In [29]:

```
import numpy as np
import numpy.random as npr
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('bmh')
import tensorflow as tf
from tensorflow import keras

# Loading Training Data
X_train_1 = np.load('flower_species_classification/data_train.npy').T
t_train_1 = np.load('flower_species_classification/labels_train.npy')
X_test_1 = np.load('flower_species_classification/data_test.npy').T/255.0
t_test_1 = np.load('flower_species_classification/labels_test.npy')

print(X_train_1.shape, t_train_1.shape, X_test_1.shape, t_test_1.shape)
```

```
(1658, 270000) (1658,) (415, 270000) (415,)
```

In [30]:

```
X_test_1_rs = tf.constant(X_test_1.reshape((X_test_1.shape[0],300,300,3)),
                          dtype=tf.float32)
```

In [34]:

```
model = keras.models.load_model('my_keras_model_1.h5')
```

In [35]:

```
y_test = np.argmax(model.predict(X_test_1_rs),axis=1)
```

```
13/13 [==============================] - 2s 91ms/step
```

In [36]:

```
class_names = ['Roses', 'Magnolias', 'Lilies', 'Sunflowers', 'Orchids',
               'Marigold', 'Hibiscus', 'Firebush', 'Pentas', 'Bougainvillea']
from sklearn.metrics import classification_report
print(classification_report(t_test_1, y_test, target_names=class_names))
```

```
              precision    recall  f1-score   support

       Roses       0.84      0.85      0.85        48
   Magnolias       0.93      0.95      0.94        44
      Lilies       0.80      0.76      0.78        46
  Sunflowers       1.00      0.92      0.96        36
     Orchids       0.92      0.80      0.86        45
    Marigold       0.90      0.95      0.93        40
    Hibiscus       0.82      0.95      0.88        43
    Firebush       0.92      0.95      0.93        37
      Pentas       0.91      0.91      0.91        32
Bougainvillea       0.74      0.73      0.74        44

    accuracy                           0.87       415
   macro avg       0.88      0.88      0.88       415
weighted avg       0.87      0.87      0.87       415
```

In [99]:

```
from sklearn.metrics import confusion_matrix
confusion_matrix(t_test_1, y_test)
```

Out[99]:

```
array([[41,  0,  1,  0,  0,  2,  1,  1,  0,  2],
       [ 0, 42,  1,  0,  0,  0,  0,  1,  0,  0],
       [ 1,  1, 35,  0,  2,  0,  2,  0,  0,  5],
       [ 0,  0,  0, 33,  0,  1,  0,  0,  2,  0],
       [ 2,  0,  5,  0, 36,  0,  0,  0,  0,  2],
       [ 1,  0,  0,  0,  0, 38,  0,  0,  1,  0],
       [ 1,  1,  0,  0,  0,  0, 41,  0,  0,  0],
       [ 0,  1,  0,  0,  0,  0,  1, 35,  0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  1, 29,  2],
       [ 3,  0,  2,  0,  1,  1,  5,  0,  0, 32]], dtype=int64)
```

Compared with training precision and accuracy, the model is a little overfit but still with good performance.

In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
```

In [53]:

```
from PIL import Image
import cv2 # install opencv, if you don't already have it (https://pypi.org/project/opencv-python/)
import pandas as pd
import os, time, random
import torch
from tqdm.auto import tqdm
import shutil as sh
import yaml

#from IPython.display import Image, clear_output
import utils
```

```
C:\anaconda new\anaconda\envs\gpu\lib\site-packages\tqdm\auto.py:22: TqdmWarning: IProgress not found. Please update jupyte
r and ipywidgets. See https://ipywidgets.readthedocs.io/en/stable/user_install.html (https://ipywidgets.readthedocs.io/en/s
table/user_install.html)
  from .autonotebook import tqdm as notebook_tqdm
```

In [55]:

```
!python yolov5/detect.py --classes 0 --weights car_detection_yolov5/result/trained/adamw_1/weights/best.pt --img 676 --conf 0.25 --source
```

