Image segmentation with CamVid

```
In [1]: %reload_ext autoreload
        %autoreload 2
        %matplotlib inline
```

```
In [2]: from fastai.vision import *
        from fastai.callbacks.hooks import *
        from fastai.utils.mem import *
```

```
In [3]: path = untar_data(URLs.CAMVID)
        path.ls()
```

```
Out[3]: [PosixPath('/home/ml1/.fastai/data/camvid/images'),
         PosixPath('/home/ml1/.fastai/data/camvid/valid.txt'),
         PosixPath('/home/ml1/.fastai/data/camvid/labels'),
         PosixPath('/home/ml1/.fastai/data/camvid/codes.txt')]
```

```
In [4]: path_lbl = path/'labels'
        path_img = path/'images'
```

Subset classes

```
In [ ]: # path = Path('./data/camvid-small')

        # def get_y_fn(x): return Path(str(x.parent)+'annot')/x.name

        # codes = array(['Sky', 'Building', 'Pole', 'Road', 'Sidewalk', 'Tree',
        #     'Sign', 'Fence', 'Car', 'Pedestrian', 'Cyclist', 'Void'])

        # src = (SegmentationItemList.from_folder(path)
        #        .split_by_folder(valid='val')
        #        .label_from_func(get_y_fn, classes=codes))

        # bs=8
        # data = (src.transform(get_transforms(), tfm_y=True)
        #         .databunch(bs=bs)
        #         .normalize(imagenet_stats))
```

Data

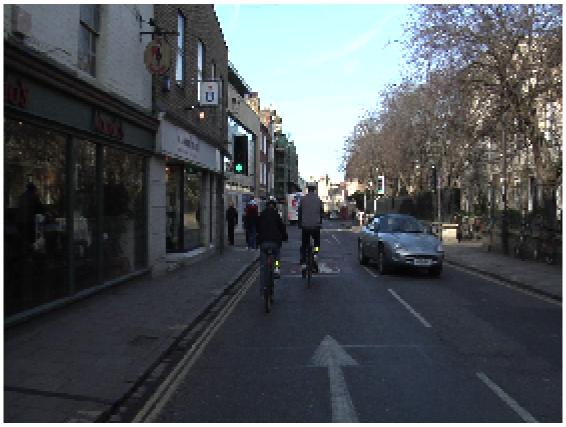
In [5]:
```
fnames = get_image_files(path_img)
fnames[:3]
```

Out[5]:
```
[PosixPath('/home/ml1/.fastai/data/camvid/images/0016E5_08021.png'),
 PosixPath('/home/ml1/.fastai/data/camvid/images/Seq05VD_f02430.png'),
 PosixPath('/home/ml1/.fastai/data/camvid/images/0006R0_f03690.png')]
```

In [6]:
```
lbl_names = get_image_files(path_lbl)
lbl_names[:3]
```

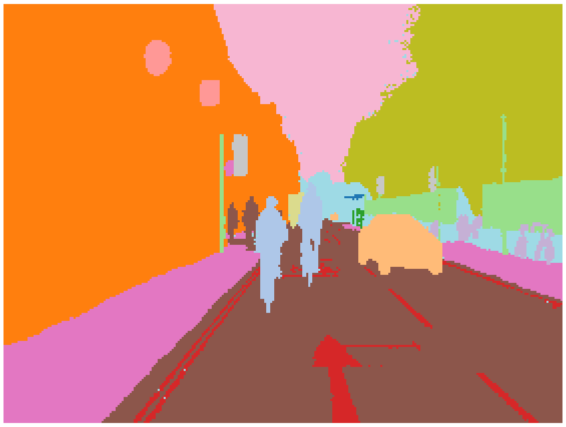
Out[6]:
```
[PosixPath('/home/ml1/.fastai/data/camvid/labels/0006R0_f03870_P.png'),
 PosixPath('/home/ml1/.fastai/data/camvid/labels/Seq05VD_f00030_P.png'),
 PosixPath('/home/ml1/.fastai/data/camvid/labels/0016E5_07830_P.png')]
```

In [7]:
```
img_f = fnames[0]
img = open_image(img_f)
img.show(figsize=(5,5))
```



In [8]:
```
get_y_fn = lambda x: path_lbl/f'{x.stem}_P{x.suffix}'
```

In [9]:
```
mask = open_mask(get_y_fn(img_f))
mask.show(figsize=(5,5), alpha=1)
```



In [10]:
```
src_size = np.array(mask.shape[1:])
src_size,mask.data
```

Out[10]:
```
(array([720, 960]), tensor([[[ 4,  4,  4,  ..., 26, 26, 26],
          [ 4,  4,  4,  ..., 26, 26, 26],
          [ 4,  4,  4,  ..., 26, 26, 26],
          ...,
          [19, 19, 19,  ..., 17, 17, 17],
          [19, 19, 19,  ..., 17, 17, 17],
          [19, 19, 19,  ..., 17, 17, 17]]]))
```

