polygons:
            _polygon = np.array(_polygon).reshape(-1, 2)
            if len(_polygon) < 3:
                print('Invalid polygon:', _polygon)
                continue

            _polygon = (_polygon * scale).reshape(-1).tolist()

            # Draw the polygon
            if fill_mask:
                draw.polygon(_polygon, outline=color, fill=fill_color)
            else:
                draw.polygon(_polygon, outline=color)

            # Draw the label text
            draw.text((_polygon[0] + 8, _polygon[1] + 2), label, fill=color)

    # Save or display the image
    #image.show()  # Display the image
    display(image)
```

[72]:
```
output_image = copy.deepcopy(image)
draw_polygons(output_image, results['<REFERRING_EXPRESSION_SEGMENTATION>'],␣
↪fill_mask=True)
```

1.4.3 region to segmentation

with additional region as inputs, format is '<loc_x1><loc_y1><loc_x2><loc_y2>', [x1, y1, x2, y2] is the quantized corrdinates in [0, 999].

[73]:
```
task_prompt = '<REGION_TO_SEGMENTATION>'
results = run_example(task_prompt,␣
 ↪text_input="<loc_702><loc_575><loc_866><loc_772>")
print(results)
```

```
{'<REGION_TO_SEGMENTATION>': {'polygons': [[[3652.1279296875, 1995.840087890625,
3657.31201171875, 1992.384033203125, 3662.49609375, 1992.384033203125,
3667.679931640625, 1992.384033203125, 3672.864013671875, 1992.384033203125,
3678.048095703125, 1992.384033203125, 3683.23193359375, 1992.384033203125,
3688.416015625, 1992.384033203125, 3693.60009765625, 1992.384033203125,
3698.783935546875, 1992.384033203125, 3703.968017578125, 1992.384033203125,
3709.152099609375, 1992.384033203125, 3714.3359375, 1992.384033203125,
3719.52001953125, 1992.384033203125, 3724.7041015625, 1992.384033203125,
3729.887939453125, 1992.384033203125, 3735.072021484375, 1992.384033203125,
3740.256103515625, 1992.384033203125, 3745.43994140625, 1992.384033203125,
3750.6240234375, 1992.384033203125, 3755.80810546875, 1992.384033203125,
3760.991943359375, 1992.384033203125, 3766.176025390625, 1992.384033203125,
3771.360107421875, 1992.384033203125, 3776.5439453125, 1992.384033203125,
3781.72802734375, 1992.384033203125, 3786.912109375, 1992.384033203125,
3792.095947265625, 1992.384033203125, 3797.280029296875, 1992.384033203125,
3802.464111328125, 1992.384033203125, 3807.64794921875, 1992.384033203125,
3812.83203125, 1992.384033203125, 3818.01611328125, 1992.384033203125,
3823.199951171875, 1992.384033203125, 3828.384033203125, 1992.384033203125,
3833.568115234375, 1992.384033203125, 3838.751953125, 1992.384033203125,
3843.93603515625, 1992.384033203125, 3849.1201171875, 1992.384033203125,
3849.1201171875, 1995.840087890625, 3849.1201171875, 1999.2960205078125,
3849.1201171875, 2002.7520751953125, 3849.1201171875, 2006.2080078125,
3849.1201171875, 2009.6640625, 3849.1201171875, 2013.1199951171875,
3849.1201171875, 2016.5760498046875, 3849.1201171875, 2020.0321044921875,
3849.1201171875, 2023.488037109375, 3849.1201171875, 2026.944091796875,
3849.1201171875, 2030.4000244140625, 3849.1201171875, 2033.8560791015625,
3849.1201171875, 2037.31201171875, 3849.1201171875, 2040.76806640625,
3849.1201171875, 2044.2239990234375, 3849.1201171875, 2047.6800537109375,
3843.93603515625, 2047.6800537109375, 3838.751953125, 2047.6800537109375,
3833.568115234375, 2047.6800537109375, 3828.384033203125, 2047.6800537109375,
3823.199951171875, 2047.6800537109375, 3818.01611328125, 2047.6800537109375,
3812.83203125, 2047.6800537109375, 3807.64794921875, 2047.6800537109375,
3802.464111328125, 2047.6800537109375, 3797.280029296875, 2047.6800537109375,
3792.095947265625, 2047.6800537109375, 3786.912109375, 2047.6800537109375,
3781.72802734375, 2047.6800537109375, 3776.5439453125, 2051.135986328125,
```

3771.360107421875, 2051.135986328125, 3766.176025390625, 2051.135986328125,
3760.991943359375, 2051.135986328125, 3755.80810546875, 2054.592041015625,
3750.6240234375, 2054.592041015625, 3745.43994140625, 2054.592041015625,
3740.256103515625, 2058.048095703125, 3740.256103515625, 2058.048095703125,
3735.072021484375, 2061.504150390625, 3735.072021484375, 2061.504150390625,
3729.887939453125, 2064.9599609375, 3729.887939453125, 2068.416015625,
3729.887939453125, 2071.8720703125, 3729.887939453125, 2075.328125,
3729.887939453125, 2078.783935546875, 3729.887939453125, 2082.239990234375,
3729.887939453125, 2085.696044921875, 3729.887939453125, 2089.152099609375,
3729.887939453125, 2092.608154296875, 3729.887939453125, 2096.06396484375,
3729.887939453125, 2099.52001953125, 3729.887939453125, 2102.97607421875,
3729.887939453125, 2106.43212890625, 3729.887939453125, 2109.887939453125,
3729.887939453125, 2113.343994140625, 3729.887939453125, 2116.800048828125,
3729.887939453125, 2120.256103515625, 3729.887939453125, 2123.712158203125,
3729.887939453125, 2127.16796875, 3729.887939453125, 2130.6240234375,
3729.887939453125, 2134.080078125, 3729.887939453125, 2137.5361328125,
3729.887939453125, 2140.991943359375, 3729.887939453125, 2144.447998046875,
3729.887939453125, 2147.904052734375, 3729.887939453125, 2151.360107421875,
3729.887939453125, 2154.816162109375, 3729.887939453125, 2158.27197265625,
3729.887939453125, 2161.72802734375, 3729.887939453125, 2165.18408203125,
3729.887939453125, 2168.64013671875, 3729.887939453125, 2172.095947265625,
3729.887939453125, 2175.552001953125, 3729.887939453125, 2179.008056640625,
3729.887939453125, 2182.464111328125, 3729.887939453125, 2185.920166015625,
3729.887939453125, 2189.3759765625, 3729.887939453125, 2192.83203125,
3729.887939453125, 2196.2880859375, 3729.887939453125, 2199.744140625,
3729.887939453125, 2203.199951171875, 3729.887939453125, 2206.656005859375,
3729.887939453125, 2210.112060546875, 3729.887939453125, 2213.568115234375,
3729.887939453125, 2217.024169921875, 3729.887939453125, 2220.47998046875,
3729.887939453125, 2223.93603515625, 3729.887939453125, 2227.39208984375,
3729.887939453125, 2230.84814453125, 3729.887939453125, 2234.303955078125,
3729.887939453125, 2237.760009765625, 3729.887939453125, 2241.216064453125,
3729.887939453125, 2244.672119140625, 3729.887939453125, 2248.128173828125,
3729.887939453125, 2251.583984375, 3729.887939453125, 2255.0400390625,
3729.887939453125, 2258.49609375, 3729.887939453125, 2261.9521484375,
3729.887939453125, 2265.407958984375, 3729.887939453125, 2268.864013671875,
3729.887939453125, 2272.320068359375, 3729.887939453125, 2275.776123046875,
3729.887939453125, 2279.232177734375, 3729.887939453125, 2282.68798828125,
3729.887939453125, 2286.14404296875, 3729.887939453125, 2289.60009765625,
3729.887939453125, 2293.0615234375, 3729.887939453125, 2296.511962890625,
3729.887939453125, 2299.968017578125, 3729.887939453125, 2303.424072265625,
3729.887939453125, 2306.880126953125, 3729.887939453125, 2310.336181640625,
3729.887939453125, 2313.7919921875, 3729.887939453125, 2317.248046875,
3729.887939453125, 2320.7041015625, 3729.887939453125, 2324.16015625,
3729.887939453125, 2327.615966796875, 3729.887939453125, 2331.072021484375,
3729.887939453125, 2334.528076171875, 3729.887939453125, 2337.984130859375,
3729.887939453125, 2341.43994140625, 3729.887939453125, 2344.89599609375,
3729.887939453125, 2348.35205078125, 3729.887939453125, 2351.80810546875,
3729.887939453125, 2355.26416015625, 3729.887939453125, 2358.719970703125,

3729.887939453125, 2362.176025390625, 3729.887939453125, 2365.632080078125,
3729.887939453125, 2369.088134765625, 3729.887939453125, 2372.5439453125,
3729.887939453125, 2376.0, 3729.887939453125, 2379.4560546875,
3729.887939453125, 2382.912109375, 3729.887939453125, 2386.3681640625,
3729.887939453125, 2389.823974609375, 3729.887939453125, 2393.280029296875,
3729.887939453125, 2396.736083984375, 3729.887939453125, 2400.192138671875,
3729.887939453125, 2403.64794921875, 3729.887939453125, 2407.10400390625,
3729.887939453125, 2410.56005859375, 3729.887939453125, 2414.01611328125,
3729.887939453125, 2417.47216796875, 3729.887939453125, 2420.927978515625,
3729.887939453125, 2424.384033203125, 3729.887939453125, 2427.840087890625,
3729.887939453125, 2431.296142578125, 3729.887939453125, 2434.751953125,
3729.887939453125, 2438.2080078125, 3729.887939453125, 2441.6640625,
3729.887939453125, 2445.1201171875, 3729.887939453125, 2448.576171875,
3729.887939453125, 2452.031982421875, 3729.887939453125, 2455.488037109375,
3729.887939453125, 2458.944091796875, 3729.887939453125, 2462.400146484375,
3729.887939453125, 2465.85595703125, 3729.887939453125, 2469.31201171875,
3729.887939453125, 2472.76806640625, 3729.887939453125, 2476.22412109375,
3729.887939453125, 2479.68017578125, 3729.887939453125, 2483.135986328125,
3729.887939453125, 2486.592041015625, 3729.887939453125, 2490.048095703125,
3729.887939453125, 2493.504150390625, 3729.887939453125, 2496.9599609375,
3729.887939453125, 2500.416015625, 3729.887939453125, 2503.8720703125,
3729.887939453125, 2507.328125, 3729.887939453125, 2510.7841796875,
3729.887939453125, 2514.239990234375, 3729.887939453125, 2517.696044921875,
3729.887939453125, 2521.152099609375, 3729.887939453125, 2524.608154296875,
3729.887939453125, 2528.06396484375, 3729.887939453125, 2531.52001953125,
3729.887939453125, 2534.97607421875, 3729.887939453125, 2538.43212890625,
3729.887939453125, 2541.88818359375, 3729.887939453125, 2545.343994140625,
3729.887939453125, 2548.800048828125, 3729.887939453125, 2552.256103515625,
3729.887939453125, 2555.712158203125, 3729.887939453125, 2559.16796875,
3729.887939453125, 2562.6240234375, 3729.887939453125, 2566.080078125,
3729.887939453125, 2569.5361328125, 3729.887939453125, 2572.9921875,
3729.887939453125, 2576.447998046875, 3729.887939453125, 2579.904052734375,
3729.887939453125, 2583.360107421875, 3729.887939453125, 2586.816162109375,
3729.887939453125, 2590.27197265625, 3729.887939453125, 2593.72802734375,
3729.887939453125, 2597.18408203125, 3729.887939453125, 2597.18408203125,
3724.7041015625, 2600.64013671875, 3724.7041015625, 2604.095947265625,
3724.7041015625, 2607.552001953125, 3724.7041015625, 2611.008056640625,
3724.7041015625, 2614.464111328125, 3724.7041015625, 2617.920166015625,
3724.7041015625, 2621.3759765625, 3724.7041015625, 2624.83203125,
3724.7041015625, 2628.2880859375, 3724.7041015625, 2631.744140625,
3724.7041015625, 2635.199951171875, 3724.7041015625, 2638.656005859375,
3724.7041015625, 2642.112060546875, 3724.7041015625, 2645.568115234375,
3724.7041015625, 2649.024169921875, 3724.7041015625, 2652.47998046875,
3724.7041015625, 2655.93603515625, 3724.7041015625, 2659.39208984375,
3724.7041015625, 2662.84814453125, 3724.7041015625, 2666.303955078125,
3724.7041015625, 2669.760009765625, 3724.7041015625, 2673.216064453125,
3724.7041015625, 2676.672119140625, 3724.7041015625, 2680.128173828125,
3724.7041015625, 2683.583984375, 3724.7041015625, 2687.0400390625,

3724.7041015625, 2690.49609375, 3724.7041015625, 2693.9521484375,
3724.7041015625, 2697.407958984375, 3724.7041015625, 2700.864013671875,
3724.7041015625, 2704.320068359375, 3724.7041015625, 2707.776123046875,
3724.7041015625, 2711.232177734375, 3724.7041015625, 2714.68798828125,
3724.7041015625, 2718.14404296875, 3724.7041015625, 2721.60009765625,
3724.7041015625, 2725.05615234375, 3724.7041015625, 2728.511962890625,
3724.7041015625, 2731.968017578125, 3724.7041015625, 2735.424072265625,
3724.7041015625, 2738.880126953125, 3724.7041015625, 2742.336181640625,
3724.7041015625, 2745.7919921875, 3724.7041015625, 2749.248046875,
3724.7041015625, 2752.7041015625, 3724.7041015625, 2756.16015625,
3724.7041015625, 2759.615966796875, 3724.7041015625, 2763.072021484375,
3724.7041015625, 2766.528076171875, 3724.7041015625, 2769.984130859375,
3724.7041015625, 2773.440185546875, 3724.7041015625, 2776.89599609375,
3724.7041015625, 2780.35205078125, 3724.7041015625, 2783.80810546875,
3724.7041015625, 2787.26416015625, 3724.7041015625, 2790.719970703125,
3724.7041015625, 2794.176025390625, 3724.7041015625, 2797.632080078125,
3724.7041015625, 2801.088134765625, 3724.7041015625, 2804.544189453125,
3724.7041015625, 2808.0, 3724.7041015625, 2811.4560546875, 3724.7041015625,
2814.912109375, 3724.7041015625, 2818.3681640625, 3724.7041015625,
2821.823974609375, 3724.7041015625, 2825.280029296875, 3724.7041015625,
2828.736083984375, 3724.7041015625, 2832.192138671875, 3724.7041015625,
2835.648193359375, 3724.7041015625, 2839.10400390625, 3724.7041015625,
2842.56005859375, 3724.7041015625, 2846.01611328125, 3724.7041015625,
2849.47216796875, 3724.7041015625, 2852.927978515625, 3724.7041015625,
2856.384033203125, 3724.7041015625, 2859.840087890625, 3724.7041015625,
2863.296142578125, 3724.7041015625, 2866.751953125, 3724.7041015625,
2870.2080078125, 3724.7041015625, 2873.6640625, 3724.7041015625,
2877.1201171875, 3724.7041015625, 2880.576171875, 3724.7041015625,
2884.031982421875, 3724.7041015625, 2887.488037109375, 3724.7041015625,
2890.944091796875, 3724.7041015625, 2894.400146484375, 3724.7041015625,
2897.85595703125, 3724.7041015625, 2901.31201171875, 3724.7041015625,
2904.76806640625, 3724.7041015625, 2908.22412109375, 3724.7041015625,
2911.68017578125, 3724.7041015625, 2915.135986328125, 3724.7041015625,
2918.592041015625, 3724.7041015625, 2922.048095703125, 3724.7041015625,
2925.504150390625, 3724.7041015625, 2928.9599609375, 3724.7041015625,
2932.416015625, 3724.7041015625, 2935.8720703125, 3724.7041015625, 2939.328125,
3724.7041015625, 2942.7841796875, 3724.7041015625, 2946.239990234375,
3724.7041015625, 2949.696044921875, 3724.7041015625, 2953.152099609375,
3724.7041015625, 2956.608154296875, 3724.7041015625, 2960.06396484375,
3724.7041015625, 2963.52001953125, 3724.7041015625, 2966.97607421875,
3724.7041015625, 2970.43212890625, 3724.7041015625, 2973.88818359375,
3724.7041015625, 2977.343994140625, 3724.7041015625, 2980.800048828125,
3724.7041015625, 2984.256103515625, 3724.7041015625, 2987.712158203125,
3724.7041015625, 2991.16796875, 3724.7041015625, 2994.6240234375,
3724.7041015625, 2998.080078125, 3724.7041015625, 3001.5361328125,
3724.7041015625, 3004.9921875, 3724.7041015625, 3008.447998046875,
3724.7041015625, 3011.904052734375, 3724.7041015625, 3015.360107421875,
3724.7041015625, 3018.816162109375, 3724.7041015625, 3022.27197265625,

3724.7041015625, 3025.72802734375, 3724.7041015625, 3029.18408203125,
3724.7041015625, 3032.64013671875, 3724.7041015625, 3036.09619140625,
3724.7041015625, 3039.552001953125, 3724.7041015625, 3043.008056640625,
3724.7041015625, 3046.464111328125, 3724.7041015625, 3049.920166015625,
3724.7041015625, 3053.3759765625, 3724.7041015625, 3056.83203125,
3724.7041015625, 3060.2880859375, 3724.7041015625, 3063.744140625,
3724.7041015625, 3067.2001953125, 3724.7041015625, 3070.656005859375,
3724.7041015625, 3074.112060546875, 3724.7041015625, 3077.568115234375,
3724.7041015625, 3081.024169921875, 3724.7041015625, 3084.47998046875,
3724.7041015625, 3087.93603515625, 3724.7041015625, 3091.39208984375,
3724.7041015625, 3094.84814453125, 3724.7041015625, 3098.30419921875,
3724.7041015625, 3101.760009765625, 3724.7041015625, 3105.216064453125,
3724.7041015625, 3108.672119140625, 3724.7041015625, 3112.128173828125,
3724.7041015625, 3115.583984375, 3724.7041015625, 3119.0400390625,
3724.7041015625, 3122.49609375, 3724.7041015625, 3125.9521484375,
3724.7041015625, 3129.408203125, 3724.7041015625, 3132.864013671875,
3724.7041015625, 3136.320068359375, 3724.7041015625, 3139.776123046875,
3724.7041015625, 3143.232177734375, 3724.7041015625, 3146.68798828125,
3724.7041015625, 3150.14404296875, 3724.7041015625, 3153.60009765625,
3724.7041015625, 3157.05615234375, 3724.7041015625, 3160.511962890625,
3724.7041015625, 3163.968017578125, 3724.7041015625, 3167.424072265625,
3724.7041015625, 3170.880126953125, 3724.7041015625, 3174.336181640625,
3724.7041015625, 3177.7919921875, 3724.7041015625, 3181.248046875,
3724.7041015625, 3184.7041015625, 3724.7041015625, 3188.16015625,
3724.7041015625, 3191.615966796875, 3724.7041015625, 3195.072021484375,
3724.7041015625, 3198.528076171875, 3724.7041015625, 3201.984130859375,
3724.7041015625, 3205.440185546875, 3724.7041015625, 3208.89599609375,
3724.7041015625, 3212.35205078125, 3724.7041015625, 3215.80810546875,
3724.7041015625, 3219.26416015625, 3724.7041015625, 3222.719970703125,
3724.7041015625, 3226.176025390625, 3724.7041015625, 3229.632080078125,
3724.7041015625, 3233.088134765625, 3724.7041015625, 3236.544189453125,
3724.7041015625, 3240.0, 3724.7041015625, 3243.4560546875, 3724.7041015625,
3246.912109375, 3724.7041015625, 3250.3681640625, 3724.7041015625,
3253.823974609375, 3724.7041015625, 3257.280029296875, 3724.7041015625,
3260.736083984375, 3724.7041015625, 3264.192138671875, 3724.7041015625,
3267.648193359375, 3724.7041015625, 3271.10400390625, 3724.7041015625,
3274.56005859375, 3724.7041015625, 3278.01611328125, 3724.7041015625,
3281.47216796875, 3724.7041015625, 3284.927978515625, 3724.7041015625,
3288.384033203125, 3724.7041015625, 3291.840087890625, 3724.7041015625,
3295.296142578125, 3724.7041015625, 3298.752197265625, 3724.7041015625,
3302.2080078125, 3724.7041015625, 3305.6640625, 3724.7041015625,
3309.1201171875, 3724.7041015625, 3312.576171875, 3724.7041015625,
3316.031982421875, 3724.7041015625, 3319.488037109375, 3724.7041015625,
3322.944091796875, 3724.7041015625, 3326.400146484375, 3724.7041015625,
3329.856201171875, 3724.7041015625, 3333.31201171875, 3724.7041015625,
3336.76806640625, 3724.7041015625, 3340.22412109375, 3724.7041015625,
3343.68017578125, 3724.7041015625, 3347.135986328125, 3724.7041015625,
3350.592041015625, 3724.7041015625, 3354.048095703125, 3724.7041015625,

```
3357.504150390625, 3724.7041015625, 3360.960205078125, 3724.7041015625,
3364.416015625, 3724.7041015625, 3367.8720703125, 3724.7041015625, 3371.328125,
3724.7041015625, 3374.7841796875, 3724.7041015625, 3378.239990234375,
3724.7041015625, 3381.696044921875, 3724.7041015625, 3385.152099609375,
3724.7041015625, 3388.608154296875, 3724.7041015625, 3392.064208984375,
3724.7041015625, 3395.52001953125, 3724.7041015625, 3398.97607421875,
3724.7041015625, 3402.43212890625, 3724.7041015625, 3405.88818359375,
3724.7041015625, 3409.343994140625, 3724.7041015625, 3412.800048828125,
3724.7041015625, 3416.256103515625, 3724.7041015625, 3419.712158203125,
3724.7041015625, 3423.16796875, 3724.7041015625, 3426.6240234375,
3724.7041015625, 3430.080078125, 3724.7041015625, 3433.5361328125,
3724.7041015625, 3436.9921875, 3724.7041015625, 3440.447998046875,
3724.7041015625, 3443.904052734375, 3724.7041015625, 3447.360107421875,
3724.7041015625, 3450.816162109375, 3724.7041015625, 3454.27197265625,
3724.7041015625, 3454.27197265625, 3719.52001953125, 3454.27197265625,
3719.52001953125, 3450.816162109375, 3719.52001953125, 3447.360107421875,
3719.52001953125, 3447.360107421875, 3724.7041015625, 3447.360107421875,
3729.887939453125, 3447.360107421875, 3729.887939453125, 3450.816162109375,
3729.887939453125, 3454.27197265625, 3729.887939453125, 3454.27197265625,
3735.072021484375, 3454.27197265625, 3740.256103515625, 3454.27197265625,
3745.43994140625, 3454.27197265625, 3750.6240234375, 3454.27197265625,
3755.80810546875, 3454.27197265625, 3760.991943359375, 3454.27197265625,
3766.176025390625, 3454.27197265625, 3771.360107421875, 3454.27197265625,
3776.5439453125, 3454.27197265625, 3781.72802734375, 3454.27197265625,
3786.912109375, 3454.27197265625, 3792.095947265625, 3454.27197265625,
3797.280029296875, 3454.27197265625, 3802.464111328125, 3454.27197265625,
3807.64794921875, 3454.27197265625, 3812.83203125, 3454.27197265625,
3818.01611328125, 3454.27197265625, 3823.199951171875, 3454.27197265625]]],
'labels': ['']}}
```

[42]:
```
output_image = copy.deepcopy(image)
draw_polygons(output_image, results['<REGION_TO_SEGMENTATION>'], fill_mask=True)
```

1.4.4 Open vocabulary detection

open vocabulary detection can detect both objects and ocr texts.

results format:

{ '<OPEN_VOCABULARY_DETECTION>': {'bboxes': [[x1, y1, x2, y2], [x1, y1, x2, y2], ...]], 'bboxes_labels': ['label_1', 'label_2', ..], 'polygons': [[[x1, y1, x2, y2, ..., xn, yn], [x1, y1, ..., xn, yn]], ...], 'polygons_labels': ['label_1', 'label_2', ...] }}

[ ]:
```
task_prompt = '<OPEN_VOCABULARY_DETECTION>'
results = run_example(task_prompt, text_input="a green car")
print(results)
```

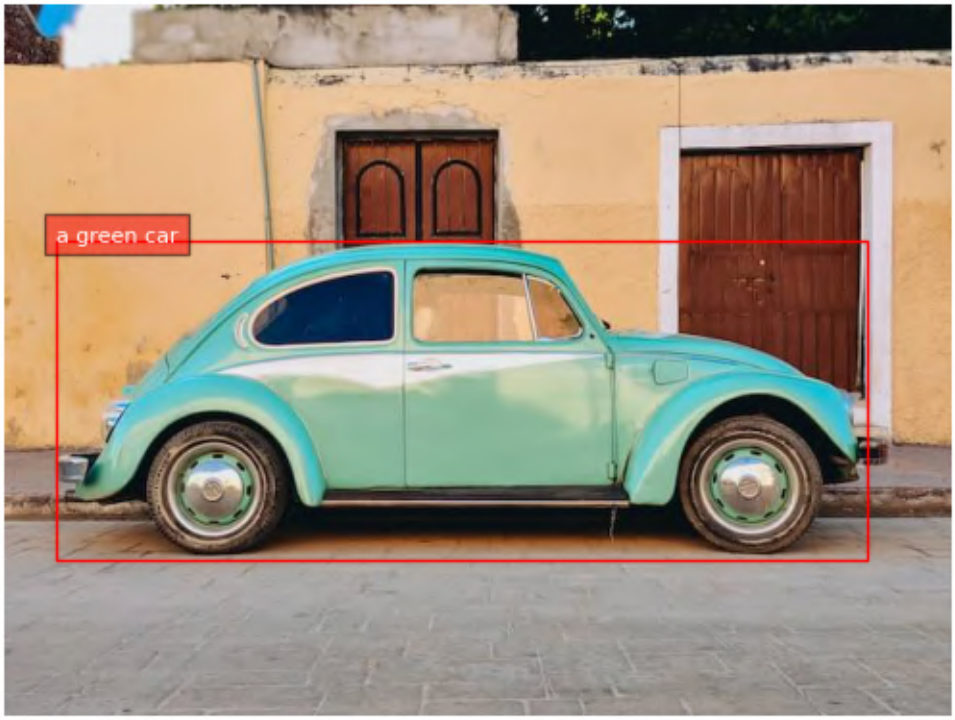
```
{'<OPEN_VOCABULARY_DETECTION>': {'bboxes': [[34.23999786376953,
158.63999938964844, 582.0800170898438, 374.1600036621094]], 'bboxes_labels': ['a
green car'], 'polygons': [], 'polygons_labels': []}}
```

[ ]:
```
def convert_to_od_format(data):
    """
    Converts a dictionary with 'bboxes' and 'bboxes_labels' into a dictionary␣
 ↪with separate 'bboxes' and 'labels' keys.

    Parameters:
```

```
    - data: The input dictionary with 'bboxes', 'bboxes_labels', 'polygons',␣
↪and 'polygons_labels' keys.

    Returns:
    - A dictionary with 'bboxes' and 'labels' keys formatted for object␣
↪detection results.
    """
    # Extract bounding boxes and labels
    bboxes = data.get('bboxes', [])
    labels = data.get('bboxes_labels', [])

    # Construct the output format
    od_results = {
        'bboxes': bboxes,
        'labels': labels
    }

    return od_results
```

[ ]:
```
bbox_results  = convert_to_od_format(results['<OPEN_VOCABULARY_DETECTION>'])
```

[ ]:
```
plot_bbox(image, bbox_results)
```



1.4.5 region to texts

[ ]:
```
task_prompt = '<REGION_TO_CATEGORY>'
results = run_example(task_prompt,␣
 ↪text_input="<loc_52><loc_332><loc_932><loc_774>")
print(results)
```

```
{'<REGION_TO_CATEGORY>': 'car<loc_52><loc_332><loc_932><loc_774>'}
```

[ ]:
```
task_prompt = '<REGION_TO_DESCRIPTION>'
results = run_example(task_prompt,␣
 ↪text_input="<loc_52><loc_332><loc_932><loc_774>")
print(results)
```

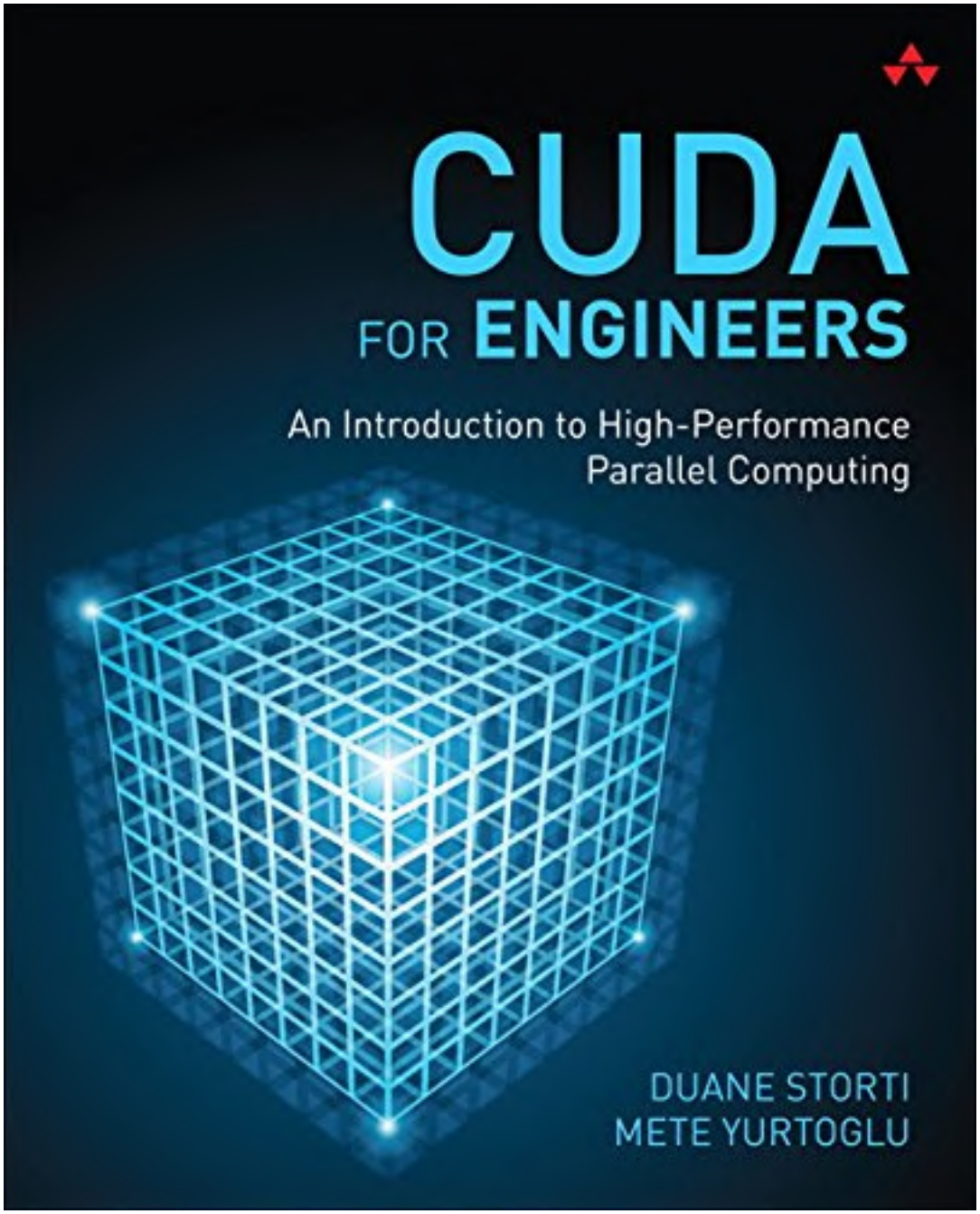
```
{'<REGION_TO_DESCRIPTION>': 'turquoise Volkswagen
Beetle<loc_52><loc_332><loc_932><loc_774>'}
```

1.5 ocr related tasks

[74]:
```
url = "http://ecx.images-amazon.com/images/I/51UUzBDAMsL.jpg?download=true"
image = Image.open(requests.get(url, stream=True).raw).convert('RGB')
```

[75]:
```
image
```

[75]:



[76]:
```
task_prompt = '<OCR>'
run_example(task_prompt)
```

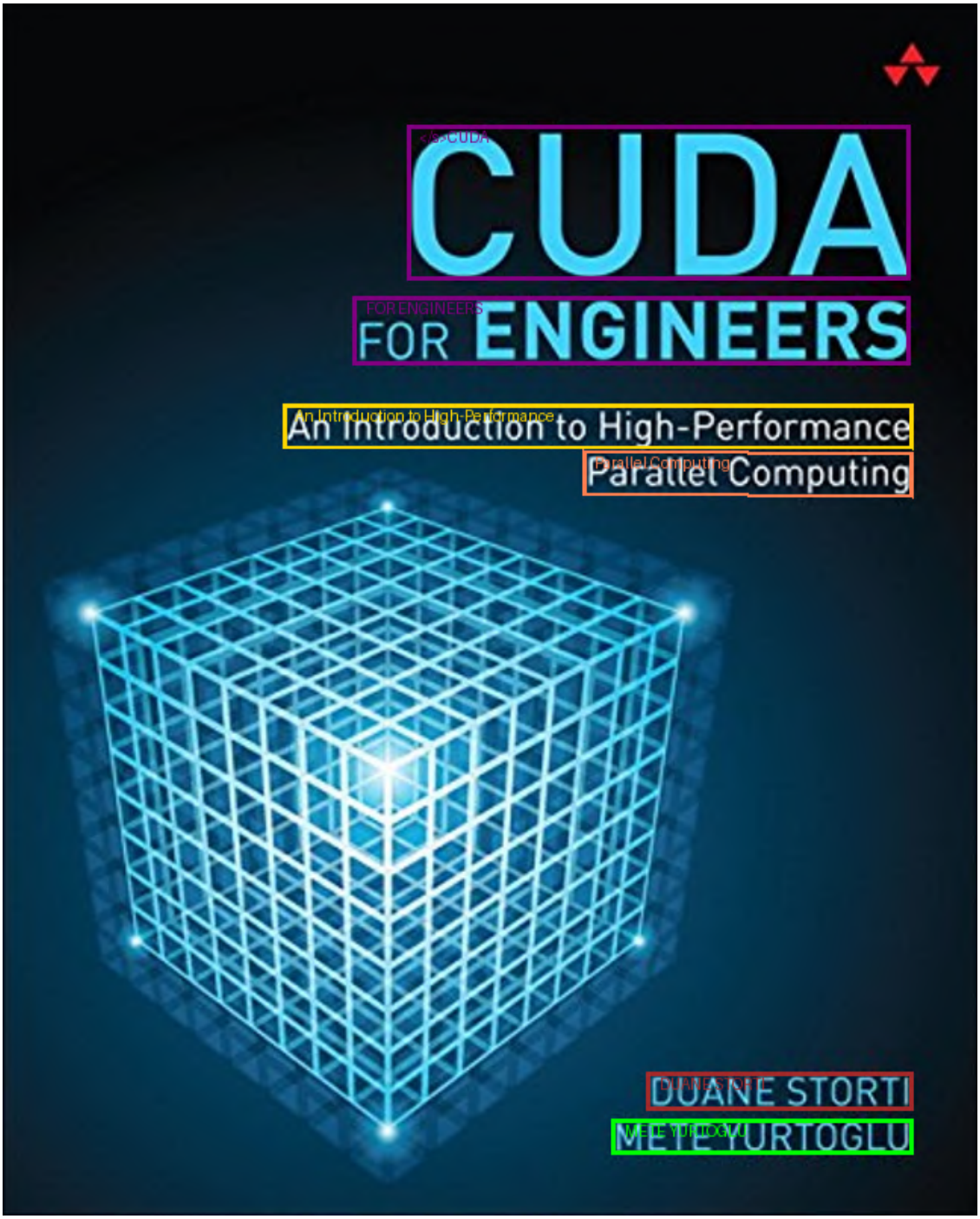
[76]:
```
{'<OCR>': 'CUDAFOR ENGINEERSAn Introduction to High-PerformanceParallel
ComputingDUANE STORTIMETE YURTOGLU'}
```

[79]:
```
task_prompt = '<OCR_WITH_REGION>'
results = run_example(task_prompt)
print(results)
# ocr results format
# {'OCR_WITH_REGION': {'quad_boxes': [[x1, y1, x2, y2, x3, y3, x4, y4], ...],␣
 ↪'labels': ['text1', ...]}}
```

