In [1]:
```
import os
import matplotlib.pyplot as plt
import pandas as pd
```

In [2]:
```
train_path = 'dataset/train'
test_path = 'dataset/test'
```

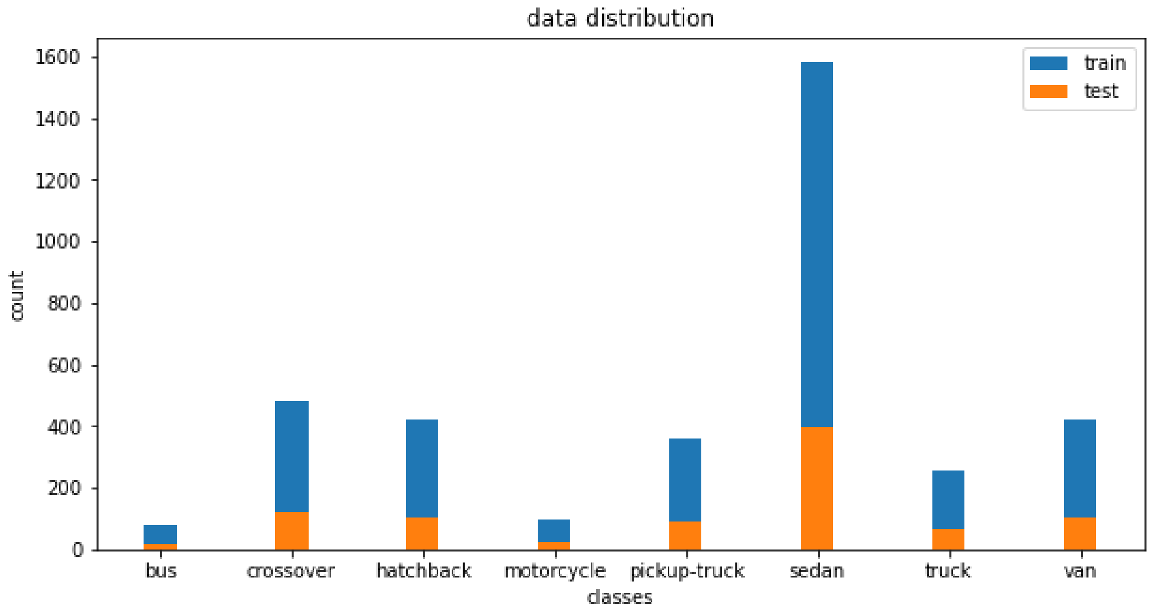
In [3]:
```
classes = os.listdir(train_path)
classes2 = os.listdir(test_path)
assert classes == classes2
classes
```

Out[3]:
```
['bus',
 'crossover',
 'hatchback',
 'motorcycle',
 'pickup-truck',
 'sedan',
 'truck',
 'van']
```

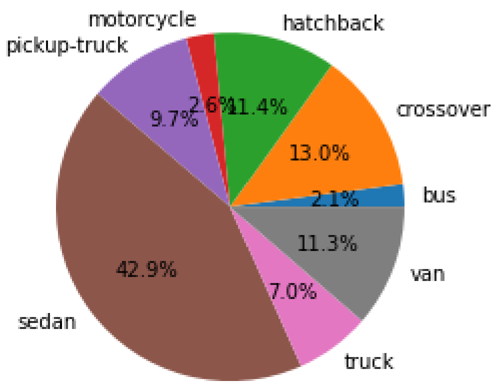
In [4]:
```
classes_count = {}
y = []
for path in [train_path, test_path]:
    counts = []
    for c in classes:
        class_path = os.path.join(path, c)
        count = len(os.listdir(class_path))
        classes_count[c] = count
        counts.append(count)
    y.append(counts)
    print('In {} classes count are:\n'.format(path[path.find('/')+1
:]))
    print(classes_count, '\n\n')
```

```
In train classes count are:

{'bus': 77, 'crossover': 480, 'hatchback': 419, 'motorcycle': 95, 'p
ickup-truck': 357, 'sedan': 1581, 'truck': 258, 'van': 418}


In test classes count are:

{'bus': 20, 'crossover': 120, 'hatchback': 105, 'motorcycle': 22, 'p
ickup-truck': 90, 'sedan': 396, 'truck': 65, 'van': 105}
```

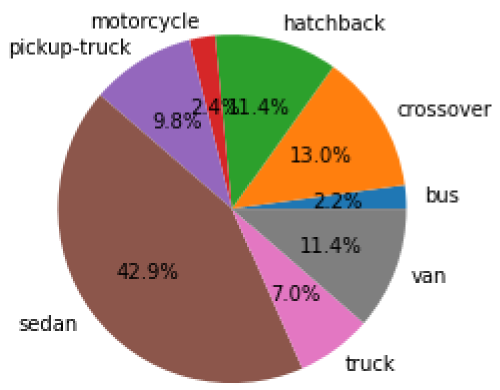
In [22]:
```
fig = plt.figure(figsize = (10, 5))
plt.bar(classes, y[0], width=0.25, label='train')
plt.bar(classes, y[1], width=0.25, label='test')
plt.xlabel('classes')
plt.ylabel('count')
plt.title('data distribution')
plt.legend()
plt.show()
```



In [30]:
```
plt.pie(y[0], labels=classes, autopct='%1.1f%%')
plt.show()
```



```
In [29]: plt.pie(y[1], labels=classes, autopct='%1.1f%%')
         plt.show()
```



## As seen from the above bar plot and both pie charts, the distribution of data among training and testing datasets are almot alike, which can be seen clearly from the percentage of classes among each in the pie charts

## However, it can be seen that the data is biased towards some classes more thatn others

--

**Methods to fix it:**

1. Augmenting less existing labels
2. In training taking weighted loss into consideration
3. Don't take accuracy metric for granted

# I will prepare seperate CSV files for train/val/test for ease of usage.

#### I know I could use Dataset ImageFolder right away and seperate train/val with indices. But I find this method easier for usage and user understanding

In [5]:
```
classes_dist = {'bus': 77, 'crossover': 480, 'hatchback': 419, 'moto
rcycle': 95, 'pickup-truck': 357, 'sedan': 1581, 'truck': 258, 'van'
: 418}
train_dist = {}
val_dist = {}
for c in classes_dist:
    train_dist[c] = int(0.8 * classes_dist[c])
    val_dist[c] = classes_dist[c] - train_dist[c]
```

In [6]:
```
train_dist
```

Out[6]:
```
{'bus': 61,
 'crossover': 384,
 'hatchback': 335,
 'motorcycle': 76,
 'pickup-truck': 285,
 'sedan': 1264,
 'truck': 206,
 'van': 334}
```

In [7]:
```
val_dist
```

Out[7]:
```
{'bus': 16,
 'crossover': 96,
 'hatchback': 84,
 'motorcycle': 19,
 'pickup-truck': 72,
 'sedan': 317,
 'truck': 52,
 'van': 84}
```

In [14]:
```
train_data = []
val_data = []
```

In [15]:
```
for c in classes:
    path = os.path.join(train_path, c)
    files = os.listdir(path)
    counter = 0
    for i, f in enumerate(files):
        file_path = os.path.join(path, f)
        if counter < train_dist[c]:
            train_data.append([file_path, c])
        else:
            val_data.append([file_path, c])
        counter += 1
```

In [22]:
```
train_csv = pd.DataFrame(train_data, columns=['path', 'class']).samp
le(frac=1)
val_csv = pd.DataFrame(val_data, columns=['path', 'class']).sample(f
rac=1)
```

In [25]:
```
train_csv
```

Out[25]:

| | path | class |
|---|---|---|
| **2572** | dataset/train/truck/truck-front (177).jpg | truck |
| **548** | dataset/train/hatchback/hatchback-back (447).jpg | hatchback |
| **2433** | dataset/train/truck/truck-back (2).jpg | truck |
| **940** | dataset/train/pickup-truck/pickup-back (205).jpg | pickup-truck |
| **1609** | dataset/train/sedan/sedan-front (1045).jpg | sedan |
| **...** | ... | ... |
| **351** | dataset/train/crossover/crossover-front (258).jpg | crossover |
| **1552** | dataset/train/sedan/sedan-back (458).jpg | sedan |
| **2004** | dataset/train/sedan/sedan-front (1647).jpg | sedan |
| **2459** | dataset/train/truck/truck-back (63).jpg | truck |
| **2069** | dataset/train/sedan/sedan-front (1754).jpg | sedan |

2945 rows × 2 columns

In [26]:
```
val_csv
```

Out[26]:

| | path | class |
|---|---|---|
| **125** | dataset/train/hatchback/hatchback-front (33).jpg | hatchback |
| **254** | dataset/train/pickup-truck/pickup-front (53).jpg | pickup-truck |
| **123** | dataset/train/hatchback/hatchback-front (328).jpg | hatchback |
| **7** | dataset/train/bus/bus-front (70).jpg | bus |
| **347** | dataset/train/sedan/sedan-front (615).jpg | sedan |
| **...** | ... | ... |
| **285** | dataset/train/pickup-truck/pickup-front (98).jpg | pickup-truck |
| **670** | dataset/train/van/van-front (292).jpg | van |
| **456** | dataset/train/sedan/sedan-front (79).jpg | sedan |
| **187** | dataset/train/hatchback/hatchback-front (87).jpg | hatchback |
| **40** | dataset/train/crossover/crossover-front (427).jpg | crossover |

740 rows × 2 columns

In [27]:
```
train_csv.to_csv('dataset/train.csv', index=False)
val_csv.to_csv('dataset/val.csv', index=False)
```

In [32]:
```
test_data = []
```

In [33]:
```
for c in classes:
    path = os.path.join(test_path, c)
    files = os.listdir(path)
    for i, f in enumerate(files):
        file_path = os.path.join(path, f)
        test_data.append([file_path, c])
```

In [34]:
```
test_csv = pd.DataFrame(test_data, columns=['path', 'class']).sample
(frac=1)
```

In [35]:
```
test_csv
```

Out[35]:

| | path | class |
|---|---|---|
| **327** | dataset/test/pickup-truck/pickup-front (142).jpg | pickup-truck |
| **213** | dataset/test/hatchback/hatchback-back (439).jpg | hatchback |
| **263** | dataset/test/motorcycle/1_BICYCLE_15-09-03-928... | motorcycle |
| **609** | dataset/test/sedan/sedan-back (1695).jpg | sedan |
| **14** | dataset/test/bus/bus-front (40).jpg | bus |
| **...** | ... | ... |
| **51** | dataset/test/crossover/crossover-back (169).jpg | crossover |
| **393** | dataset/test/sedan/sedan-back (1098).jpg | sedan |
| **261** | dataset/test/motorcycle/1_BICYCLE_15-07-01-411... | motorcycle |
| **399** | dataset/test/sedan/sedan-back (1113).jpg | sedan |
| **345** | dataset/test/pickup-truck/pickup-front (42).jpg | pickup-truck |

923 rows × 2 columns

In [36]:
```
test_csv.to_csv('dataset/test.csv', index=False)
```