In [11]:
```
codes = np.loadtxt(path/'codes.txt', dtype=str); codes
```

Out[11]:
```
array(['Animal', 'Archway', 'Bicyclist', 'Bridge', 'Building', 'Car', 'CartLu
ggagePram', 'Child', 'Column_Pole',
       'Fence', 'LaneMkgsDriv', 'LaneMkgsNonDriv', 'Misc_Text', 'MotorcycleSc
ooter', 'OtherMoving', 'ParkingBlock',
       'Pedestrian', 'Road', 'RoadShoulder', 'Sidewalk', 'SignSymbol', 'Sky',
'SUVPickupTruck', 'TrafficCone',
       'TrafficLight', 'Train', 'Tree', 'Truck_Bus', 'Tunnel', 'VegetationMis
c', 'Void', 'Wall'], dtype='<U17')
```

Datasets

In [12]:
```
size = src_size//2

free = gpu_mem_get_free_no_cache()
# the max size of bs depends on the available GPU RAM
if free > 8200: bs=8
else:           bs=4
print(f"using bs={bs}, have {free}MB of GPU RAM free")
```

```
using bs=4, have 7941MB of GPU RAM free
```

In [13]:
```
path_img
```

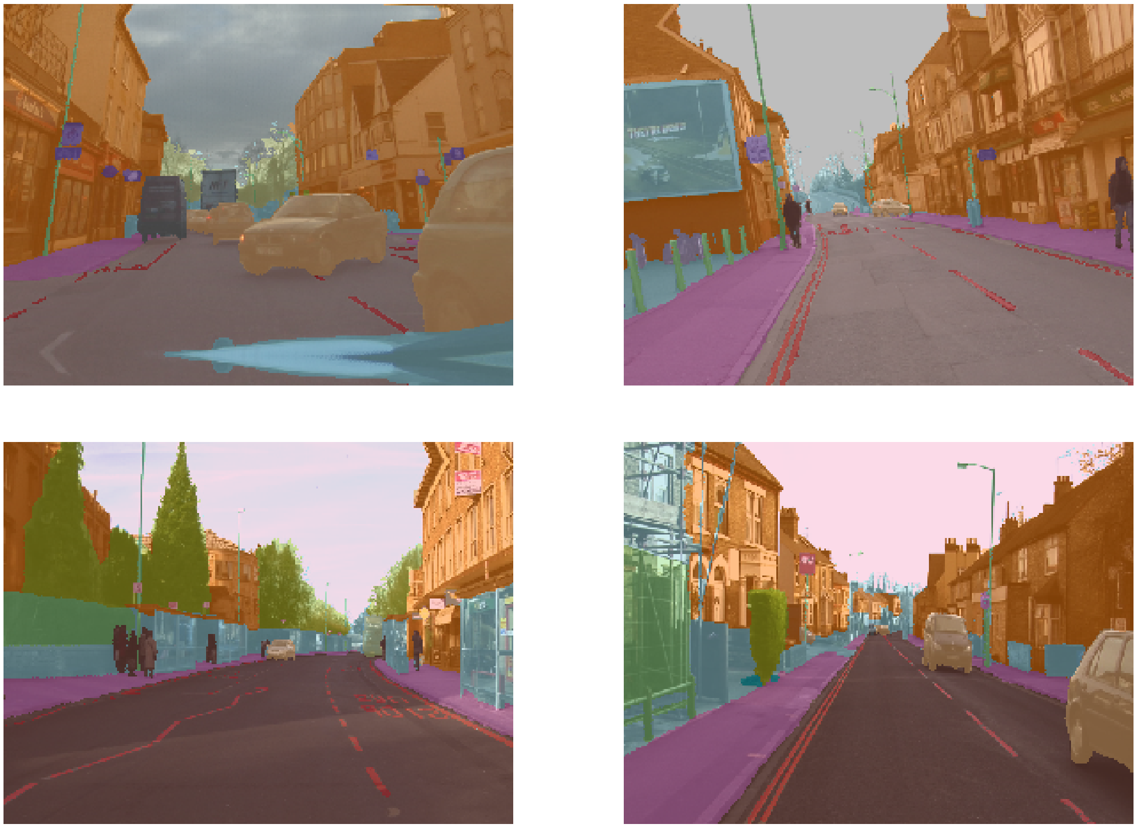
Out[13]:
```
PosixPath('/home/ml1/.fastai/data/camvid/images')
```

In [14]:
```
src = (SegmentationItemList.from_folder(path_img)
       .split_by_fname_file('../valid.txt')
       .label_from_func(get_y_fn, classes=codes))
```

In [15]:
```
data = (src.transform(get_transforms(), size=size, tfm_y=True)
        .databunch(bs=bs)
        .normalize(imagenet_stats))
```

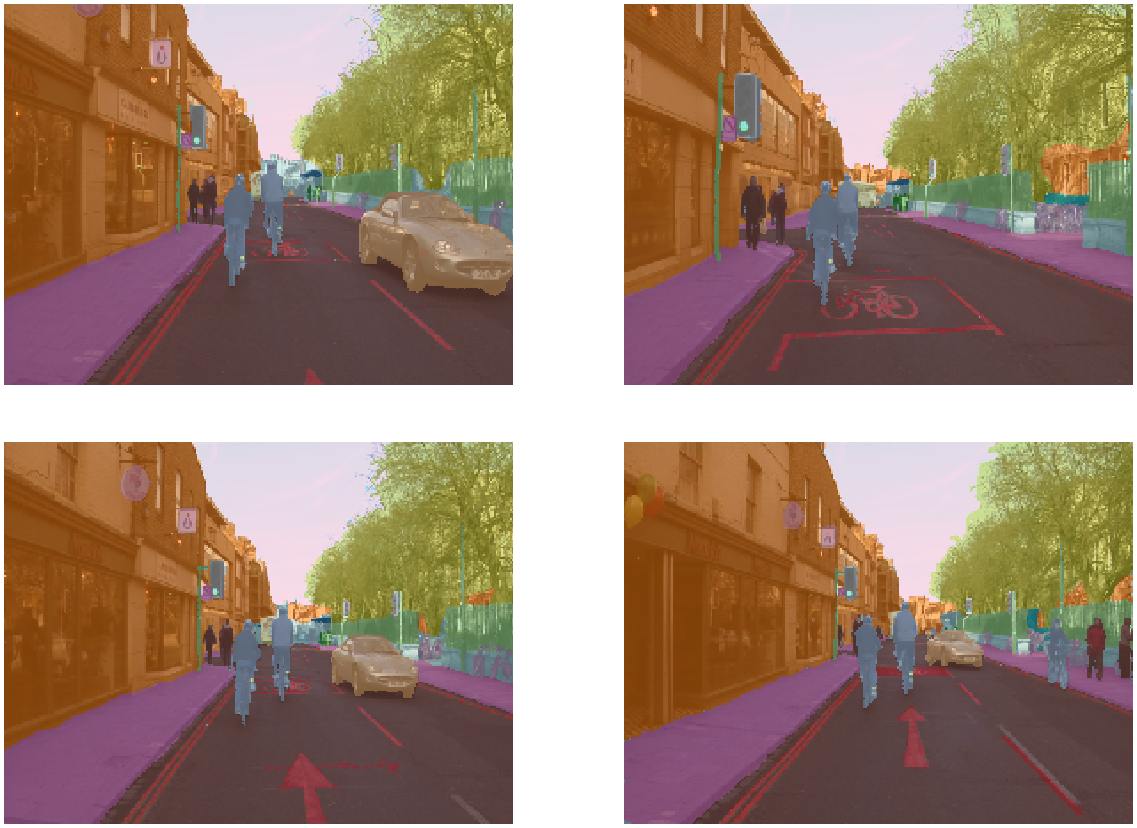
In [16]:

```
data.show_batch(2, figsize=(10,7))
```



In [ ]:

```
data.show_batch(2, figsize=(10,7), ds_type=DatasetType.Valid)
```



Model

In [17]:

```
name2id = {v:k for k,v in enumerate(codes)}
void_code = name2id['Void']

def acc_camvid(input, target):
    target = target.squeeze(1)
    mask = target != void_code
    return (input.argmax(dim=1)[mask]==target[mask]).float().mean()
```

In [18]:

```
metrics=acc_camvid
# metrics=accuracy
```

In [19]:

```
wd=1e-2
```

In [20]:

```
learn = unet_learner(data, models.resnet34, metrics=metrics, wd=wd)
```

```
In [48]: print(learn.summary())
```

```
======================================================================
Layer (type)         Output Shape         Param #    Trainable 
======================================================================
Conv2d               [1, 64, 360, 480]    9,408      True      
______________________________________________________________________
BatchNorm2d          [1, 64, 360, 480]    128        True      
______________________________________________________________________
ReLU                 [1, 64, 360, 480]    0          False     
______________________________________________________________________
MaxPool2d            [1, 64, 180, 240]    0          False     
______________________________________________________________________
Conv2d               [1, 64, 180, 240]    36,864     True      
______________________________________________________________________
BatchNorm2d          [1, 64, 180, 240]    128        True      
______________________________________________________________________
ReLU                 [1, 64, 180, 240]    0          False     
______________________________________________________________________
Conv2d               [1, 64, 180, 240]    36,864     True      
______________________________________________________________________
BatchNorm2d          [1, 64, 180, 240]    128        True      
______________________________________________________________________
Conv2d               [1, 64, 180, 240]    36,864     True      
______________________________________________________________________
BatchNorm2d          [1, 64, 180, 240]    128        True      
______________________________________________________________________
ReLU                 [1, 64, 180, 240]    0          False     
______________________________________________________________________
Conv2d               [1, 64, 180, 240]    36,864     True      
______________________________________________________________________
BatchNorm2d          [1, 64, 180, 240]    128        True      
______________________________________________________________________
Conv2d               [1, 64, 180, 240]    36,864     True      
______________________________________________________________________
BatchNorm2d          [1, 64, 180, 240]    128        True      
______________________________________________________________________
ReLU                 [1, 64, 180, 240]    0          False     
______________________________________________________________________
Conv2d               [1, 64, 180, 240]    36,864     True      
______________________________________________________________________
BatchNorm2d          [1, 64, 180, 240]    128        True      
______________________________________________________________________
Conv2d               [1, 128, 90, 120]    73,728     True      
______________________________________________________________________
BatchNorm2d          [1, 128, 90, 120]    256        True      
______________________________________________________________________
ReLU                 [1, 128, 90, 120]    0          False     
______________________________________________________________________
Conv2d               [1, 128, 90, 120]    147,456    True      
______________________________________________________________________
BatchNorm2d          [1, 128, 90, 120]    256        True      
______________________________________________________________________
Conv2d               [1, 128, 90, 120]    8,192      True      
______________________________________________________________________
BatchNorm2d          [1, 128, 90, 120]    256        True      
______________________________________________________________________
Conv2d               [1, 128, 90, 120]    147,456    True      
______________________________________________________________________
```

```
BatchNorm2d          [1, 128, 90, 120]    256        True      
______________________________________________________________________
ReLU                 [1, 128, 90, 120]    0          False     
______________________________________________________________________
Conv2d               [1, 128, 90, 120]    147,456    True      
______________________________________________________________________
BatchNorm2d          [1, 128, 90, 120]    256        True      
______________________________________________________________________
Conv2d               [1, 128, 90, 120]    147,456    True      
______________________________________________________________________
BatchNorm2d          [1, 128, 90, 120]    256        True      
______________________________________________________________________
ReLU                 [1, 128, 90, 120]    0          False     
______________________________________________________________________
Conv2d               [1, 128, 90, 120]    147,456    True      
______________________________________________________________________
BatchNorm2d          [1, 128, 90, 120]    256        True      
______________________________________________________________________
Conv2d               [1, 128, 90, 120]    147,456    True      
______________________________________________________________________
BatchNorm2d          [1, 128, 90, 120]    256        True      
______________________________________________________________________
ReLU                 [1, 128, 90, 120]    0          False     
______________________________________________________________________
Conv2d               [1, 128, 90, 120]    147,456    True      
______________________________________________________________________
BatchNorm2d          [1, 128, 90, 120]    256        True      
______________________________________________________________________
Conv2d               [1, 256, 45, 60]     294,912    True      
______________________________________________________________________
BatchNorm2d          [1, 256, 45, 60]     512        True      
______________________________________________________________________
ReLU                 [1, 256, 45, 60]     0          False     
______________________________________________________________________
Conv2d               [1, 256, 45, 60]     589,824    True      
______________________________________________________________________
BatchNorm2d          [1, 256, 45, 60]     512        True      
______________________________________________________________________
Conv2d               [1, 256, 45, 60]     32,768     True      
______________________________________________________________________
BatchNorm2d          [1, 256, 45, 60]     512        True      
______________________________________________________________________
Conv2d               [1, 256, 45, 60]     589,824    True      
______________________________________________________________________
BatchNorm2d          [1, 256, 45, 60]     512        True      
______________________________________________________________________
ReLU                 [1, 256, 45, 60]     0          False     
______________________________________________________________________
Conv2d               [1, 256, 45, 60]     589,824    True      
______________________________________________________________________
BatchNorm2d          [1, 256, 45, 60]     512        True      
______________________________________________________________________
Conv2d               [1, 256, 45, 60]     589,824    True      
______________________________________________________________________
BatchNorm2d          [1, 256, 45, 60]     512        True      
______________________________________________________________________
ReLU                 [1, 256, 45, 60]     0          False     
```

