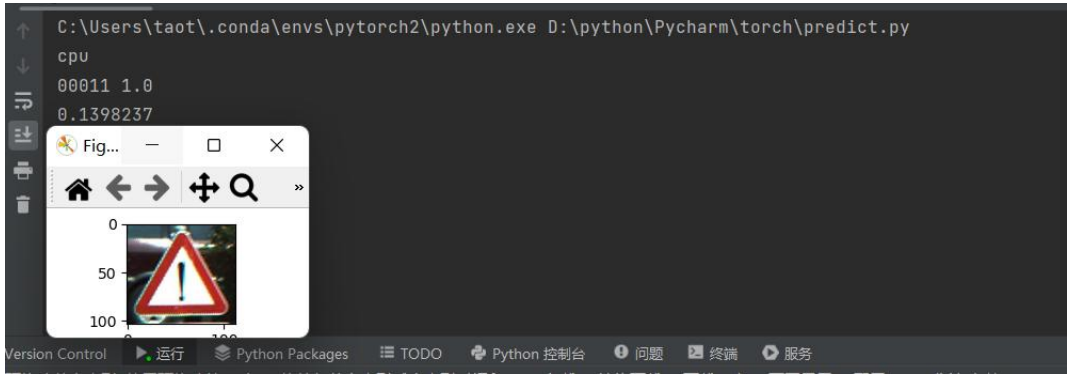
- 模型推演结果及比较：

```
 -1.93212390e-01 -1.70112640e-01]
[-1.60855129e-02 -2.03645632e-01  4.16578650e-02  4.22625616e-02
  3.01629305e-03 -1.39177904e-01  2.47835740e-02 -4.64780703e-02
 -1.98652163e-01  4.78569046e-03]]
Time consuming: 0.08834 sec
------------------------------------------------------------
right
```

```
C:\Users\taot\.conda\envs\pytorch2\python.exe D:\python\Pycharm\torch\predict.py
cuda:0
00011 1.0
0.13877509999999998
```



处理器	推演时长
TUP	0.08834s
CPU	0.1398s
GPU	0.1387s

同样的模型，分别使用CUP，TUP，GPU进行推理，TUP在张量计算时的效率远远高于CUP和GUP,充分展现TPU承载机器学习是实现人工智能的强有力方法。明显的体现了TPU的AI加速特性。

关键技术介绍

- 工具链展示：

```python
# net = AlexNet(num_classes=17, init_weights=True)
net = models.vgg16(pretrained=True)
del net.classifier[2]
del net.classifier[4]
# print(net)
# num_features = net.fc.in_features
net.classifier[4] = nn.Linear(4096, 17)
print(net)
net.to(device)
#损失函数:这里用交叉熵
loss_function = nn.CrossEntropyLoss()
```

```python
print(device)


del vgg.classifier[2]
del vgg.classifier[4]
vgg.classifier[4] = nn.Linear(4096, 17)
model =vgg
print(vgg)

model.load_state_dict(torch.load('Vgg_16.pth', map_location=device))
model = model.to(device)

example_input = torch.randn(1, 3, 32, 32).to(device)
traced_model = torch.jit.trace(model, example_input)
traced_model.save('vgg_16.pt')

print("Model traced and saved successfully.")
model = torch.jit.load('vgg_16.pt', map_location="cpu")
model.eval()
input_data = torch.randn(1, 3, 32, 32)
traced_model = torch.jit.trace(model, [input_data])
traced_model_name = "vgg_16.pt"
traced_model.save('vgg_16.pt')
```

```
  -1.93212390e-01 -1.70112640e-01]
 [-1.60855129e-02 -2.03645632e-01  4.16578650e-02  4.22625616e-02
   3.01629305e-03 -1.39177904e-01  2.47835740e-02 -4.64780703e-02
  -1.98652163e-01  4.78569046e-03]]
Time consuming: 0.08834 sec
--------------------------------------------------------------------
right
```

集成一套工具链，将vgg模型移植至少林派（BM1684），并正确推演。

```
*** Instruction generation process for subnet 2
.....I0719 16:09:37.385697   221 bmcompiler_c
I0719 16:09:37.385763   221 bmcompiler_context_subne
I0719 16:09:37.388875   221 bmcompiler_context.cpp:3

*** Store bmodel of BMCompiler...
```

