

## Detecting Objects from a Image:

In [17]:

```
pip install opencv-contrib-python==3.4.13.47
```

Collecting opencv-contrib-python==3.4.13.47Note: you may need to restart the kernel to use updated packages.

Downloading opencv\_contrib\_python-3.4.13.47-cp39-cp39-win\_amd64.whl (36.4 MB)  
Requirement already satisfied: numpy>=1.19.3 in c:\users\nanth\anaconda3\lib\site-packages (from opencv-contrib-python==3.4.13.47) (1.21.5)  
Installing collected packages: opencv-contrib-python  
Successfully installed opencv-contrib-python-3.4.13.47

In [3]:

```
pip install cvlib
```

```
a4c982a34053000731841a9053e2c0
Stored in directory: c:\users\nanth\appdata\local\pip\cache\wheels\96\ad\4c\d80d4bf4143c154ab297e2384915c89aff43d77d17
87b9a702
Building wheel for imutils (setup.py): started
Building wheel for imutils (setup.py): finished with status 'done'Note: you may need to restart the kernel to use upda
ted packages.
Created wheel for imutils: filename=imutils-0.5.4-py3-none-any.whl size=25872 sha256=9828abc95cafeafa66af60c356197aadb
aad49b3af0197c46b5eecbd5bd3fb48
Stored in directory: c:\users\nanth\appdata\local\pip\cache\wheels\4b\45\2d\4a070a801d3a3d93f033d3ee9728f470f514826e89
952df3ea
Building wheel for progressbar (setup.py): started
Building wheel for progressbar (setup.py): finished with status 'done'
Created wheel for progressbar: filename=progressbar-2.5-py3-none-any.whl size=12082 sha256=24cda3fe45e5b71574299e2e7f6
74741e349952eb6f713ee6a29cfcdba7bf4a3
Stored in directory: c:\users\nanth\appdata\local\pip\cache\wheels\d7\d9\89\43f31c76ff6d51dc3b1575628f59afe59e4ceae3f2
748cd7ad
Successfully built cvlib imutils progressbar
Installing collected packages: progressbar, imutils, cvlib

Successfully installed cvlib-0.2.7 imutils-0.5.4 progressbar-2.5
```

In [3]:

```
# checking the version:
import cv2
cv2.__version__
```

Out[3]:

```
'4.7.0'
```

In [20]:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import cvlib as cv
from cvlib.object_detection import draw_bbox
from numpy.lib.polynomial import poly

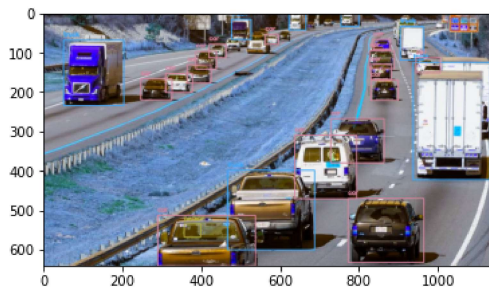
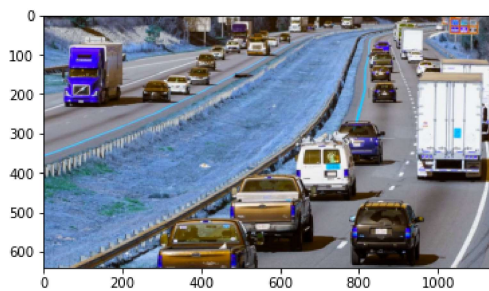
#Read the image using imread and display imshow functions
image = cv2.imread("cars2.jpg")
plt.imshow(image)
plt.show()

#Detect common objects using detect_common_objects function of cv2
box, label, count = cv.detect_common_objects(image)

#Draw the box around the detected objects and assign labels
output = draw_bbox(image, box, label, count)

#Display the image
plt.imshow(output)
plt.show()

#Display the common objects detected
print("Number of cars in this image are " +str(label.count('car'))))
print("Number of trucks in this image are " +str(label.count('truck'))))
```



Number of cars in this image are 15  
Number of trucks in this image are 7

In [21]:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import cvlib as cv
from cvlib.object_detection import draw_bbox
from numpy.lib.polynomial import poly

#Read the image using imread and display imshow functions
image = cv2.imread("cars1.jpg")
plt.imshow(image)
plt.show()

#Detect common objects using detect_common_objects function of cv2
box, label, count = cv.detect_common_objects(image)

#Draw the box around the detected objects and assign labels
output = draw_bbox(image, box, label, count)

#Display the image
plt.imshow(output)
plt.show()

#Display the common objects detected
print("Number of cars in this image are " +str(label.count('car'))))
print("Number of trucks in this image are " +str(label.count('truck'))))
```



Number of cars in this image are 8  
Number of trucks in this image are 3

In [22]:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import cvlib as cv
from cvlib.object_detection import draw_bbox
from numpy.lib.polynomial import poly

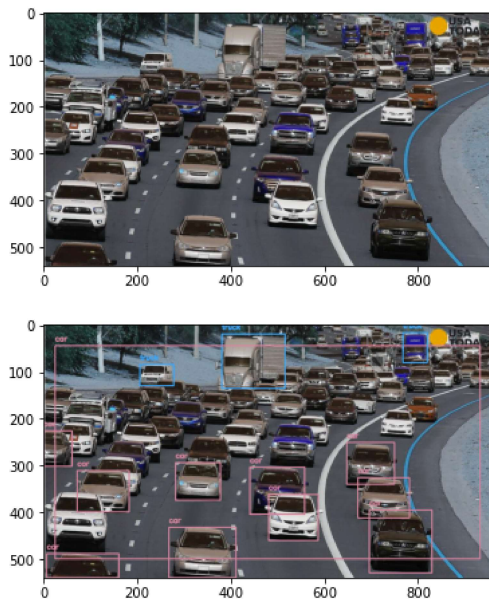
#Read the image using imread and display imshow functions
image = cv2.imread("cars.jpg")
plt.imshow(image)
plt.show()

#Detect common objects using detect_common_objects function of cv2
box, label, count = cv.detect_common_objects(image)

#Draw the box around the detected objects and assign labels
output = draw_bbox(image, box, label, count)

#Display the image
plt.imshow(output)
plt.show()

#Display the common objects detected
print("Number of cars in this image are " +str(label.count('car'))))
print("Number of trucks in this image are " +str(label.count('truck'))))
```



Number of cars in this image are 11  
Number of trucks in this image are 3

In [23]:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import cvlib as cv
from cvlib.object_detection import draw_bbox
from numpy.lib.polynomial import poly

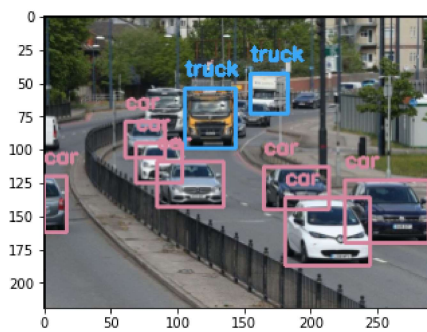
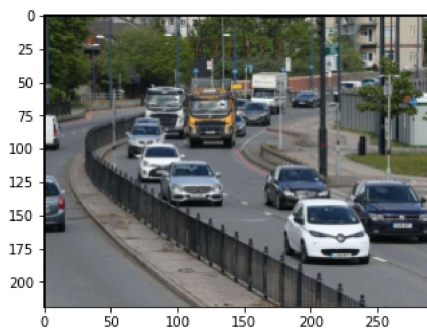
#Read the image using imread and display imshow functions
image = cv2.imread("cars3.png")
plt.imshow(image)
plt.show()

#Detect common objects using detect_common_objects function of cv2
box, label, count = cv.detect_common_objects(image)

#Draw the box around the detected objects and assign Labels
output = draw_bbox(image, box, label, count)

#Display the image
plt.imshow(output)
plt.show()

#Display the common objects detected
print("Number of cars in this image are " +str(label.count('car')))
print("Number of trucks in this image are " +str(label.count('truck')))
```



Number of cars in this image are 7  
Number of trucks in this image are 2

In [ ]: