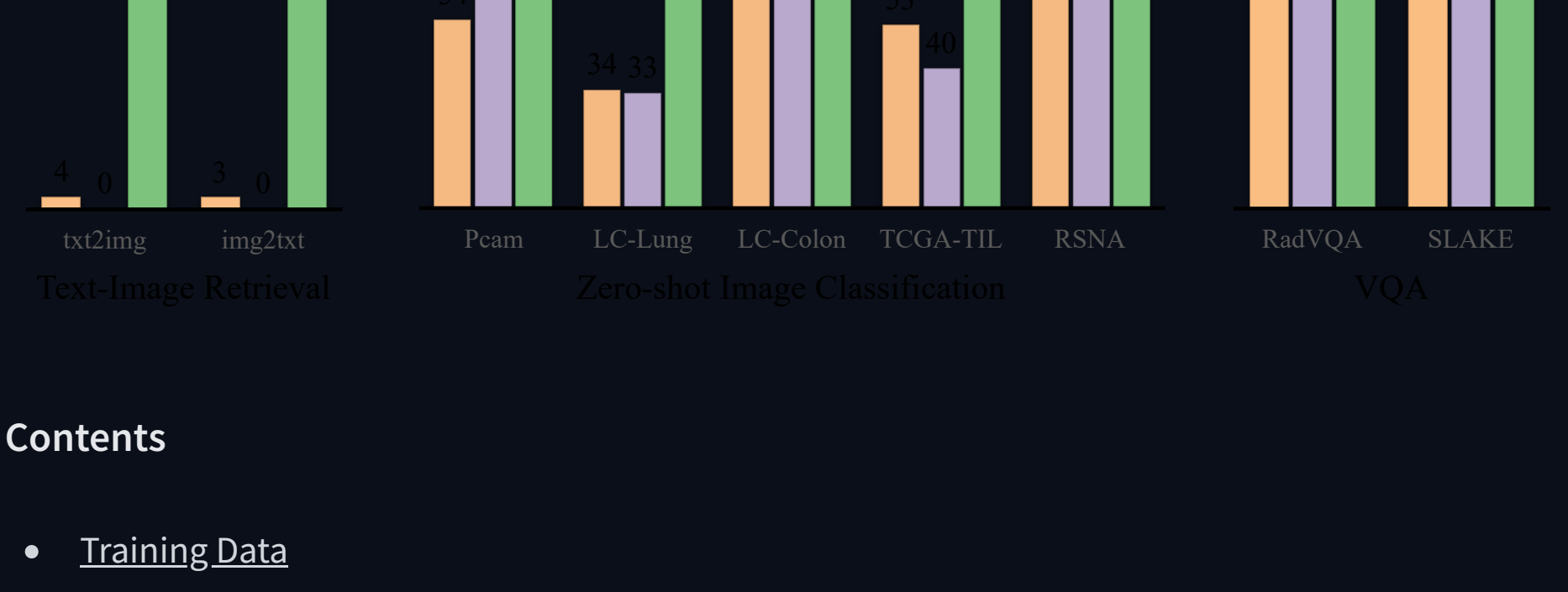


## BiomedCLIP-PubMedBERT\_256-vit\_base\_patch16\_224

BiomedCLIP is a biomedical vision-language foundation model that is pretrained on [PMC15M](#), a dataset of 15 million figure-caption pairs extracted from biomedical research articles in PubMed Central, using contrastive learning. It uses PubMedBERT as the text encoder and Vision Transformer as the image encoder, with domain-specific adaptations. It can perform various vision-language processing (VLP) tasks such as cross-modal retrieval, image classification, and visual question answering. BiomedCLIP establishes new state of the art in a wide range of standard datasets, and substantially outperforms prior VLP approaches:



### Contents

- Training Data
- Model Use
- Reference
- Limitations
- Further Information

### Training Data

We have released BiomedCLIP Data Pipeline at [https://github.com/microsoft/BiomedCLIP\\_data\\_pipeline](https://github.com/microsoft/BiomedCLIP_data_pipeline), which automatically downloads and processes a set of articles from the PubMed Central Open Access dataset. BiomedCLIP builds upon the PMC-15M dataset, which is a large-scale parallel image-text dataset generated by this data pipeline for biomedical vision-language processing. It contains 15 million figure-caption pairs extracted from biomedical research articles in PubMed Central and covers a diverse range of biomedical image types, such as microscopy, radiography, histology, and more.

### Model Use

#### 1. Environment

```
conda create -n biomedclip python=3.10 -y
conda activate biomedclip
pip install open_clip_torch==2.23.0 transformers==4.35.2 matplotlib
```

#### 2.1 Load from HF hub

```
import torch
from urllib.request import urlopen
from PIL import Image
from open_clip import create_model_from_pretrained, get_tokenizer

# Load the model and config files from the Hugging Face Hub
model, preprocess = create_model_from_pretrained('hf-hub:microsoft/BiomedCLIP-PubMedBERT_256-vit_base_patch16_224')
tokenizer = get_tokenizer('hf-hub:microsoft/BiomedCLIP-PubMedBERT_256-vit_base_patch16_224')

# Zero-shot image classification
template = 'this is a photo of '
labels = [
    'adenocarcinoma histopathology',
    'brain MRI',
    'covid line chart',
    'squamous cell carcinoma histopathology',
    'immunohistochemistry histopathology',
    'bone X-ray',
    'chest X-ray',
    'pie chart',
    'hematoxylin and eosin histopathology'
]

dataset_url = 'https://huggingface.co/microsoft/BiomedCLIP-PubMedBERT_256-vit_base_patch16_224'
test_imgs = [
    'squamous_cell_carcinoma_histopathology.jpeg',
    'H_and_E_histopathology.jpg',
    'bone_X-ray.jpg',
    'adenocarcinoma_histopathology.jpg',
    'covid_line_chart.png',
    'IHC_histopathology.jpg',
    'chest_X-ray.jpg',
    'brain_MRI.jpg',
    'pie_chart.png'
]

device = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
model.to(device)
model.eval()

context_length = 256

images = torch.stack([preprocess(Image.open(urlopen(dataset_url + img))) for img in test_imgs])
texts = tokenizer([template + l for l in labels], context_length=context_length)

with torch.no_grad():
    image_features, text_features, logit_scale = model(images, texts)

logits = (logit_scale * image_features @ text_features.t()).detach().softmax(-1)
sorted_indices = torch.argsort(logits, dim=-1, descending=True)

logits = logits.cpu().numpy()
sorted_indices = sorted_indices.cpu().numpy()

top_k = -1

for i, img in enumerate(test_imgs):
    pred = labels[sorted_indices[i][0]]

    top_k = len(labels) if top_k == -1 else top_k
    print(img.split('/')[-1] + ' : ' + pred)
    for j in range(top_k):
        jth_index = sorted_indices[i][j]
        print(f'{labels[jth_index]}: {logits[i][jth_index]}')
    print('\n')
```

#### 2.2 Load from local files

```
import json

from urllib.request import urlopen
from PIL import Image
import torch
from huggingface_hub import hf_hub_download
from open_clip import create_model_and_transforms, get_tokenizer
from open_clip.factory import HF_HUB_PREFIX, _MODEL_CONFIGS

# Download the model and config files
hf_hub_download(
    repo_id='microsoft/BiomedCLIP-PubMedBERT_256-vit_base_patch16_224',
    filename='open_clip_pytorch_model.bin',
    local_dir='checkpoints'
)
hf_hub_download(
    repo_id='microsoft/BiomedCLIP-PubMedBERT_256-vit_base_patch16_224',
    filename='open_clip_config.json',
    local_dir='checkpoints'
)

# Load the model and config files
model_name = 'biomedclip_local'

with open('checkpoints/open_clip_config.json', 'r') as f:
    config = json.load(f)
    model_cfg = config['model_cfg']
    preprocess_cfg = config['preprocess_cfg']

if (not model_name.startswith(HF_HUB_PREFIX)
    and model_name not in _MODEL_CONFIGS
    and config is not None):
    _MODEL_CONFIGS[model_name] = model_cfg

tokenizer = get_tokenizer(model_name)

model, _, preprocess = create_model_and_transforms(
    model_name=model_name,
    pretrained='checkpoints/open_clip_pytorch_model.bin',
    **{f'image_{k}': v for k, v in preprocess_cfg.items()}
)

# Zero-shot image classification
template = 'this is a photo of '
labels = [
    'adenocarcinoma histopathology',
    'brain MRI',
    'covid line chart',
    'squamous cell carcinoma histopathology',
    'immunohistochemistry histopathology',
    'bone X-ray',
    'chest X-ray',
    'pie chart',
    'hematoxylin and eosin histopathology'
]

dataset_url = 'https://huggingface.co/microsoft/BiomedCLIP-PubMedBERT_256-vit_base_patch16_224'
test_imgs = [
    'squamous_cell_carcinoma_histopathology.jpeg',
    'H_and_E_histopathology.jpg',
    'bone_X-ray.jpg',
    'adenocarcinoma_histopathology.jpg',
    'covid_line_chart.png',
    'IHC_histopathology.jpg',
    'chest_X-ray.jpg',
    'brain_MRI.jpg',
    'pie_chart.png'
]

device = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
model.to(device)
model.eval()

context_length = 256

images = torch.stack([preprocess(Image.open(urlopen(dataset_url + img))) for img in test_imgs])
texts = tokenizer([template + l for l in labels], context_length=context_length)

with torch.no_grad():
    image_features, text_features, logit_scale = model(images, texts)

logits = (logit_scale * image_features @ text_features.t()).detach().softmax(-1)
sorted_indices = torch.argsort(logits, dim=-1, descending=True)

logits = logits.cpu().numpy()
sorted_indices = sorted_indices.cpu().numpy()

top_k = -1

for i, img in enumerate(test_imgs):
    pred = labels[sorted_indices[i][0]]

    top_k = len(labels) if top_k == -1 else top_k
    print(img.split('/')[-1] + ' : ' + pred)
    for j in range(top_k):
        jth_index = sorted_indices[i][j]
        print(f'{labels[jth_index]}: {logits[i][jth_index]}')
    print('\n')
```

### Use in Jupyter Notebook

Please refer to this [example notebook](#)

### Intended Use

This model is intended to be used solely for (i) future research on visual-language processing and (ii) reproducibility of the experimental results reported in the reference paper.

### Primary Intended Use

The primary intended use is to support AI researchers building on top of this work. BiomedCLIP and its associated models should be helpful for exploring various biomedical VLP research questions, especially in the radiology domain.

### Out-of-Scope Use

Any deployed use case of the model --- commercial or otherwise --- is currently out of scope. Although we evaluated the models using a broad set of publicly-available research benchmarks, the models and evaluations are not intended for deployed use cases. Please refer to [the associated paper](#) for more details.

### Reference

```
@article{zhang2024biomedclip,
  title={A Multimodal Biomedical Foundation Model Trained from Fifteen Million Images},
  author={Shang Zhang and Yanbo Xu and Naoto Usuyama and Hammen Xu and Jaspreet B. Jeyaraj and NEJM AI},
  year={2024},
  volume={2},
  number={1},
  doi={10.1056/AIoa2408640},
  url={https://ai.nejm.org/doi/full/10.1056/AIoa2408640}
}
```

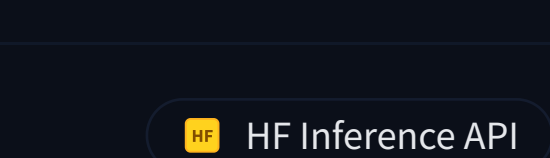
### Limitations

This model was developed using English corpora, and thus can be considered English-only.

### Further information

Please refer to the corresponding paper, ["Large-Scale Domain-Specific Pretraining for Biomedical Vision-Language Processing"](#), for additional details on the model training and evaluation.

Downloads last month  
70,841



### Inference Providers

Zero Shot Image Classification

Drag image file here or click to browse from your device

Possible class names (comma-separated)

Possible class names...

Compute

View Code

Maximize

Model tree for microsoft/BiomedCLIP-PubMedBERT\_256-vit\_base\_patch16\_224

Finetunes

1 model

### Spaces using microsoft/BiomedCLIP-PubMedBERT\_256-vit\_base\_patch16\_224

- Baron GG/LLAUS
- mMonika/Medical\_bot
- StefanDenner/MedicalVisualPromptEngineering
- zenitsu55/microsoft-BiomedCLIP-PubMedBERT\_256-vit\_base\_patch16\_224
- mitchmomo/microsoft-BiomedCLIP-PubMedBERT\_256-vit\_base\_patch16\_224
- charlestonX/microsoft-BiomedCLIP-PubMedBERT\_256-vit\_base\_patch16\_224
- Raja456/microsoft-BiomedCLIP-PubMedBERT\_256-vit\_base\_patch16\_224
- pZacca/microsoft-BiomedCLIP-PubMedBERT\_256-vit\_base\_patch16\_224
- Dobator/microsoft-BiomedCLIP-PubMedBERT\_256-vit\_base\_patch16\_224
- Mustafaage/microsoft-BiomedCLIP-PubMedBERT\_256-vit\_base\_patch16\_224
- moammedRiad/microsoft-BiomedCLIP-PubMedBERT\_256-vit\_base\_patch16\_224
- Harshdhi/microsoft-BiomedCLIP-PubMedBERT\_256-vit\_base\_patch16\_224
- 12 Spaces

### Collection including microsoft/BiomedCLIP-PubMedBERT\_256-vit\_base\_patch16\_224

Biomedical Collection  
Models for biomedical research applications... • 10 items • Updated 3 days ago • 10