CAR NUMBER PLATE DETECTION





1 DATA

This dataset contains 433 images with bounding box annotations of the car license plates within the image. Annotations are provided in the PASCAL VOC format.
link:

2 METHOD

The flowing is the workflow:

Import the data and verify the data with annotation in xml format

We need to down size the image pixels

Then pass the reduced size images into models

3. Data Loding

We first check the path of each image and annotations from dataset

we take every and each picture and convert it into an array the usage of OpenCV and resize the picture into 200 x 200 that's the usual well suited length of the pre-educated switch getting to know model.

Thenceforth, we will normalize the photo simply through dividing with most quantity as we realize that the most quantity for an 8-bit photo is

255. That the cause we are able to divide our photo 255.0. Then convert the image in to vector array

Then split the data into training and testing set using sklearn.

4. Deep Learning

1. VGG19

VGG-19 is a convolutional neural network that is 19 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database [1]. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224. For more pretrained networks in MATLAB®, see Pretrained Deep Neural Networks.

Architecture used

```
model = Sequential()
```

```
model.add(VGG19(weights="imagenet", include_top=False, input_shape=(200,200, 3)))
model.add(Flatten())
model.add(Dropout(0.4))
model.add(Dense(256, activation="relu"))
model.add(Dense(128, activation="relu"))
model.add(Dense(64, activation="relu"))
model.add(Dense(4, activation="sigmoid"))
model.layers[-7].trainable = False

model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg19 (Functional)          (None, 6, 6, 512)         20024384

 flatten_2 (Flatten)         (None, 18432)             0

 dropout_2 (Dropout)         (None, 18432)             0

 dense_8 (Dense)             (None, 256)               4718848

 dense_9 (Dense)             (None, 128)               32896

 dense_10 (Dense)            (None, 64)                8256

 dense_11 (Dense)            (None, 4)                 260

=================================================================
Total params: 24,784,644
Trainable params: 4,760,260
```

Non-trainable params: 20,024,384

_____

2 RESNET50

ResNet-50 is a convolutional neural network that is 50 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database [1]. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224. For more pretrained networks in MATLAB®, see Pretrained Deep Neural Networks.

You can use classify to classify new images using the ResNet-50 model. Follow the steps of Classify Image Using GoogLeNet and replace GoogLeNet with ResNet-50

Architecture used

```
model2 = Sequential()
model2.add(ResNet50(weights='imagenet',include_top=False, input_shape=(200,200, 3)))
model2.add(Flatten())
model2.add(Dropout(0.4))
model2.add(Dense(500, activation="relu"))
model2.add(Dense(250, activation="relu"))
```

```
model2.add(Dense(150, activation="relu"))
model2.add(Dense(4, activation="sigmoid"))
model2.layers[-7].trainable = False
model2.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 7, 7, 2048)        23587712

 flatten_3 (Flatten)         (None, 100352)            0

 dropout_3 (Dropout)         (None, 100352)            0

 dense_12 (Dense)            (None, 500)               50176500

 dense_13 (Dense)            (None, 250)               125250

 dense_14 (Dense)            (None, 150)               37650

 dense_15 (Dense)            (None, 4)                 604

=================================================================
Total params: 73,927,716
Trainable params: 50,340,004
Non-trainable params: 23,587,712
_________________________________________________________________
```

5. Predictions

The result came out as

1 VGG19 model is 0.7715

2 RESNET50 model is 0.6126

6. Future Improvements

If we add some more layer and try more hyper tuning the result could improve