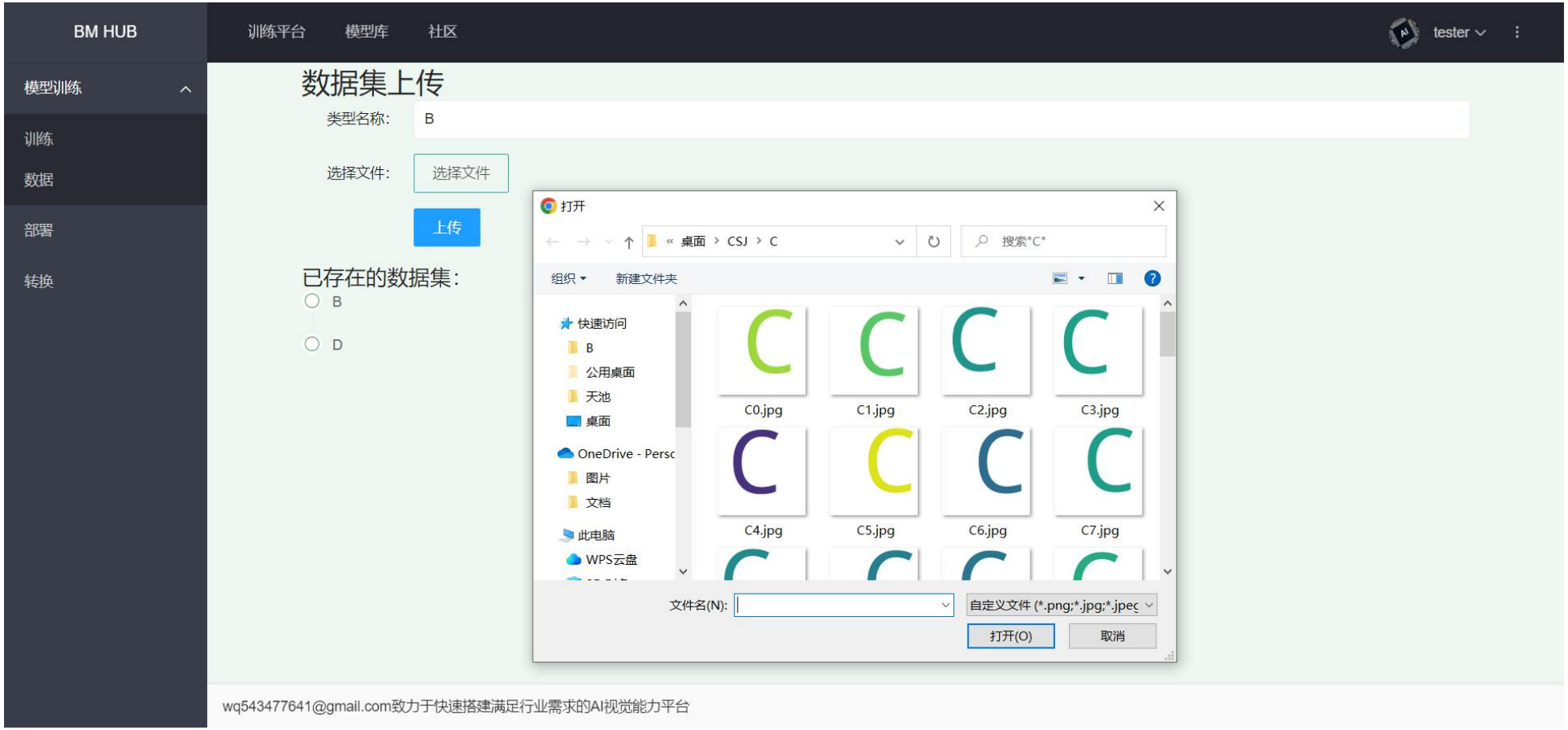
```python
import cv2
import time
import argparse
import numpy as np
from PIL import Image
import sophon.sail as sail


def preprocess(img):
    image = Image.fromarray(img)
    resized_img = np.array(image.resize((224,224), resample=2))
    out = np.array(resized_img / 255., dtype=np.float32)
    return out.transpose((2, 0, 1))
```