```
\test\images\10157.jpg: 416x704 1 car, 211.8ms
image 159/175 C:\Users\User\Desktop\UF-MAE\2022 Fall\Apply Machine Learning\project-3-Danielyaoan\car_detection_yolov5
\test\images\10158.jpg: 416x704 (no detections), 199.0ms
image 160/175 C:\Users\User\Desktop\UF-MAE\2022 Fall\Apply Machine Learning\project-3-Danielyaoan\car_detection_yolov5
\test\images\10159.jpg: 416x704 (no detections), 213.0ms
image 161/175 C:\Users\User\Desktop\UF-MAE\2022 Fall\Apply Machine Learning\project-3-Danielyaoan\car_detection_yolov5
\test\images\10160.jpg: 416x704 (no detections), 215.6ms
image 162/175 C:\Users\User\Desktop\UF-MAE\2022 Fall\Apply Machine Learning\project-3-Danielyaoan\car_detection_yolov5
\test\images\10161.jpg: 416x704 (no detections), 203.6ms
image 163/175 C:\Users\User\Desktop\UF-MAE\2022 Fall\Apply Machine Learning\project-3-Danielyaoan\car_detection_yolov5
\test\images\10162.jpg: 416x704 (no detections), 205.1ms
image 164/175 C:\Users\User\Desktop\UF-MAE\2022 Fall\Apply Machine Learning\project-3-Danielyaoan\car_detection_yolov5
\test\images\10163.jpg: 416x704 (no detections), 214.0ms
image 165/175 C:\Users\User\Desktop\UF-MAE\2022 Fall\Apply Machine Learning\project-3-Danielyaoan\car_detection_yolov5
\test\images\10164.jpg: 416x704 1 car, 201.1ms
image 166/175 C:\Users\User\Desktop\UF-MAE\2022 Fall\Apply Machine Learning\project-3-Danielyaoan\car_detection_yolov5
\test\images\10165.jpg: 416x704 2 cars, 207.0ms
image 167/175 C:\Users\User\Desktop\UF-MAE\2022 Fall\Apply Machine Learning\project-3-Danielyaoan\car_detection_yolov5
\test\images\10166.jpg: 416x704 1 car, 201.1ms
```

In [90]:

```
from IPython.display import Image
test_range = range(10000,10174,1)
valid_index = npr.choice(test_range, 5, replace = False)
file_paths = []
plt.figure(figsize=(15,12))
for i in range(5):
    plt.subplot(5, 5, i + 1)
    img = plt.imread('car_detection_yolov5/result/detect/exp/{}'.format(valid_index[i])+'.jpg')
    plt.imshow(img)
    plt.title('Predictions', size=10, color="black")
    plt.xticks([])
    plt.yticks([])

plt.show()
```

Since I'm using yolov5, I decide to make use of its built-in validation function to valid the performance of test set using precision, recall, f1-score and precision-recall curve.

In [92]:

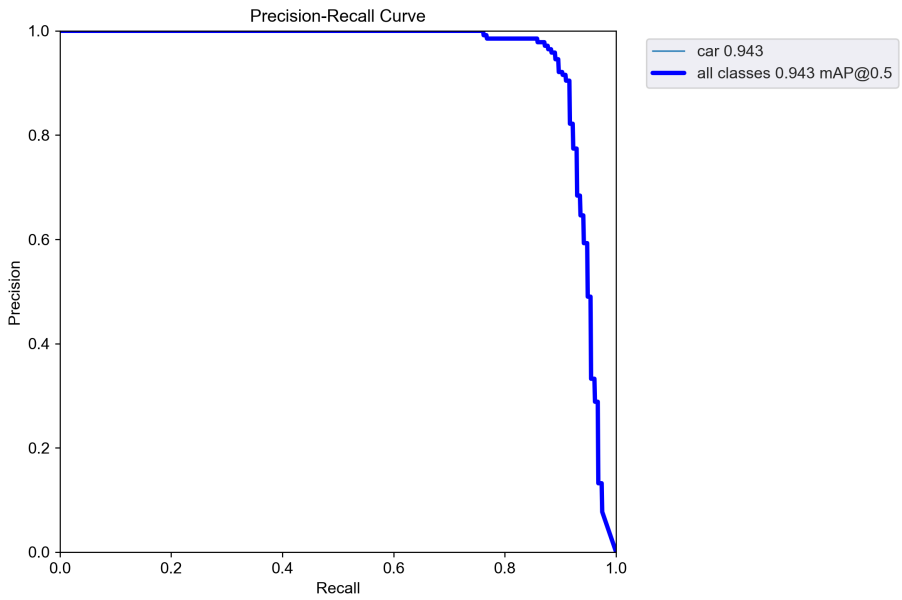
```
!python yolov5/val.py --weights car_detection_yolov5/result/trained/adamw_1/weights/best.pt --data car_detection_yolov5/data/data.yaml --
```

```
'cp950' codec can't decode byte 0xf0 in position 9: illegal multibyte sequence

val: data=car_detection_yolov5/data/data.yaml, weights=['car_detection_yolov5/result/trained/adamw_1/weights/best.pt'], bat
ch_size=32, imgsz=676, conf_thres=0.001, iou_thres=0.6, max_det=300, task=test, device=, workers=8, single_cls=False, augme
nt=False, verbose=False, save_txt=False, save_hybrid=False, save_conf=False, save_json=False, project=car_detection_yolov5/
result/valid, name=exp, exist_ok=False, half=False, dnn=False
fatal: cannot change to 'C:\Users\User\Desktop\UF-MAE\2022': No such file or directory
YOLOv5  2022-12-8 Python-3.8.15 torch-1.13.0+cpu CPU