Conv2d	[1, 256, 45, 60]	589,824	True
BatchNorm2d	[1, 256, 45, 60]	512	True
Conv2d	[1, 256, 45, 60]	589,824	True
BatchNorm2d	[1, 256, 45, 60]	512	True
ReLU	[1, 256, 45, 60]	0	False
Conv2d	[1, 256, 45, 60]	589,824	True
BatchNorm2d	[1, 256, 45, 60]	512	True
Conv2d	[1, 256, 45, 60]	589,824	True
BatchNorm2d	[1, 256, 45, 60]	512	True
ReLU	[1, 256, 45, 60]	0	False
Conv2d	[1, 256, 45, 60]	589,824	True
BatchNorm2d	[1, 256, 45, 60]	512	True
Conv2d	[1, 256, 45, 60]	589,824	True
BatchNorm2d	[1, 256, 45, 60]	512	True
ReLU	[1, 256, 45, 60]	0	False
Conv2d	[1, 256, 45, 60]	589,824	True
BatchNorm2d	[1, 256, 45, 60]	512	True
Conv2d	[1, 512, 23, 30]	1,179,648	True
BatchNorm2d	[1, 512, 23, 30]	1,024	True
ReLU	[1, 512, 23, 30]	0	False
Conv2d	[1, 512, 23, 30]	2,359,296	True
BatchNorm2d	[1, 512, 23, 30]	1,024	True
Conv2d	[1, 512, 23, 30]	131,072	True
BatchNorm2d	[1, 512, 23, 30]	1,024	True
Conv2d	[1, 512, 23, 30]	2,359,296	True
BatchNorm2d	[1, 512, 23, 30]	1,024	True
ReLU	[1, 512, 23, 30]	0	False
Conv2d	[1, 512, 23, 30]	2,359,296	True

BatchNorm2d	[1, 512, 23, 30]	1,024	True
Conv2d	[1, 512, 23, 30]	2,359,296	True
BatchNorm2d	[1, 512, 23, 30]	1,024	True
ReLU	[1, 512, 23, 30]	0	False
Conv2d	[1, 512, 23, 30]	2,359,296	True
BatchNorm2d	[1, 512, 23, 30]	1,024	True
BatchNorm2d	[1, 512, 23, 30]	1,024	True
ReLU	[1, 512, 23, 30]	0	False
Conv2d	[1, 1024, 23, 30]	4,719,616	True
ReLU	[1, 1024, 23, 30]	0	False
Conv2d	[1, 512, 23, 30]	4,719,104	True
ReLU	[1, 512, 23, 30]	0	False
Conv2d	[1, 1024, 23, 30]	525,312	True
PixelShuffle	[1, 256, 46, 60]	0	False
ReplicationPad2d	[1, 256, 47, 61]	0	False
AvgPool2d	[1, 256, 46, 60]	0	False
ReLU	[1, 1024, 23, 30]	0	False
BatchNorm2d	[1, 256, 45, 60]	512	True
Conv2d	[1, 512, 45, 60]	2,359,808	True
ReLU	[1, 512, 45, 60]	0	False
Conv2d	[1, 512, 45, 60]	2,359,808	True
ReLU	[1, 512, 45, 60]	0	False
ReLU	[1, 512, 45, 60]	0	False
Conv2d	[1, 1024, 45, 60]	525,312	True
PixelShuffle	[1, 256, 90, 120]	0	False
ReplicationPad2d	[1, 256, 91, 121]	0	False
AvgPool2d	[1, 256, 90, 120]	0	False
ReLU	[1, 1024, 45, 60]	0	False
BatchNorm2d	[1, 128, 90, 120]	256	True

```
Conv2d               [1, 384, 90, 120]    1,327,488  True
ReLU                 [1, 384, 90, 120]    0          False
Conv2d               [1, 384, 90, 120]    1,327,488  True
ReLU                 [1, 384, 90, 120]    0          False
ReLU                 [1, 384, 90, 120]    0          False
Conv2d               [1, 768, 90, 120]    295,680    True
PixelShuffle         [1, 192, 180, 240]   0          False
ReplicationPad2d     [1, 192, 181, 241]   0          False
AvgPool2d            [1, 192, 180, 240]   0          False
ReLU                 [1, 768, 90, 120]    0          False
BatchNorm2d          [1, 64, 180, 240]    128        True
Conv2d               [1, 256, 180, 240]   590,080    True
ReLU                 [1, 256, 180, 240]   0          False
Conv2d               [1, 256, 180, 240]   590,080    True
ReLU                 [1, 256, 180, 240]   0          False
ReLU                 [1, 256, 180, 240]   0          False
Conv2d               [1, 512, 180, 240]   131,584    True
PixelShuffle         [1, 128, 360, 480]   0          False
ReplicationPad2d     [1, 128, 361, 481]   0          False
AvgPool2d            [1, 128, 360, 480]   0          False
ReLU                 [1, 512, 180, 240]   0          False
BatchNorm2d          [1, 64, 360, 480]    128        True
Conv2d               [1, 96, 360, 480]    165,984    True
ReLU                 [1, 96, 360, 480]    0          False
Conv2d               [1, 96, 360, 480]    83,040     True
ReLU                 [1, 96, 360, 480]    0          False
ReLU                 [1, 192, 360, 480]   0          False
Conv2d               [1, 384, 360, 480]   37,248     True
```

