
```
from fastai.vision.all import *
from fastai.distributed import *
from fastai.vision.models.xresnet import *

from accelerate import notebook_launcher
from accelerate import Accelerator
from accelerate.utils import set_seed
from timm import create_model

from accelerate.utils import write_basic_config
```

```
path = '/home/andrea/Documents/
Segmentation/Operative/Batch_5'
path_im = path + '/Impng'
path_lbl = path + '/fuse'
path_Rflbl = path + '/New_Labels'
```

```
# from accelerate import notebook_launcher
```

```
def get_msk(o):
    return path_Rflbl+fr'/RfM_{o.stem}
{o.suffix.lower()}___fuse{o.suffix.lower()}'

numeral_codes=[i for i in range(0,16)]
print('numeral codes ', numeral_codes)
#numeral codes understod by FastAI

file = open(path+'/codes.txt', "w+")

# Saving the array in a text file
content = str(numeral_codes)
file.write(content)
file.close()



def train():
    dls =
SegmentationDataLoaders.from_label_func(
        path, bs=8, fnames =
get_image_files(path+'/Impng'),
        label_func = get_msk,
after_item=ToTensor(),
        codes = np.loadtxt(path+'/codes.txt',
dtype=str)
        )
    learn = unet_learner(resnet34,dls,
    dls=TfmdDL(after_item=ToTensor(4,80,80),
                 after_batch=[IntToFloatTensor(),
*aug_transforms()], bs=8))
    with learn.distrib_ctx(in_notebook=True,
sync_bn=False):
        learn.fit(10)



notebook_launcher(train, num_processes=4)
```

----

this is where it comes from

+*In[ ]:*+
[source, ipython3]

```
path = untar_data(URLs.CAMVID_TINY)

def train():
    dls =
SegmentationDataLoaders.from_label_func(
        path, bs=8, fnames =
get_image_files(path/"images"),
        label_func = lambda o: path/'labels'/
f'{o.stem}_P{o.suffix}',
        codes = np.loadtxt(path/'codes.txt',
dtype=str)
    )
    learn = unet_learner(dls, resnet34)
    with learn.distrib_ctx(in_notebook=True,
sync_bn=False):
        learn.fine_tune(8)

notebook_launcher(train, num_processes=2)

Launching training on 2 GPUs.
```
