

mue_demo_czjing_20220331

March 31, 2022

1 mue_demo_czjing_20220331

1.1 setting

```
[1]: import sys
import timm
import torch
import torchvision
import numpy as np
import pandas as pd
from PIL import Image    # open image
import matplotlib.pyplot as plt

device = torch.device("cuda:3" if torch.cuda.is_available() else "cpu")
print(">>> using: ", device)
```

>>> using: cuda:3

```
[2]: from data.video_folder_custom import VideoFolderCustom
from data.show_video_dataloaders import showVideoInputs

# data path
data_path = '/home/zjcao/work/data/nvGesture/nvGesture_color_video_processed'
data_anno_path = {
    'val' :  '/home/zjcao/work/data/nvGesture/nvGesture_color_video_processed/
    ↪val_v2.csv'
}

# data transforms
data_transforms = {
    'val': torchvision.transforms.Compose([
        torchvision.transforms.Resize(256),      # image batch, resize smaller_
        ↪edge to 224
        torchvision.transforms.CenterCrop(224),  # image batch, center crop to_
        ↪square 224*224
        # torchvision.transforms.ToTensor(),
```

```

        torchvision.transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.
→229, 0.224, 0.225]),
    ])
}

# video_datasets
video_datasets = {
    'val': VideoFolderCustom(
        data_path=data_path,
        anno_path=data_anno_path['val'],
        model='val',
        num_segments=16,
        clip_length=1,
        frame_sample_rate=2,
        transform=data_transforms['val']
    )
}

# video dataloaders
video_dataloaders = {
    'val' : torch.utils.data.DataLoader(
        video_datasets['val'],
        batch_size=4,
        shuffle=True,
        num_workers=4,
        drop_last=True,
    )
}

# video dataset sizes
print(">>> Val set: ", len(video_dataloaders['val'].dataset))

# testing dataloaders
_inputs, _classes = next(iter(video_dataloaders['val']))
print(">>> Inputs shape:", _inputs.shape, " labels:", _classes)

# show video
showVideoInputs(_inputs, _classes)

```

```

>>> Val set: 482
>>> Inputs shape: torch.Size([4, 3, 16, 224, 224]) labels: tensor([ 6, 14,  0,
3])
>>> Display inputs[0], labels: 6

```



1.2 model

```
[3]: # model
model = torch.hub.load('facebookresearch/pytorchvideo', 'slow_r50',
    ↪pretrained=True)

# modify last fc
num_fts = model.blocks[5].proj.in_features
model.blocks[5].proj = torch.nn.Linear(num_fts, 25)

# testing model
_output = model(torch.randn(2,3,16,256,256))
print(_output.shape)
```

Using cache found in
/home/zjcao/.cache/torch/hub/facebookresearch_pytorchvideo_main
torch.Size([2, 25])

1.3 loading weights

```
[4]: # model
new_model = torch.hub.load('facebookresearch/pytorchvideo', 'slow_r50',
    ↪pretrained=True)

# modify fc
num_fts = new_model.blocks[5].proj.in_features
new_model.blocks[5].proj = torch.nn.Linear(num_fts, 25)

# loading weights
new_model.load_state_dict(torch.load('./model_pth/res50_3d_color_20220325.pth',
    ↪map_location=device))

print(new_model.blocks[5])
```

```

Using cache found in
/home/zjcao/.cache/torch/hub/facebookresearch_pytorchvideo_main

ResNetBasicHead(
  (pool): AvgPool3d(kernel_size=(8, 7, 7), stride=(1, 1, 1), padding=(0, 0, 0))
  (dropout): Dropout(p=0.5, inplace=False)
  (proj): Linear(in_features=2048, out_features=25, bias=True)
  (output_pool): AdaptiveAvgPool3d(output_size=1)
)

```

1.4 MC dropouts

```

[5]: from utils.mc_dropout import enable_dropout

# eval
new_model.eval()

# enable model
new_model = enable_dropout(new_model)

```

1.5 demo_1 (in domain)

```

[6]: from utils.mc_dropout import demo_single_video

# testing dataloaders
_inputs, _labels = next(iter(video_dataloaders['val']))

print(">>> Inputs shape:", _inputs.shape, " labels:", _labels)
showVideoInputs(_inputs, _labels)

new_model.to(device)
inputs = _inputs[0:1].to(device)

_, preds, uncertainty = demo_single_video(new_model, inputs, num_forwards=5)
print(">>> Predict:", preds)
print(">>> Uncertainty:", uncertainty)

```

```

>>> Inputs shape: torch.Size([4, 3, 16, 224, 224]) labels: tensor([ 9, 12,  6,
21])
>>> Display inputs[0], labels: 9

```



```
>>> Predict: 9
>>> Uncertainty: 0.3489
```

1.6 demo_2 (out of domain)

```
[7]: from data.video_folder_custom import VideoFolderCustom
      from data.show_video_data loaders import showVideoInputs

      # data path
      data_path = '/home/zjcao/work/data/UCF101/'
      data_anno_path = {
          'val': '/home/zjcao/work/data/UCF101/ucfTrainTestlist/un_test.csv'
      }

      # data transforms
      data_transforms = {
          'val': torchvision.transforms.Compose([
              torchvision.transforms.Resize(256),
              torchvision.transforms.CenterCrop(224),
              torchvision.transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.
↪229, 0.224, 0.225])),
          ])
      }

      # video_datasets
      video_datasets = {
          'val': VideoFolderCustom(
              data_path=data_path,
              anno_path=data_anno_path['val'],
              model='val',
              num_segments=16,
              clip_length=1,
              frame_sample_rate=2,
              transform=data_transforms['val']
          )
      }
```

```

}

# video dataloaders
video_dataloaders = {
    'val' : torch.utils.data.DataLoader(
        video_datasets['val'],
        batch_size=4,
        shuffle=True,
        num_workers=4,
        drop_last=True,
    )
}

# video dataset sizes
print(">>> Val set: ", len(video_dataloaders['val'].dataset))

# testing dataloaders
_inputs, _classes = next(iter(video_dataloaders['val']))
print(">>> Inputs shape:", _inputs.shape, " labels:", _classes)

# show video
showVideoInputs(_inputs, _classes)

```

```

>>> Val set: 178
>>> Inputs shape: torch.Size([4, 3, 16, 224, 224]) labels: tensor([0, 0, 0, 0])
>>> Display inputs[0], labels: 0

```



```

[8]: # testing dataloaders
_inputs, _labels = next(iter(video_dataloaders['val']))

showVideoInputs(_inputs, _labels)

new_model.to(device)
inputs = _inputs[0:1].to(device)

_, preds, uncertainty = demo_single_video(new_model, inputs, num_forwards=5)

```

```
print(">>> Predict:", preds)
print(">>> Uncertainty:", uncertainty)
```

```
>>> Display inputs[0], labels: 0
```



```
>>> Predict: 18
>>> Uncertainty: 0.8899
```

```
[ ]:
```