SAM.cpp

Inference of Meta's Segment Anything Model in pure C/C++

- 使用纯C/C++对meta的segment anything model进行推理

Description（描述）

The example currently supports only the ViT-B SAM model checkpoint.

- 这个例子目前仅支持ViT-B SAM model checkpoint
- 其他类型的model现在还不支持

Next steps（下一步）

- ☑ Reduce memory usage by utilizing the new ggml-alloc（通过使用新的ggml-alloc减少内存的使用）
- ☑ Remove redundant graph nodes（移除冗余的graph nodes）
- □ Make inference faster(使得推理更快)
- ☑ Fix the difference in output masks compared to the PyTorch implementation（修复与 PyTorch 实现相比输出掩码的差异）
- ☑ Filter masks based on stability score（根据稳定性评分过滤口罩）
- □ Add support for user input（支持用户输入）
- □ Support F16 for heavy F32 ops
- □ Test quantization
- ☑ Support bigger model checkpoints（支持更大的model checkpoints）
- □ GPU support

Quick start（快速开始）

Setup Python and build examples according to main README.

- 设置python
- 根据main README构建examples

```
# Download PTH model
wget -P examples/sam/ https://dl.fbaipublicfiles.com/segment_anything/sam_vit_b_01ec64.pth

# Convert PTH model to ggml
python examples/sam/convert-pth-to-ggml.py examples/sam/sam_vit_b_01ec64.pth examples/sam/ 1

# run inference
./bin/sam -t 16 -i ../examples/sam/example.jpg -m ../examples/sam/ggml-model-f16.bin
```

Downloading and converting the model checkpoints(下载和转化model checkpoints)

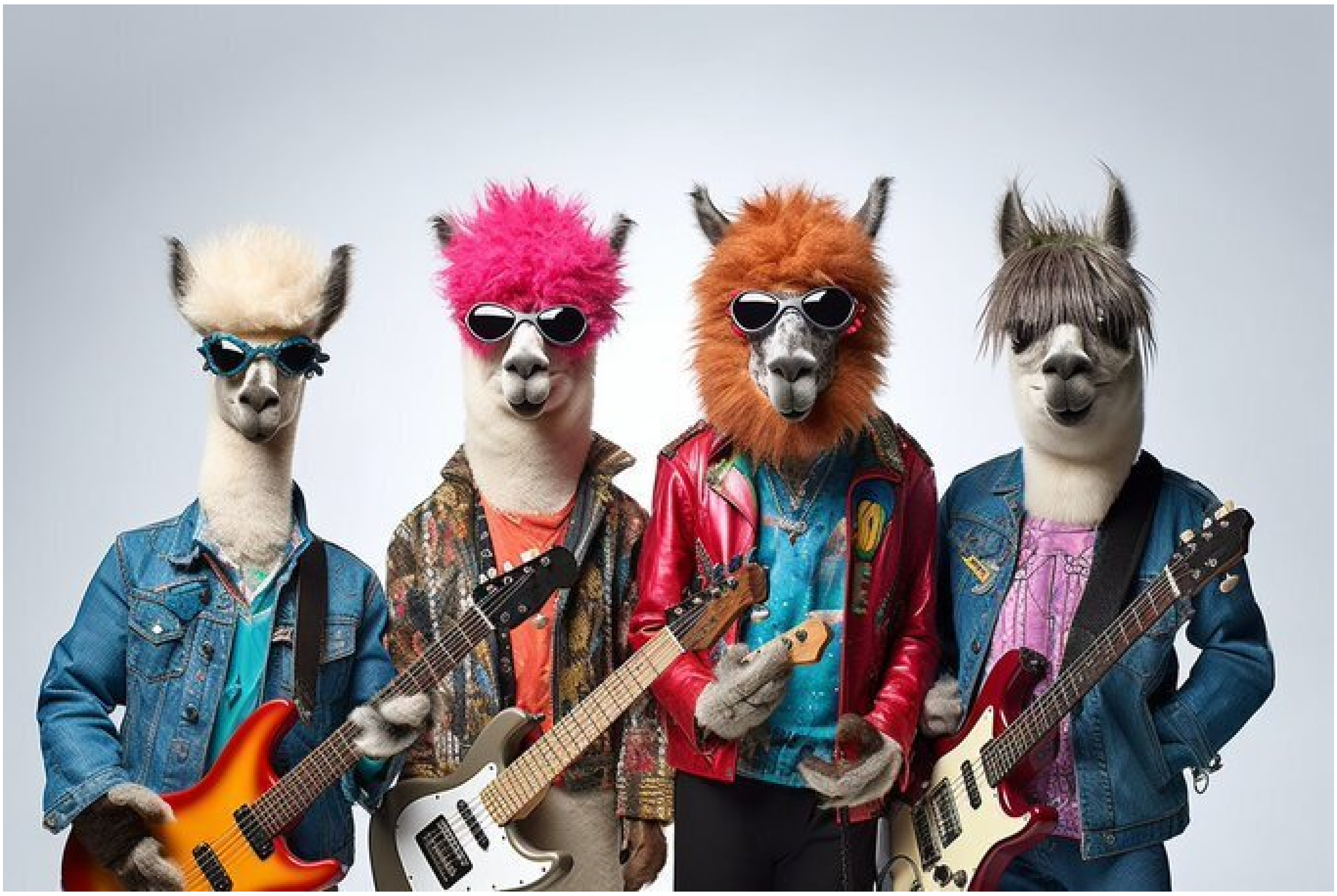
You can download a model checkpoint and convert it to `ggml` format using the script `convert-pth-to-ggml.py`:

Example output on M2 Ultra

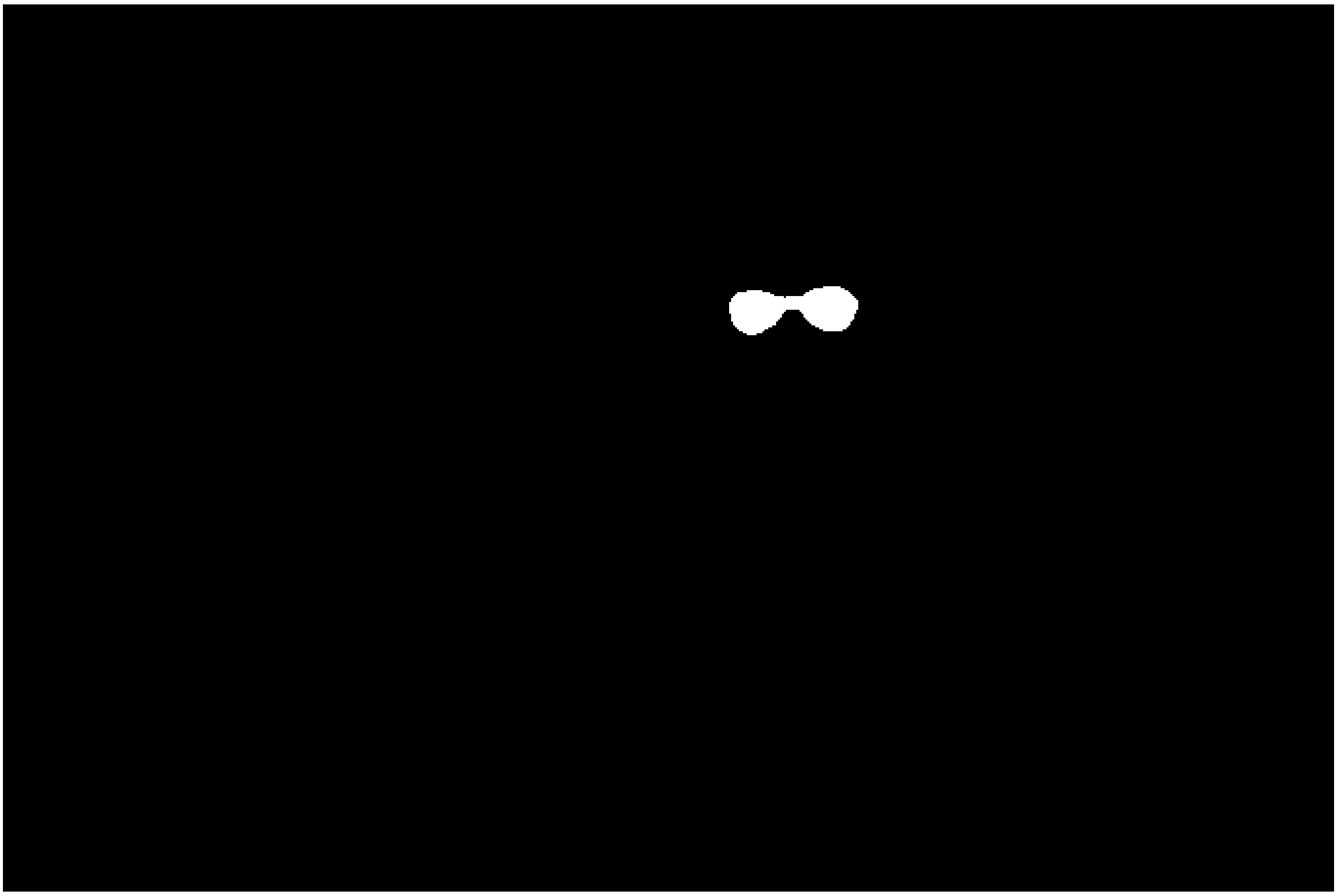
```
 $ ▶ make -j sam && time ./bin/sam -t 8 -i img.jpg
[ 28%] Built target common
[ 71%] Built target ggml
[100%] Built target sam
main: seed = 1693224265
main: loaded image 'img.jpg' (680 x 453)
sam_image_preprocess: scale = 0.664062
main: preprocessed image (1024 x 1024)
sam_model_load: loading model from 'models/sam-vit-b/ggml-model-f16.bin' - please
wait ...
sam_model_load: n_enc_state      = 768
sam_model_load: n_enc_layer      = 12
sam_model_load: n_enc_head       = 12
sam_model_load: n_enc_out_chans  = 256
sam_model_load: n_pt_embd        = 4
sam_model_load: ftype            = 1
sam_model_load: qntvr            = 0
operator(): ggml ctx size = 202.32 MB
sam_model_load: ...................................... done
sam_model_load: model size =   185.05 MB / num tensors = 304
embd_img
dims: 64 64 256 1 f32
First & Last 10 elements:
-0.05117 -0.06408 -0.07154 -0.06991 -0.07212 -0.07690 -0.07508 -0.07281 -0.07383
-0.06779
0.01589 0.01775 0.02250 0.01675 0.01766 0.01661 0.01811 0.02051 0.02103 0.03382
sum:  12736.272313

Skipping mask 0 with iou 0.705935 below threshold 0.880000
Skipping mask 1 with iou 0.762136 below threshold 0.880000
Mask 2: iou = 0.947081, stability_score = 0.955437, bbox (371, 436), (144, 168)


main:     load time =    51.28 ms
main:    total time =  2047.49 ms

real    0m2.068s
user    0m16.343s
sys 0m0.214s
```

Input point is (414.375, 162.796875) (currently hardcoded)

Input image:



Output mask (mask_out_2.png in build folder):



References

- ggml
- SAM
- SAM demo