In [11]:
```
import numpy as np
import os
import pickle
from glob import glob
from tqdm import tqdm
import time
from pprint import pprint as pp
from pathlib import Path
import shutil
import random
from matplotlib import pyplot as plt
import subprocess
```

In [12]:
```
def xyxy2xywhn(x, w=1360, h=800):
    y = np.copy(x)
    y[0] = ((x[0] + x[2]) / 2) / w  # x center
    y[1] = ((x[1] + x[3]) / 2) / h  # y center
    y[2] = (x[2] - x[0]) / w  # width
    y[3] = (x[3] - x[1]) / h  # height
    return y

def save_label(fname, lines):
    with open(fname,'w') as fl:
        for ll in lines:
            fl.write("{} {} {} {} {}\n".format(*ll))

def make_dirs(dir):
    if not os.path.exists(dir):
        os.makedirs(dir)
```

Download data and set paths

In [ ]:
```
dataroot = "/home/ubuntu/Shlok/gtsdb/data"
filename = "FullIJCNN2013.zip"
urlroot = "https://sid.erda.dk/public/archives/ff17dc924eba88d5d01a807357d
! wget {urlroot}/{filename} -O {dataroot}/{filename}
! unzip {dataroot}/{filename}
```

Prepare labels Full

In [4]:
```
gtpath = os.path.join(dataroot, "FullIJCNN2013", "gt.txt")
gtlist = open(gtpath).read().split('\n')
gtlist[:10]
```

Out[4]:
```
['00000.ppm;774;411;815;446;11',
 '00001.ppm;983;388;1024;432;40',
 '00001.ppm;386;494;442;552;38',
 '00001.ppm;973;335;1031;390;13',
 '00002.ppm;892;476;1006;592;39',
 '00003.ppm;742;443;765;466;4',
 '00003.ppm;742;466;764;489;9',
 '00003.ppm;737;412;769;443;21',
 '00004.ppm;898;342;967;409;21',
 '00004.ppm;906;407;955;459;2']
```

In [5]:
```
gtdict = {}
for line in gtlist:
    if line=="":
        break
    cols = line.split(';')
    fname = cols[0]
    cls_id = int(cols[-1])
    bbox = list(map(lambda x: float(x), cols[1:5]))
    bbox = xyxy2xywhn(bbox)
    if fname in gtdict:
        gtdict[fname].append((cls_id, *bbox))
    else:
        gtdict[fname] = [(cls_id, *bbox)]


split = "full"
labelroot = os.path.join(dataroot, split, "labels")
make_dirs(labelroot)

for key in tqdm(gtdict):
    path = os.path.join(labelroot, str(Path(key).stem)+".txt")
    save_label(path, gtdict[key])
```

```
100%|██████████| 741/741 [00:00<00:00, 8773.48it/s]
```

Copy images full

In [6]:
```
images_path = os.path.join(dataroot, "FullIJCNN2013")
imglist = glob(images_path+"/*.ppm")

split = "full"
imgroot = os.path.join(dataroot, split, "images")
make_dirs(imgroot)

for img in tqdm(imglist):
    shutil.copy(img, imgroot)
```

```
100%|██████████| 900/900 [00:21<00:00, 42.13it/s]
```

Prepare train-valid split

Take first 600 images for train and last 300 images for valid.

Random train-valid split is not a good idea here since

In [16]:
```
full_imglist = glob(os.path.join(dataroot, "full", "images")+"/*.ppm")
full_imglist.sort()
print(len(full_imglist)); full_imglist[:5]
```

```
900
```

Out[16]:
```
['/home/ubuntu/Shlok/gtsdb/data/full/images/00000.ppm',
 '/home/ubuntu/Shlok/gtsdb/data/full/images/00001.ppm',
 '/home/ubuntu/Shlok/gtsdb/data/full/images/00002.ppm',
 '/home/ubuntu/Shlok/gtsdb/data/full/images/00003.ppm',
 '/home/ubuntu/Shlok/gtsdb/data/full/images/00004.ppm']
```