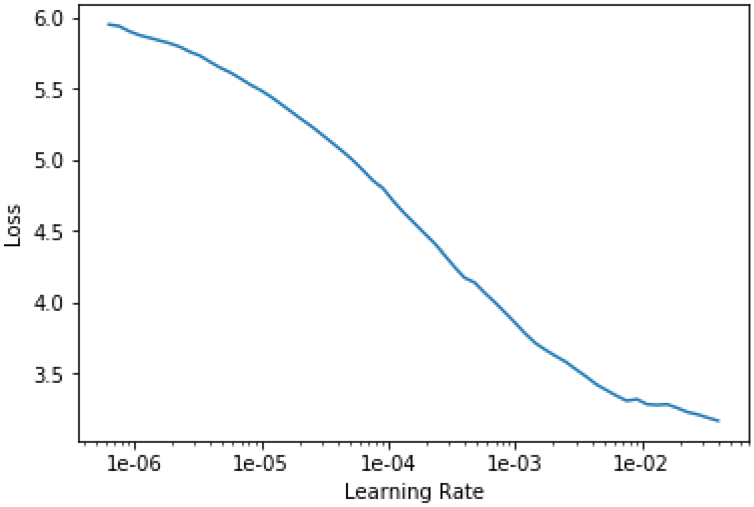
```
PixelShuffle         [1, 96, 720, 960]    0          False
______________________________________________________________________
ReplicationPad2d     [1, 96, 721, 961]    0          False
______________________________________________________________________
AvgPool2d            [1, 96, 720, 960]    0          False
______________________________________________________________________
ReLU                 [1, 384, 360, 480]   0          False
______________________________________________________________________
MergeLayer           [1, 99, 720, 960]    0          False
______________________________________________________________________
Conv2d               [1, 99, 720, 960]    88,308     True
______________________________________________________________________
ReLU                 [1, 99, 720, 960]    0          False
______________________________________________________________________
Conv2d               [1, 99, 720, 960]    88,308     True
______________________________________________________________________
ReLU                 [1, 99, 720, 960]    0          False
______________________________________________________________________
MergeLayer           [1, 99, 720, 960]    0          False
______________________________________________________________________
Conv2d               [1, 32, 720, 960]    3,200      True
______________________________________________________________________

Total params: 41,224,168
Total trainable params: 41,224,168
Total non-trainable params: 0
```

In [21]:

```
lr_find(learn)
learn.recorder.plot()
```

```
LR Finder is complete, type {learner_name}.recorder.plot() to see the graph.
```



In [22]:

```
lr=3e-3
```

In [23]:

```
learn.fit_one_cycle(10, slice(lr), pct_start=0.9)
```

Total time: 08:13

epoch	train_loss	valid_loss	acc_camvid	time
1	0.963349	0.751218	0.814461	00:50
2	0.736162	0.616474	0.845585	00:49
3	0.656558	0.519377	0.861838	00:49
4	0.656998	0.504336	0.860156	00:49
5	0.595636	0.411735	0.891830	00:49
6	0.613298	0.498074	0.867209	00:49
7	0.549370	0.442067	0.877282	00:49
8	0.530588	0.447959	0.875017	00:49
9	0.524049	0.533025	0.845427	00:49
10	0.433424	0.323676	0.902110	00:49

In [24]:

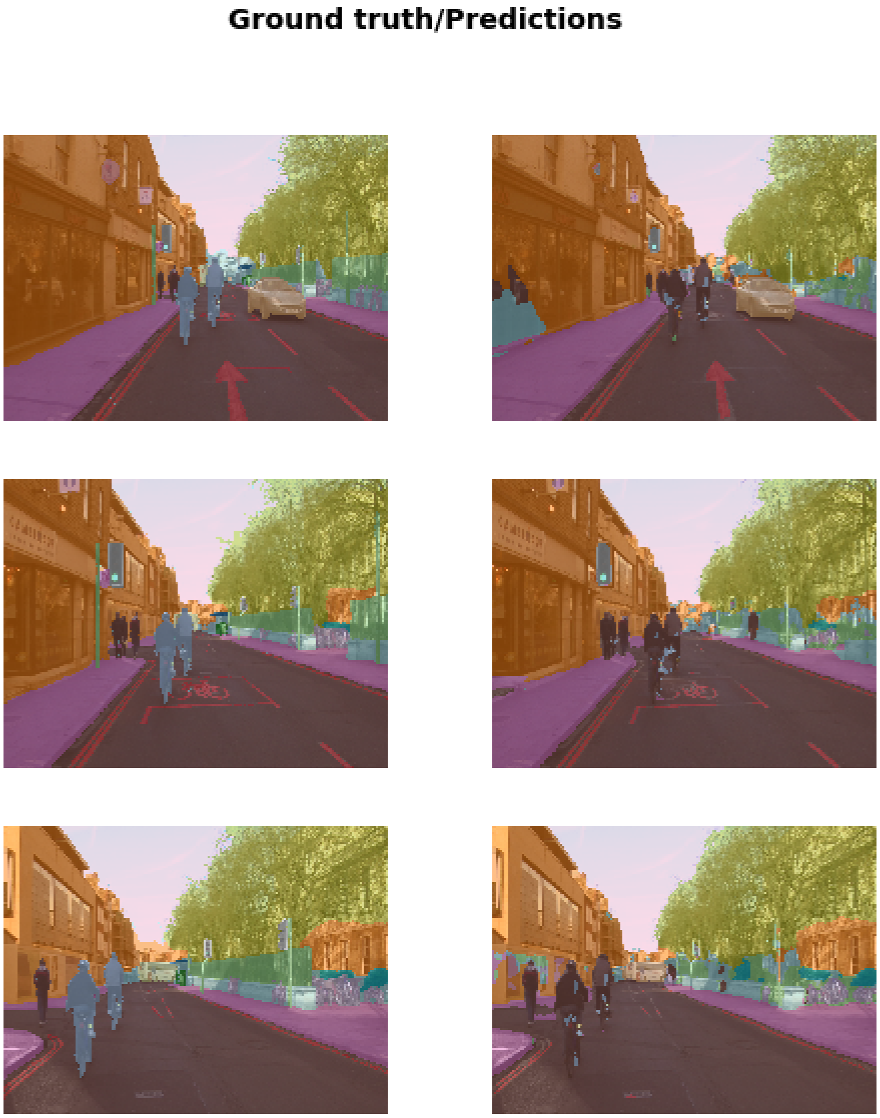
```
learn.save('stage-1')
```

In [25]:

```
learn.load('stage-1');
```

In [26]:

```
learn.show_results(rows=3, figsize=(8,9))
```



Ground truth/Predictions

In [27]:

```
learn.unfreeze()
```

In [28]:

```
lrs = slice(lr/400,lr/4)
```

In [29]:
```
learn.fit_one_cycle(12, lrs, pct_start=0.8)
```

Total time: 10:21

epoch	train_loss	valid_loss	acc_camvid	time
1	0.378445	0.321333	0.901552	00:51
2	0.377286	0.310525	0.906221	00:51
3	0.364139	0.298323	0.915841	00:51
4	0.357998	0.304364	0.911059	00:51
5	0.354306	0.291636	0.917309	00:51
6	0.334062	0.294054	0.915953	00:51
7	0.334089	0.284347	0.921153	00:51
8	0.329408	0.288293	0.922269	00:51
9	0.324237	0.292824	0.921247	00:51
10	0.304223	0.279643	0.923053	00:51
11	0.297298	0.259105	0.928049	00:51
12	0.262606	0.262566	0.926647	00:51

In [30]:
```
learn.save('stage-2');
```

Go big

You may have to restart your kernel and come back to this stage if you run out of memory, and may also need to decrease `bs`.

In [31]:
```
import fastai; fastai.__version__
```

Out[31]:
```
'1.0.46'
```

In [32]:
```
learn.destroy() # uncomment once 1.0.46 is out

size = src_size

free = gpu_mem_get_free_no_cache()
# the max size of bs depends on the available GPU RAM
if free > 8200: bs=3
else:           bs=1
print(f"using bs={bs}, have {free}MB of GPU RAM free")
```

```
this Learner object self-destroyed - it still exists, but no longer usable
using bs=1, have 7198MB of GPU RAM free
```

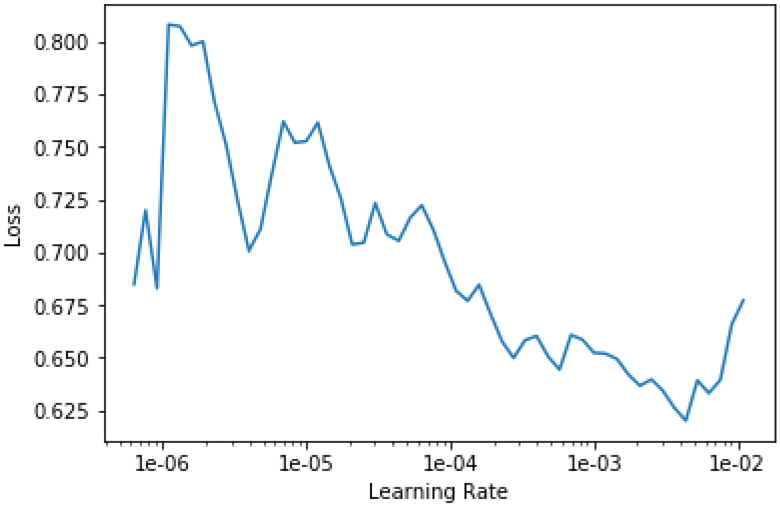
```
In [33]: data = (src.transform(get_transforms(), size=size, tfm_y=True)
                 .databunch(bs=bs)
                 .normalize(imagenet_stats))
```

```
In [34]: learn = unet_learner(data, models.resnet34, metrics=metrics, wd=wd)
```

```
In [35]: learn.load('stage-2');
```

```
In [36]: lr_find(learn)
         learn.recorder.plot()
```

```
LR Finder is complete, type {learner_name}.recorder.plot() to see the graph.
```



```
In [37]: lr=1e-3
```

In [38]:
```
learn.fit_one_cycle(10, slice(lr), pct_start=0.8)
```

Total time: 32:37

epoch	train_loss	valid_loss	acc_camvid	time
1	0.451209	0.332734	0.907046	03:15
2	0.373687	0.323635	0.909574	03:15
3	0.358684	0.330189	0.914097	03:15
4	0.357483	0.383490	0.895552	03:15
5	0.364380	0.344437	0.904873	03:15
6	0.344527	0.386376	0.903012	03:15
7	0.355646	0.308764	0.914824	03:15
8	0.370582	0.421195	0.904338	03:15
9	0.293038	0.313249	0.919430	03:15
10	0.242236	0.261322	0.928538	03:15

In [39]:
```
learn.save('stage-1-big')
```

In [40]:
```
learn.load('stage-1-big');
```

In [41]:
```
learn.unfreeze()
```

In [42]:
```
lrs = slice(1e-6,lr/10)
```

In [43]:

```
learn.fit_one_cycle(10, lrs)
```

Total time: 34:10

epoch	train_loss	valid_loss	acc_camvid	time
1	0.214535	0.269964	0.927733	03:23
2	0.217715	0.259375	0.929636	03:24
3	0.207981	0.260343	0.929338	03:25
4	0.232637	0.262718	0.928548	03:25
5	0.197295	0.259011	0.930244	03:25
6	0.223782	0.285691	0.925848	03:25
7	0.198858	0.263976	0.928670	03:25
8	0.190397	0.290263	0.925659	03:25
9	0.189331	0.279614	0.927386	03:25
10	0.197763	0.276660	0.927477	03:25

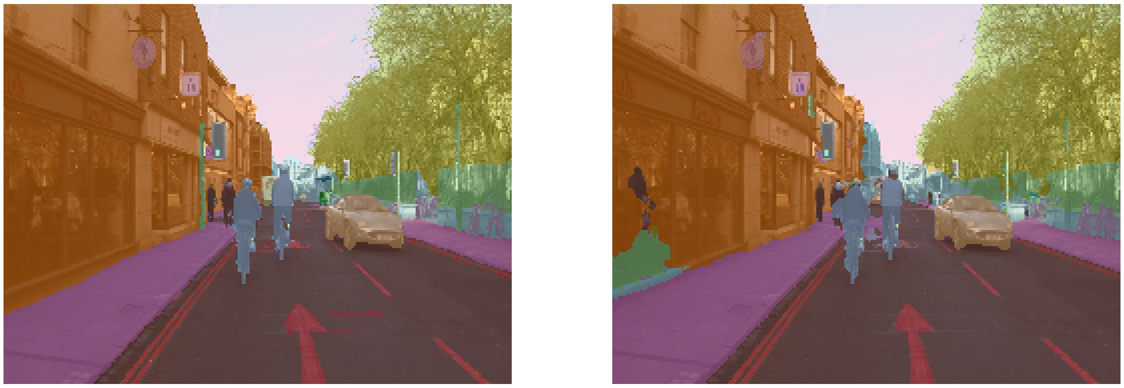
In [44]:

```
learn.save('stage-2-big')
```

In [45]:

```
learn.load('stage-2-big');
```

In [46]:
```
learn.show_results(rows=3, figsize=(10,10))
```



Ground truth/Predictions

fin