Fadilla Zennifa , Model building for pedestrian car detection

In [69]:
```
import cv2
import pickle
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from sklearn.preprocessing import LabelBinarizer, LabelEncoder
from keras.utils.np_utils import to_categorical
```

In [70]:
```
cols1 = ["frame", "xmin", "ymin", "xmax", "ymax", "occluded", "label",
"attributes"]
df1 = pd.read_csv('labels.csv', names=cols1, delimiter=' ')
df1 = df1.query("label != 'trafficLight' and label != 'biker'")
df1.drop(['occluded', 'attributes'], axis=1, inplace=True)
df1.frame = '' + df1.frame
df1 = df1.reset_index(drop=True)

cols2 = ["xmin", "ymin", "xmax", "ymax", "frame", "label"]
df2 = pd.read_csv('labels_crowdai.csv')
df2.drop('Preview URL', axis=1, inplace=True)
df2.columns = cols2
df2['label'] = df2['label'].str.lower()
df2.frame = '' + df2.frame
```

In [71]:
```
print('df1.shape', df1.shape)
print('df2.shape', df2.shape)
```

```
df1.shape (74157, 6)
df2.shape (72064, 6)
```

In [72]:
```
df1.head()
```

Out[72]:

	frame	xmin	ymin	xmax	ymax	label
0	1478019952686311006.jpg	950	574	1004	620	car
1	1478019952686311006.jpg	1748	482	1818	744	pedestrian
2	1478019953180167674.jpg	872	586	926	632	car
3	1478019953689774621.jpg	686	566	728	618	truck
4	1478019953689774621.jpg	716	578	764	622	car

In [73]:
```
df2.head()
```

Out[73]:

	xmin	ymin	xmax	ymax	frame	label
0	785	533	905	644	1479498371963069978.jpg	car
1	89	551	291	680	1479498371963069978.jpg	car
2	268	546	383	650	1479498371963069978.jpg	car
3	455	522	548	615	1479498371963069978.jpg	truck
4	548	522	625	605	1479498371963069978.jpg	truck

In [74]:
```
grp1 = df1.groupby('label').size()
grp1 / grp1.sum() * 100
```

Out[74]:
```
label
car           81.972032
pedestrian    13.304206
truck          4.723762
dtype: float64
```

In [75]:
```
grp2 = df2.groupby('label').size()
grp2 / grp2.sum() * 100
```

Out[75]:
```
label
```

```
car           86.825599
pedestrian     7.874944
truck          5.299456
dtype: float64
```

In [76]:
```
df = pd.concat([df1, df2], axis=0)
df.head()
```

Out[76]:

	frame	xmin	ymin	xmax	ymax	label
0	1478019952686311006.jpg	950	574	1004	620	car
1	1478019952686311006.jpg	1748	482	1818	744	pedestrian
2	1478019953180167674.jpg	872	586	926	632	car
3	1478019953689774621.jpg	686	566	728	618	truck
4	1478019953689774621.jpg	716	578	764	622	car

In [77]:
```
df.label = np.where(df.label == 'truck', 'car', df.label)
# df.label = np.where(df.label != 'car', 'non-car', df.label)
# df.label = df.label.astype('category')
```

In [78]:
```
df.head()
```

Out[78]:

	frame	xmin	ymin	xmax	ymax	label
0	1478019952686311006.jpg	950	574	1004	620	car
1	1478019952686311006.jpg	1748	482	1818	744	pedestrian
2	1478019953180167674.jpg	872	586	926	632	car
3	1478019953689774621.jpg	686	566	728	618	car
4	1478019953689774621.jpg	716	578	764	622	car

In [79]:
```
grp = df.groupby('label').size()
grp / grp.sum() * 100
```

Out[79]:
```
label
car           89.371568
pedestrian    10.628432
dtype: float64
```

In [80]:
```
df.shape
```

Out[80]:
```
(146221, 6)
```

In [81]:
```
df[df.xmax == 0]
```

Out[81]:

	frame	xmin	ymin	xmax	ymax	label

In [82]:
```
df[df.ymax == 0] # this is error (ymax shouldn't be zero)
```

Out[82]:

	frame	xmin	ymin	xmax	ymax	label
3051	1479498564477313399.jpg	912	0	951	0	car
5482	1479498820473341507.jpg	705	0	732	0	car
13486	1479499937073018706.jpg	721	0	751	0	car
58312	1479505030914958665.jpg	763	0	793	0	pedestrian

In [83]:
```
df = df[df.ymax != 0]
df.shape
```

Out[83]:
```
(146217, 6)
```

In [84]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 146217 entries, 0 to 72063
Data columns (total 6 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   frame   146217 non-null  object
```

```
 1   xmin    146217 non-null  int64
 2   ymin    146217 non-null  int64
 3   xmax    146217 non-null  int64
 4   ymax    146217 non-null  int64
 5   label   146217 non-null  object
dtypes: int64(4), object(2)
memory usage: 7.8+ MB
```

In [85]:
```
# classes = ['background'] + list(df.label.unique())
classes = df.label.unique()
n_classes = len(classes)
le = LabelEncoder()
le.fit(classes)
classes
```

Out[85]:
```
array(['car', 'pedestrian'], dtype=object)
```

In [86]:
```
print(le.transform(['car']))
print(le.transform(['pedestrian']))
```

```
[0]
[1]
```

In [87]:
```
print(to_categorical(le.transform(['car']), n_classes))
print(to_categorical(le.transform(['pedestrian']), n_classes))
```

```
[[1. 0.]]
[[0. 1.]]
```

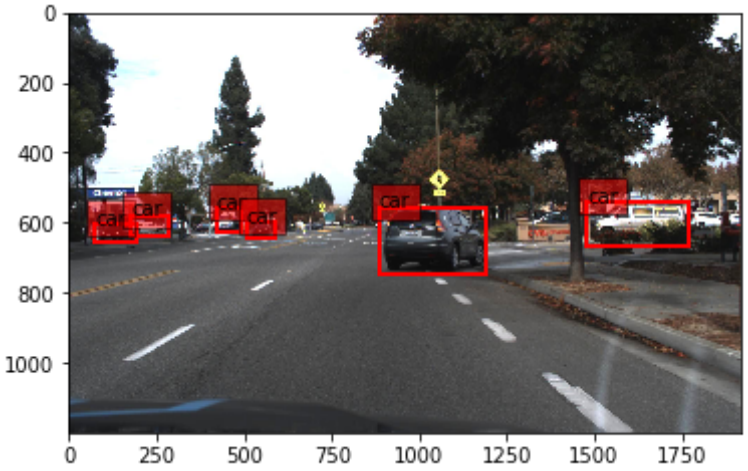
In [88]:
```
def draw(key):
    image = mpimg.imread('' + key)
    data = df[df.frame == key]
    colors = plt.cm.hsv(np.linspace(0, 0.5, len(classes))).tolist()

    ax = plt.gca()
    for idx, img in data.iterrows():
        cv2.rectangle(image, (img.xmin, img.ymin), (img.xmax, img.ymax
),(0, 255, 0), 6)
```

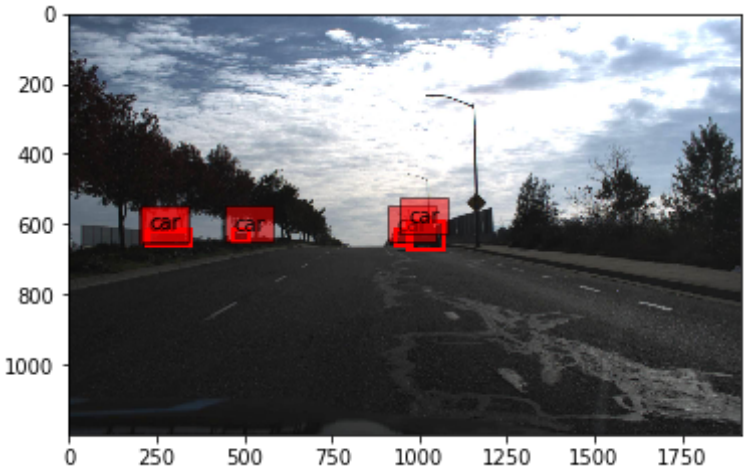
```
        coords = (img.xmin, img.ymin), img.xmax-img.xmin+1, img.ymax-im
g.ymin+1
        color = colors[le.transform([img.label])[0]]
        ax.add_patch(plt.Rectangle(*coords, fill=False, edgecolor=color
, linewidth=2))
        ax.text(img.xmin, img.ymin, img.label, bbox={'facecolor':color,
 'alpha':0.5})

    plt.imshow(image)
```

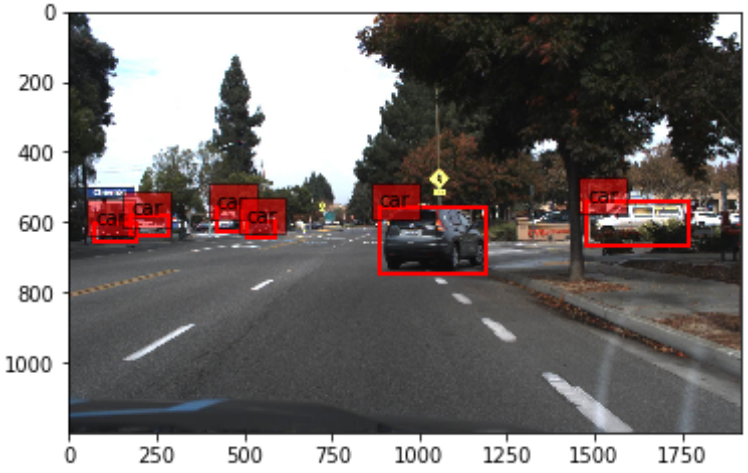
In [89]:
```
draw('14788966381499541 92.jpg')
```



In [90]:
```
draw('14788990575621459 35.jpg')
```



In [91]:
```
draw('1478896638149954192.jpg')
```



In [92]:
```
draw('1478019952686311006.jpg')
```