Prepare train split

In [17]:
```
split = "train"
imglist = full_imglist[:600]

imgroot = os.path.join(dataroot, split, "images")
labelroot = os.path.join(dataroot, split, "labels")
make_dirs(imgroot)
make_dirs(labelroot)
for img in tqdm(imglist):
    p = Path(img)
    src_label = str(p.parent.parent/"labels"/p.stem)+".txt"
    shutil.copy(img, imgroot)
    try:
        shutil.copy(src_label, labelroot)
    except Exception as err:
        pass
```

```
100%|██████████| 600/600 [00:01<00:00, 304.78it/s]
```

In [20]:
```
split = "valid"
imglist = full_imglist[600:]

imgroot = os.path.join(dataroot, split, "images")
labelroot = os.path.join(dataroot, split, "labels")
make_dirs(imgroot)
make_dirs(labelroot)
for img in tqdm(imglist):
    p = Path(img)
    src_label = str(p.parent.parent/"labels"/p.stem)+".txt"
    shutil.copy(img, imgroot)
    try:
        shutil.copy(src_label, labelroot)
    except Exception as err:
        pass
```

```
100%|██████████| 300/300 [00:00<00:00, 305.00it/s]
```

Prepare classes name

In [21]:
```
classnames = open(os.path.join(dataroot, "classes.txt")).read().split("\n"
classnames[:4]
```

Out[21]:
```
['0 = speed limit 20 (prohibitory)',
 '1 = speed limit 30 (prohibitory)',
 '2 = speed limit 50 (prohibitory)',
 '3 = speed limit 60 (prohibitory)']
```

In [22]:
```
classnames = list(map(lambda x: x.split('=')[1].split('(')[0].strip(), cla
print(classnames)
```

```
['speed limit 20', 'speed limit 30', 'speed limit 50', 'speed limit 60', 's
peed limit 70', 'speed limit 80', 'restriction ends 80', 'speed limit 100',
'speed limit 120', 'no overtaking', 'no overtaking', 'priority at next inte
rsection', 'priority road', 'give way', 'stop', 'no traffic both ways', 'no
trucks', 'no entry', 'danger', 'bend left', 'bend right', 'bend', 'uneven r
oad', 'slippery road', 'road narrows', 'construction', 'traffic signal', 'p
edestrian crossing', 'school crossing', 'cycles crossing', 'snow', 'animals
', 'restriction ends', 'go right', 'go left', 'go straight', 'go right or s
traight', 'go left or straight', 'keep right', 'keep left', 'roundabout', '
restriction ends', 'restriction ends']
```