Fusing layers...
Model summary: 212 layers, 20852934 parameters, 0 gradients
WARNING  --img-size 676 must be multiple of max stride 32, updating to 704

test: Scanning C:\Users\User\Desktop\UF-MAE\2022 Fall\Apply Machine Learning\project-3-Danielyaoan\car_detection_yolov5\tes
t\labels.cache... 175 images, 83 backgrounds, 0 corrupt: 100%|██████████| 175/175 [00:00<?, ?it/s]
test: Scanning C:\Users\User\Desktop\UF-MAE\2022 Fall\Apply Machine Learning\project-3-Danielyaoan\car_detection_yolov5\tes
t\labels.cache... 175 images, 83 backgrounds, 0 corrupt: 100%|██████████| 175/175 [00:00<?, ?it/s]

                 Class     Images  Instances          P          R      mAP50   mAP50-95:   0%|          | 0/6 [00:00<?, ?i
t/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95:  17%|██         | 1/6 [00:09<00:4
5,  9.08s/it]
                 Class     Images  Instances          P          R      mAP50   mAP50-95:  33%|███        | 2/6 [00:15<00:2
9,  7.48s/it]
                 Class     Images  Instances          P          R      mAP50   mAP50-95:  50%|█████      | 3/6 [00:21<00:2
0,  6.76s/it]
                 Class     Images  Instances          P          R      mAP50   mAP50-95:  67%|██████▋    | 4/6 [00:27<00:1
2,  6.41s/it]
                 Class     Images  Instances          P          R      mAP50   mAP50-95:  83%|████████▎  | 5/6 [00:33<00:0
6,  6.19s/it]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 6/6 [00:35<00:0
0,  5.01s/it]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 6/6 [00:35<00:0
0,  5.96s/it]
                   all        175        155      0.961      0.884      0.943       0.54
Speed: 1.5ms pre-process, 199.7ms inference, 0.5ms NMS per image at shape (32, 3, 704, 704)
Results saved to car_detection_yolov5\result\valid\exp2
```

In [93]:

```
Image('car_detection_yolov5/result/valid/exp2/PR_curve.png',width=700)
```

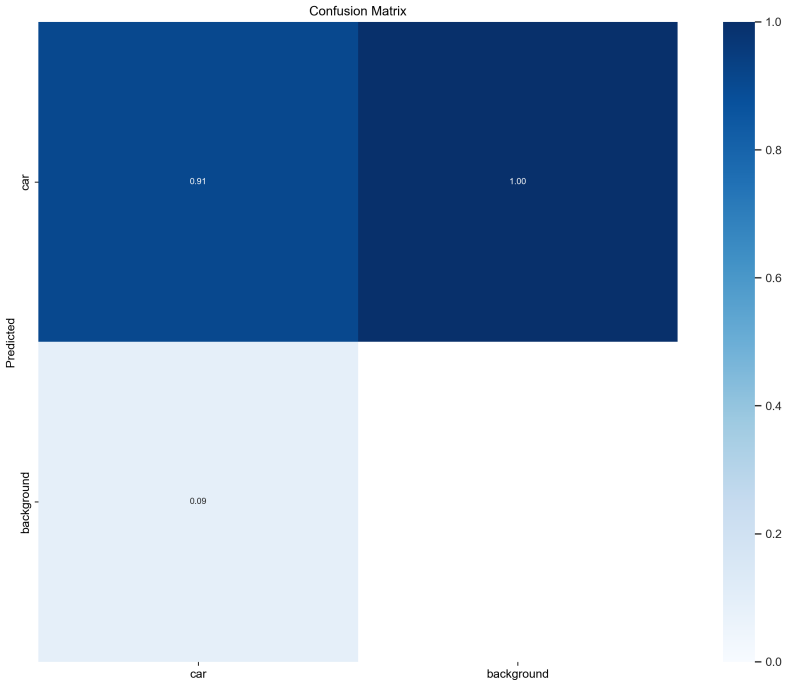
Out[93]:



In [95]:

```
Image('car_detection_yolov5/result/valid/exp2/confusion_matrix.png',width=700)
```

Out[95]:



Although the performance is a little lower than training data, we can still conlcude that when IoU with threshold = 0.5, performance is good and mAP can be over 0.94.

In [ ]:

```

```