关键技术介绍

- 在线训推一体平台展示：

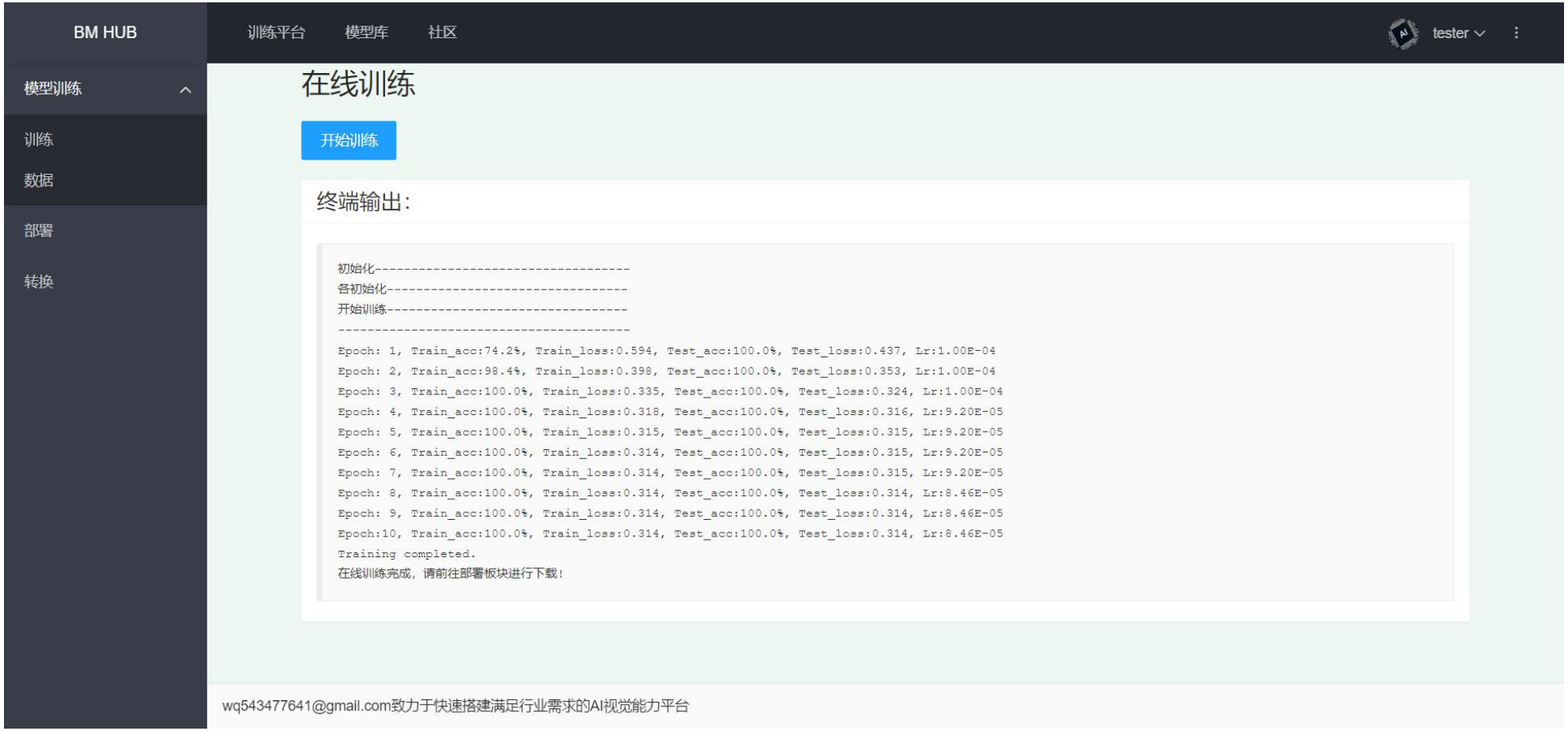
数据集上传功能。用户可以轻松地自行上传其所需的数据集，并设置数据的不同类别和属性。这为个性化的训练提供了便捷的手段。



关键技术介绍

- 在线训推一体平台展示：

此外，我们的在线训练功能为用户提供了强大的在线训练和转换服务。



关键技术介绍

- 在线训推一体平台展示

在训练完成后，我们的工具链会自动为用户生成bmodel文件和一份推演代码。这些代码可以直接供用户使用，或封装为接口函数，以支持用户发新的AI应用。