```
{'<OCR_WITH_REGION>': {'quad_boxes': [[167.0435028076172, 50.25,
374.9914855957031, 50.25, 374.9914855957031, 114.25, 167.0435028076172, 114.25],
[144.8784942626953, 120.75, 374.9914855957031, 120.75, 374.9914855957031,
149.25, 144.8784942626953, 149.25], [115.86249542236328, 165.25,
376.20050048828125, 165.25, 376.20050048828125, 184.25, 115.86249542236328,
183.25], [239.58349609375, 184.25, 376.20050048828125, 185.25,
376.20050048828125, 203.75, 239.58349609375, 202.75], [266.1814880371094,
440.75, 376.20050048828125, 440.75, 376.20050048828125, 456.25,
266.1814880371094, 456.25], [251.67349243164062, 460.25, 376.20050048828125,
460.25, 376.20050048828125, 474.75, 251.67349243164062, 474.75]], 'labels':
['</s>CUDA', 'FOR ENGINEERS', 'An Introduction to High-Performance', 'Parallel
Computing', 'DUANE STORTI', 'METE YURTOGLU']}}
```

[77]:
```
def draw_ocr_bboxes(image, prediction, scale=1):
    draw = ImageDraw.Draw(image)
    bboxes, labels = prediction['quad_boxes'], prediction['labels']
    for box, label in zip(bboxes, labels):
        color = random.choice(colormap)
        new_box = (np.array(box) * scale).tolist()
        draw.polygon(new_box, width=3, outline=color)
        draw.text((new_box[0]+8, new_box[1]+2),
                    "{}".format(label),
                    align="right",

                    fill=color)

    display(image)
```

[80]:
```
output_image = copy.deepcopy(image)
w, h = output_image.size
scale = 800 / max(w, h)
new_output_image = output_image.resize((int(w * scale), int(h * scale)))
draw_ocr_bboxes(new_output_image, results['<OCR_WITH_REGION>'], scale=scale)
```



1.6 Cascaded tasks

1.6.1 Caption + Phrase Grounding

results format:

{ '<CAPTION': pure_text, {'<CAPTION_TO_PHRASE_GROUNDING>': {'bboxes': [[x1, y1,

x2, y2], ...], 'labels': ['', '', ...]}} }

[ ]:
```
url = "https://huggingface.co/datasets/huggingface/documentation-images/resolve/
↪main/transformers/tasks/car.jpg?download=true"
image = Image.open(requests.get(url, stream=True).raw)
```

[ ]:
```
task_prompt = '<CAPTION>'
results = run_example(task_prompt)
text_input = results[task_prompt]
task_prompt = '<CAPTION_TO_PHRASE_GROUNDING>'
results = run_example(task_prompt, text_input)
results['<CAPTION>'] = text_input
```

[ ]:
```
results
```

[ ]:
```
{'<CAPTION_TO_PHRASE_GROUNDING>': {'bboxes': [[34.23999786376953,
    159.1199951171875,
    582.0800170898438,
    374.6399841308594],
   [1.5999999046325684,
    4.079999923706055,
    639.0399780273438,
    305.03997802734375]],
  'labels': ['A green car', 'a yellow building']},
 '<CAPTION>': 'A green car parked in front of a yellow building.'}
```

[ ]:
```
plot_bbox(image, results['<CAPTION_TO_PHRASE_GROUNDING>'])
```

1.6.2 Detailed Caption + Phrase Grounding

results format:

{ '<DETAILED_CAPTION': pure_text, {'<CAPTION_TO_PHRASE_GROUNDING>': {'bboxes': [[x1, y1, x2, y2], ...], 'labels': ['', '', ...]}} }

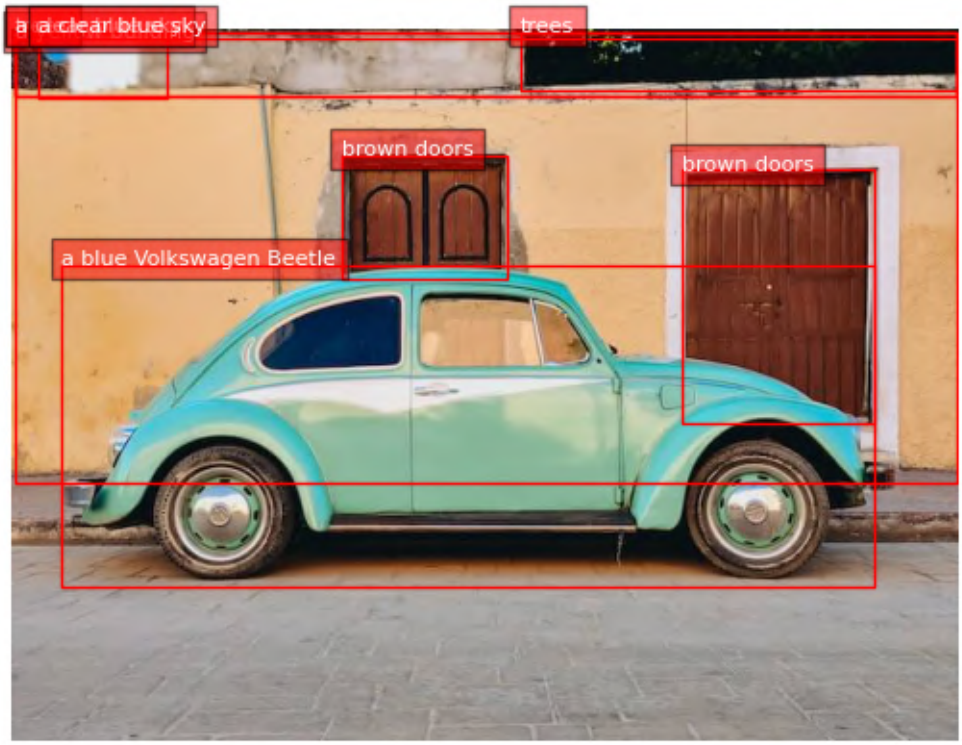
[ ]:
```
task_prompt = '<DETAILED_CAPTION>'
results = run_example(task_prompt)
text_input = results[task_prompt]
task_prompt = '<CAPTION_TO_PHRASE_GROUNDING>'
results = run_example(task_prompt, text_input)
results['<DETAILED_CAPTION>'] = text_input
```

[ ]:
```
results
```

[ ]:
```
{'<CAPTION_TO_PHRASE_GROUNDING>': {'bboxes': [[33.599998474121094,
    158.63999938964844,
    582.719970703125,
    375.1199951171875],
   [1.5999999046325684, 5.039999961853027, 639.0399780273438, 306.0],
   [452.79998779296875,
    94.31999969482422,
    582.0800170898438,
```

```
      265.67999267578125],
     [222.39999389648438,
      84.23999786376953,
      335.03997802734375,
      167.27999877929688],
     [2.240000009536743,
      1.1999999284744263,
      639.0399780273438,
      43.91999816894531],
     [343.3599853515625,
      1.1999999284744263,
      639.0399780273438,
      40.55999755859375],
     [2.240000009536743,
      1.1999999284744263,
      638.3999633789062,
      43.91999816894531],
     [18.239999771118164,
      1.1999999284744263,
      104.63999938964844,
      45.36000061035156]],
    'labels': ['a blue Volkswagen Beetle',
     'a yellow building',
     'brown doors',
     'brown doors',
     'trees',
     'trees',
     'a clear blue sky',
     'a clear blue sky']},
   '<DETAILED_CAPTION>': 'The image shows a blue Volkswagen Beetle parked in front
  of a yellow building with two brown doors, surrounded by trees and a clear blue
  sky.'}
```

[ ]:
```
plot_bbox(image, results['<CAPTION_TO_PHRASE_GROUNDING>'])
```



1.6.3 More Detailed Caption + Phrase Grounding

results format:

{ '<MORE_DETAILED_CAPTION': pure_text, {'<CAPTION_TO_PHRASE_GROUNDING>': {'bboxes': [[x1, y1, x2, y2], ...], 'labels': ['',", ...]}} }

[ ]:
```
task_prompt = '<MORE_DETAILED_CAPTION>'
results = run_example(task_prompt)
text_input = results[task_prompt]
task_prompt = '<CAPTION_TO_PHRASE_GROUNDING>'
results = run_example(task_prompt, text_input)
results['<MORE_DETAILED_CAPTION>'] = text_input
```

[ ]:
```
results
```

[ ]:
```
{'<CAPTION_TO_PHRASE_GROUNDING>': {'bboxes': [[35.52000045776367,
    157.67999267578125,
    581.4400024414062,
    372.7200012207031],
   [1.5999999046325684,
    340.0799865722656,
    639.0399780273438,
    479.2799987792969],
```

```
 [454.0799865722656,
  93.83999633789062,
  580.7999877929688,
  263.2799987792969],
 [223.67999267578125,
  84.23999786376953,
  333.7599792480469,
  163.9199981689453],
 [36.79999923706055,
  157.67999267578125,
  580.1599731445312,
  372.239990234375],
 [454.0799865722656,
  93.83999633789062,
  581.4400024414062,
  263.2799987792969],
 [223.67999267578125,
  84.72000122070312,
  333.7599792480469,
  163.9199981689453],
 [164.16000366210938,
  177.36000061035156,
  265.2799987792969,
  232.0800018310547],
 [2.879999876022339,
  4.559999942779541,
  639.0399780273438,
  301.1999816894531],
 [19.520000457763672, 1.1999999284744263, 639.0399780273438, 42.0],
 [20.15999984741211,
  0.7199999690055847,
  102.07999420166016,
  42.959999084472656],
 [345.91998291015625,
  1.1999999284744263,
  639.0399780273438,
  37.68000030517578]],
'labels': ['a vintage Volkswagen Beetle car',
 'a cobblestone street',
 'wooden doors',
 'wooden doors',
 'The car',
 'doors',
 'doors',
 'small window',
 'The building',
 'The sky',
```

```
    'The sky',
    'trees']},
 '<MORE_DETAILED_CAPTION>': 'The image shows a vintage Volkswagen Beetle car
parked on a cobblestone street in front of a yellow building with two wooden
doors. The car is painted in a bright turquoise color and has a sleek,
streamlined design. It has two doors on either side of the car, one on top of
the other, and a small window on the front. The building appears to be old and
dilapidated, with peeling paint and crumbling walls. The sky is blue and there
are trees in the background.'}
```

[ ]:
```
plot_bbox(image, results['<CAPTION_TO_PHRASE_GROUNDING>'])
```