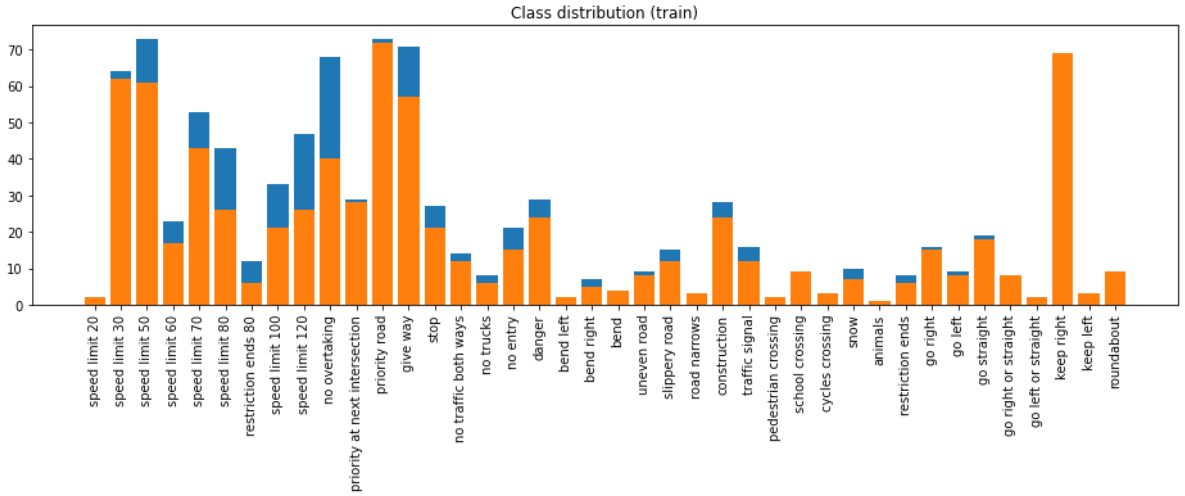
Plot class distribution

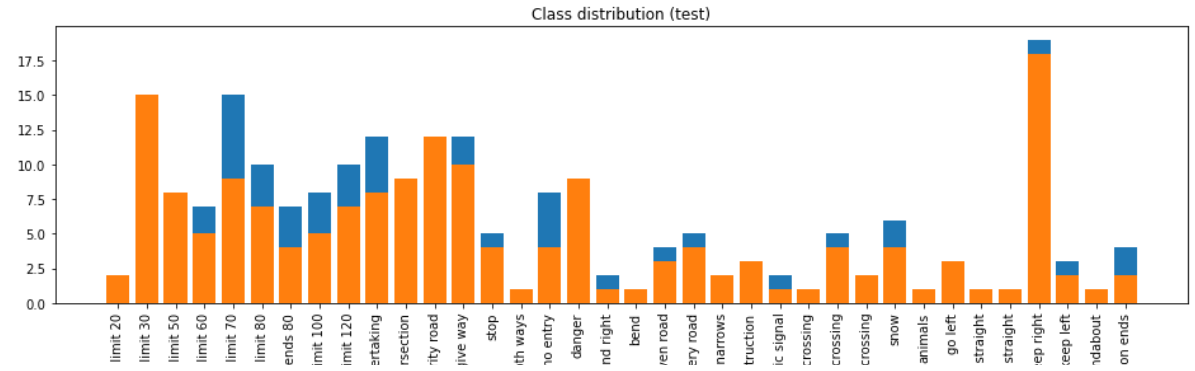
In [23]:
```
full_labelist = glob(os.path.join(dataroot, "full", "labels")+"/*.txt")
full_labelist[:5]
```

Out[23]:
```
['/home/ubuntu/Shlok/gtsdb/data/full/labels/00281.txt',
 '/home/ubuntu/Shlok/gtsdb/data/full/labels/00305.txt',
 '/home/ubuntu/Shlok/gtsdb/data/full/labels/00112.txt',
 '/home/ubuntu/Shlok/gtsdb/data/full/labels/00644.txt',
 '/home/ubuntu/Shlok/gtsdb/data/full/labels/00374.txt']
```

In [24]:

```
names = np.array(['speed limit 20', 'speed limit 30', 'speed limit 50', 'sp
          'restriction ends 80', 'speed limit 100', 'speed limit 120', 'no
          'priority at next intersection', 'priority road', 'give way', 'st
          'no entry', 'danger', 'bend left', 'bend right', 'bend', 'uneven
          'construction', 'traffic signal', 'pedestrian crossing', 'school
          'animals', 'restriction ends', 'go right', 'go left', 'go straigh
          'keep right', 'keep left', 'roundabout', 'restriction ends', 'res

def plot_distribution(labelist, split):
    label_to_files = {}
    for lblfile in labelist:
        lines = open(lblfile).read().split('\n')[:-1]
        labels = list(map(lambda x: x.split()[0], lines))
        for label in labels:
            label = int(label)
            if label not in label_to_files:
                label_to_files[label] = [lblfile]
            else:
                label_to_files[label].append(lblfile)

    label_to_no = {_k:len(label_to_files[_k]) for _k in label_to_files}
    # pp(label_to_no)
    _ltn = sorted(list(label_to_no.items()), key=lambda x: x[0])
    _ltnc = [x[0] for x in _ltn]
    _ltnn = [x[1] for x in _ltn]

    label_to_no_set = {_k:len(set(label_to_files[_k])) for _k in label_to_
    _ltns = sorted(list(label_to_no_set.items()), key=lambda x: x[0])
    _ltncs = [x[0] for x in _ltns]
    _ltnns = [x[1] for x in _ltns]

    plt.figure(figsize=(16,4))
    plt.bar(names[_ltnc], _ltnn)
    plt.xticks(rotation=90)
    plt.bar(names[_ltncs], _ltnns)
    plt.xticks(rotation=90)
    plt.title("Class distribution ({})".format(split))
    plt.show()

plot_distribution(full_labelist[:600], "train")
plot_distribution(full_labelist[600:], "test")
```





In [ ]: