In [91]:

```
import os
%matplotlib inline
# Prevent CUDA from using GPU as it does not work well on my pc
os.environ["CUDA_VISIBLE_DEVICES"] = "-1"

# Set Constants of the model
BATCH_SIZE = 64
SHUFFLE_BUFFER_SIZE = 100
```

In [2]:

```
# Helper functions
import numpy as np

# Breaks down a list of integer values into a one-hot like format
def one_hot_training(np_array):
    transformed_list = []
    for arr in np_array:
        new_arr = np.zeros(10)
        new_arr[int(arr)] = 1
        transformed_list.append(new_arr)
    return np.array(transformed_list)

# This translates the highest value from the one-hot encoding into the correct sign nam
def one_hot_translator(np_array):
    labels_names = ['Stop','Yield','Red Light','Green Light','Roundabout','Right Turn O
                    'Do Not Enter','Crosswalk','Handicap Parking','No Parking']
    return labels_names[np.argmax(np_array)]

# This translates an entire array of one-hot encoded sign predictions
def translate_all(np_array):
    translated_values = []
    for i in np_array:
        translated_values.append(one_hot_translator(i))
    return np.array(translated_values)
```

In [3]:

```
# First import the data
import tensorflow as tf
data_train = np.load('data_train.npy').transpose()
labels_train = np.load('labels_train.npy')
data_train = np.array([i.reshape(300,300,3) for i in data_train])
data_train = np.array(tf.cast(tf.image.resize(data_train,(150,150)), np.uint8))
```

In [4]:

```
# Process the data so that it is in the expected form for the InceptionV3 model
import tensorflow as tf
processed = tf.keras.applications.inception_v3.preprocess_input(data_train, data_format

# Break down data into training and test sets
from sklearn.model_selection import train_test_split
x_train, x_test, t_train, t_test = train_test_split(processed, one_hot_training(labels_
```

In [5]:

```
# Augment data to reduce overfitting

from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(horizontal_flip=True,
```

```
                                   vertical_flip=True,
                                   rotation_range=90,
                                   brightness_range=(.75, 1))

train_generator = train_datagen.flow(
    x_train,
    y = t_train,
    batch_size=BATCH_SIZE)
```

In [6]:

```
# Import the InceptionV3 Model

from tensorflow.keras.applications.inception_v3 import InceptionV3
inception = InceptionV3(input_shape=(150,150,3),
                       include_top=False,
                       weights='imagenet')

# Set layers to false to prevent overwriting the existing model
for layer in inception.layers:
    layer.trainable = False

# Create output layers that will be trained
from tensorflow.keras.optimizers import SGD
x = tf.keras.layers.Flatten()(inception.output)
x = tf.keras.layers.Dense(1024, activation="relu")(x)
x = tf.keras.layers.Dropout(0.15)(x)
x = tf.keras.layers.Dense(10, activation='softmax')(x)

# Create Optimizer
Adam = tf.keras.optimizers.Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.999, epsilon=
Nadam = tf.keras.optimizers.Nadam(learning_rate=0.001, beta_1=0.9, beta_2=0.999, epsilo
SGD = SGD(learning_rate=0.01, nesterov=True)
optimizer = SGD

# Finalize and compile the model
model = tf.keras.Model(inception.input, outputs = x)
model.compile(optimizer = optimizer,
             loss = 'categorical_crossentropy',
             metrics = ['categorical_accuracy', 'acc','mean_squared_error'])
```

In [7]:

```
# Fit the model to the dataset
es = tf.keras.callbacks.EarlyStopping(monitor='acc', mode='max', verbose=1, patience=10
callbacks = tf.keras.callbacks.Callback()
history = model.fit(train_generator, epochs=140, batch_size=BATCH_SIZE, callbacks=[es])
model.save("140_epoch.h5")
```

```
Epoch 1/140
78/78 [==============================] - 40s 475ms/step - loss: 2.1024 - categorical_acc
uracy: 0.6959 - acc: 0.6959 - mean_squared_error: 0.0451
Epoch 2/140
78/78 [==============================] - 38s 487ms/step - loss: 0.4627 - categorical_acc
uracy: 0.8565 - acc: 0.8565 - mean_squared_error: 0.0206
Epoch 3/140
78/78 [==============================] - 37s 472ms/step - loss: 0.3757 - categorical_acc
uracy: 0.8931 - acc: 0.8931 - mean_squared_error: 0.0164
Epoch 4/140
78/78 [==============================] - 36s 464ms/step - loss: 0.3250 - categorical_acc
uracy: 0.9036 - acc: 0.9036 - mean_squared_error: 0.0144
```

```
Epoch 5/140
78/78 [==============================] - 36s 461ms/step - loss: 0.3092 - categorical_acc
uracy: 0.9118 - acc: 0.9118 - mean_squared_error: 0.0132
Epoch 6/140
78/78 [==============================] - 36s 462ms/step - loss: 0.2946 - categorical_acc
uracy: 0.9151 - acc: 0.9151 - mean_squared_error: 0.0127
Epoch 7/140
78/78 [==============================] - 37s 467ms/step - loss: 0.2733 - categorical_acc
uracy: 0.9221 - acc: 0.9221 - mean_squared_error: 0.0118
Epoch 8/140
78/78 [==============================] - 37s 468ms/step - loss: 0.2688 - categorical_acc
uracy: 0.9249 - acc: 0.9249 - mean_squared_error: 0.0116
Epoch 9/140
78/78 [==============================] - 37s 466ms/step - loss: 0.2391 - categorical_acc
uracy: 0.9326 - acc: 0.9326 - mean_squared_error: 0.0104
Epoch 10/140
78/78 [==============================] - 37s 465ms/step - loss: 0.2465 - categorical_acc
uracy: 0.9334 - acc: 0.9334 - mean_squared_error: 0.0104
Epoch 11/140
78/78 [==============================] - 36s 464ms/step - loss: 0.2149 - categorical_acc
uracy: 0.9383 - acc: 0.9383 - mean_squared_error: 0.0095
Epoch 12/140
78/78 [==============================] - 37s 469ms/step - loss: 0.2099 - categorical_acc
uracy: 0.9411 - acc: 0.9411 - mean_squared_error: 0.0092
Epoch 13/140
78/78 [==============================] - 37s 467ms/step - loss: 0.2060 - categorical_acc
uracy: 0.9407 - acc: 0.9407 - mean_squared_error: 0.0090
Epoch 14/140
78/78 [==============================] - 37s 465ms/step - loss: 0.2102 - categorical_acc
uracy: 0.9403 - acc: 0.9403 - mean_squared_error: 0.0092
Epoch 15/140
78/78 [==============================] - 37s 465ms/step - loss: 0.1916 - categorical_acc
uracy: 0.9459 - acc: 0.9459 - mean_squared_error: 0.0085
Epoch 16/140
78/78 [==============================] - 37s 477ms/step - loss: 0.1919 - categorical_acc
uracy: 0.9465 - acc: 0.9465 - mean_squared_error: 0.0084
Epoch 17/140
78/78 [==============================] - 37s 467ms/step - loss: 0.1870 - categorical_acc
uracy: 0.9473 - acc: 0.9473 - mean_squared_error: 0.0082
Epoch 18/140
78/78 [==============================] - 37s 467ms/step - loss: 0.1913 - categorical_acc
uracy: 0.9447 - acc: 0.9447 - mean_squared_error: 0.0085
Epoch 19/140
78/78 [==============================] - 36s 465ms/step - loss: 0.1691 - categorical_acc
uracy: 0.9508 - acc: 0.9508 - mean_squared_error: 0.0076
Epoch 20/140
78/78 [==============================] - 37s 465ms/step - loss: 0.1754 - categorical_acc
uracy: 0.9483 - acc: 0.9483 - mean_squared_error: 0.0080
Epoch 21/140
78/78 [==============================] - 37s 465ms/step - loss: 0.1823 - categorical_acc
uracy: 0.9467 - acc: 0.9467 - mean_squared_error: 0.0082
Epoch 22/140
78/78 [==============================] - 37s 468ms/step - loss: 0.1691 - categorical_acc
uracy: 0.9548 - acc: 0.9548 - mean_squared_error: 0.0072
Epoch 23/140
78/78 [==============================] - 37s 465ms/step - loss: 0.1492 - categorical_acc
uracy: 0.9580 - acc: 0.9580 - mean_squared_error: 0.0067
Epoch 24/140
78/78 [==============================] - 37s 466ms/step - loss: 0.1664 - categorical_acc
uracy: 0.9518 - acc: 0.9518 - mean_squared_error: 0.0074
```

```
Epoch 25/140
78/78 [==============================] - 37s 468ms/step - loss: 0.1494 - categorical_acc
uracy: 0.9586 - acc: 0.9586 - mean_squared_error: 0.0066
Epoch 26/140
78/78 [==============================] - 37s 469ms/step - loss: 0.1433 - categorical_acc
uracy: 0.9560 - acc: 0.9560 - mean_squared_error: 0.0067
Epoch 27/140
78/78 [==============================] - 37s 466ms/step - loss: 0.1447 - categorical_acc
uracy: 0.9598 - acc: 0.9598 - mean_squared_error: 0.0065
Epoch 28/140
78/78 [==============================] - 37s 468ms/step - loss: 0.1345 - categorical_acc
uracy: 0.9607 - acc: 0.9607 - mean_squared_error: 0.0062
Epoch 29/140
78/78 [==============================] - 37s 468ms/step - loss: 0.1432 - categorical_acc
uracy: 0.9615 - acc: 0.9615 - mean_squared_error: 0.0064
Epoch 30/140
78/78 [==============================] - 37s 466ms/step - loss: 0.1304 - categorical_acc
uracy: 0.9617 - acc: 0.9617 - mean_squared_error: 0.0060
Epoch 31/140
78/78 [==============================] - 37s 468ms/step - loss: 0.1269 - categorical_acc
uracy: 0.9637 - acc: 0.9637 - mean_squared_error: 0.0058
Epoch 32/140
78/78 [==============================] - 37s 467ms/step - loss: 0.1252 - categorical_acc
uracy: 0.9629 - acc: 0.9629 - mean_squared_error: 0.0058
Epoch 33/140
78/78 [==============================] - 37s 469ms/step - loss: 0.1243 - categorical_acc
uracy: 0.9637 - acc: 0.9637 - mean_squared_error: 0.0059
Epoch 34/140
78/78 [==============================] - 37s 469ms/step - loss: 0.1307 - categorical_acc
uracy: 0.9592 - acc: 0.9592 - mean_squared_error: 0.0060
Epoch 35/140
78/78 [==============================] - 37s 466ms/step - loss: 0.1208 - categorical_acc
uracy: 0.9657 - acc: 0.9657 - mean_squared_error: 0.0055
Epoch 36/140
78/78 [==============================] - 37s 469ms/step - loss: 0.1236 - categorical_acc
uracy: 0.9647 - acc: 0.9647 - mean_squared_error: 0.0058
Epoch 37/140
78/78 [==============================] - 37s 467ms/step - loss: 0.1254 - categorical_acc
uracy: 0.9631 - acc: 0.9631 - mean_squared_error: 0.0058
Epoch 38/140
78/78 [==============================] - 37s 468ms/step - loss: 0.1191 - categorical_acc
uracy: 0.9659 - acc: 0.9659 - mean_squared_error: 0.0054
Epoch 39/140
78/78 [==============================] - 37s 468ms/step - loss: 0.1196 - categorical_acc
uracy: 0.9641 - acc: 0.9641 - mean_squared_error: 0.0057
Epoch 40/140
78/78 [==============================] - 37s 470ms/step - loss: 0.1072 - categorical_acc
uracy: 0.9675 - acc: 0.9675 - mean_squared_error: 0.0050
Epoch 41/140
78/78 [==============================] - 37s 469ms/step - loss: 0.1116 - categorical_acc
uracy: 0.9695 - acc: 0.9695 - mean_squared_error: 0.0049
Epoch 42/140
78/78 [==============================] - 37s 467ms/step - loss: 0.1214 - categorical_acc
uracy: 0.9637 - acc: 0.9637 - mean_squared_error: 0.0057
Epoch 43/140
78/78 [==============================] - 37s 469ms/step - loss: 0.1035 - categorical_acc
uracy: 0.9643 - acc: 0.9643 - mean_squared_error: 0.0050
Epoch 44/140
78/78 [==============================] - 37s 467ms/step - loss: 0.1029 - categorical_acc
uracy: 0.9673 - acc: 0.9673 - mean_squared_error: 0.0049
```

```
Epoch 45/140
78/78 [==============================] - 37s 469ms/step - loss: 0.1046 - categorical_acc
uracy: 0.9679 - acc: 0.9679 - mean_squared_error: 0.0049
Epoch 46/140
78/78 [==============================] - 37s 469ms/step - loss: 0.0994 - categorical_acc
uracy: 0.9681 - acc: 0.9681 - mean_squared_error: 0.0048
Epoch 47/140
78/78 [==============================] - 37s 467ms/step - loss: 0.1030 - categorical_acc
uracy: 0.9683 - acc: 0.9683 - mean_squared_error: 0.0048
Epoch 48/140
78/78 [==============================] - 37s 466ms/step - loss: 0.1017 - categorical_acc
uracy: 0.9707 - acc: 0.9707 - mean_squared_error: 0.0048
Epoch 49/140
78/78 [==============================] - 37s 468ms/step - loss: 0.0911 - categorical_acc
uracy: 0.9736 - acc: 0.9736 - mean_squared_error: 0.0042
Epoch 50/140
78/78 [==============================] - 37s 466ms/step - loss: 0.0958 - categorical_acc
uracy: 0.9709 - acc: 0.9709 - mean_squared_error: 0.0046
Epoch 51/140
78/78 [==============================] - 37s 466ms/step - loss: 0.0944 - categorical_acc
uracy: 0.9740 - acc: 0.9740 - mean_squared_error: 0.0044
Epoch 52/140
78/78 [==============================] - 37s 465ms/step - loss: 0.0918 - categorical_acc
uracy: 0.9736 - acc: 0.9736 - mean_squared_error: 0.0044
Epoch 53/140
78/78 [==============================] - 37s 467ms/step - loss: 0.0855 - categorical_acc
uracy: 0.9742 - acc: 0.9742 - mean_squared_error: 0.0041
Epoch 54/140
78/78 [==============================] - 37s 467ms/step - loss: 0.0935 - categorical_acc
uracy: 0.9734 - acc: 0.9734 - mean_squared_error: 0.0044
Epoch 55/140
78/78 [==============================] - 37s 473ms/step - loss: 0.0882 - categorical_acc
uracy: 0.9736 - acc: 0.9736 - mean_squared_error: 0.0042
Epoch 56/140
78/78 [==============================] - 37s 466ms/step - loss: 0.0894 - categorical_acc
uracy: 0.9738 - acc: 0.9738 - mean_squared_error: 0.0041
Epoch 57/140
78/78 [==============================] - 37s 469ms/step - loss: 0.0786 - categorical_acc
uracy: 0.9770 - acc: 0.9770 - mean_squared_error: 0.0038
Epoch 58/140
78/78 [==============================] - 37s 469ms/step - loss: 0.0829 - categorical_acc
uracy: 0.9746 - acc: 0.9746 - mean_squared_error: 0.0039
Epoch 59/140
78/78 [==============================] - 37s 470ms/step - loss: 0.0823 - categorical_acc
uracy: 0.9764 - acc: 0.9764 - mean_squared_error: 0.0039
Epoch 60/140
78/78 [==============================] - 37s 472ms/step - loss: 0.0873 - categorical_acc
uracy: 0.9762 - acc: 0.9762 - mean_squared_error: 0.0040
Epoch 61/140
78/78 [==============================] - 38s 487ms/step - loss: 0.0788 - categorical_acc
uracy: 0.9770 - acc: 0.9770 - mean_squared_error: 0.0037
Epoch 62/140
78/78 [==============================] - 39s 500ms/step - loss: 0.0819 - categorical_acc
uracy: 0.9748 - acc: 0.9748 - mean_squared_error: 0.0038
Epoch 63/140
78/78 [==============================] - 39s 499ms/step - loss: 0.0767 - categorical_acc
uracy: 0.9784 - acc: 0.9784 - mean_squared_error: 0.0034
Epoch 64/140
78/78 [==============================] - 39s 501ms/step - loss: 0.0722 - categorical_acc
uracy: 0.9784 - acc: 0.9784 - mean_squared_error: 0.0034
```

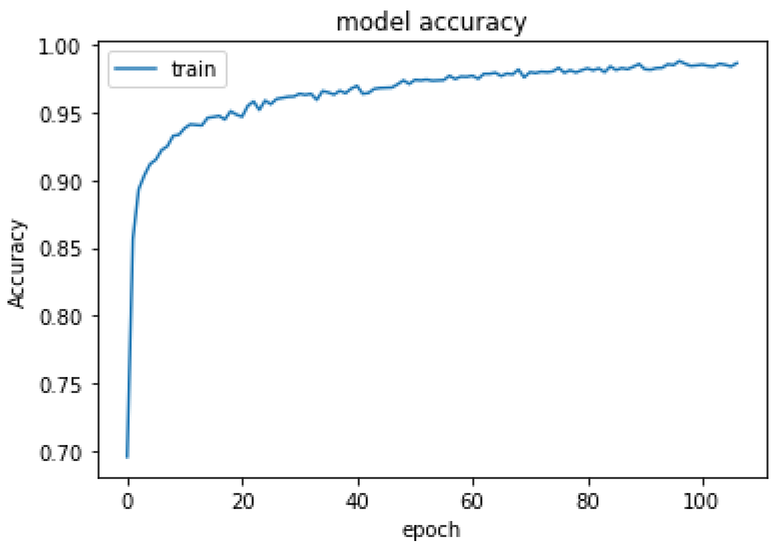
```
Epoch 65/140
78/78 [==============================] - 38s 488ms/step - loss: 0.0701 - categorical_acc
uracy: 0.9792 - acc: 0.9792 - mean_squared_error: 0.0034
Epoch 66/140
78/78 [==============================] - 37s 475ms/step - loss: 0.0741 - categorical_acc
uracy: 0.9768 - acc: 0.9768 - mean_squared_error: 0.0036
Epoch 67/140
78/78 [==============================] - 39s 504ms/step - loss: 0.0709 - categorical_acc
uracy: 0.9786 - acc: 0.9786 - mean_squared_error: 0.0033
Epoch 68/140
78/78 [==============================] - 38s 489ms/step - loss: 0.0673 - categorical_acc
uracy: 0.9780 - acc: 0.9780 - mean_squared_error: 0.0033
Epoch 69/140
78/78 [==============================] - 38s 489ms/step - loss: 0.0629 - categorical_acc
uracy: 0.9816 - acc: 0.9816 - mean_squared_error: 0.0030
Epoch 70/140
78/78 [==============================] - 38s 487ms/step - loss: 0.0719 - categorical_acc
uracy: 0.9758 - acc: 0.9758 - mean_squared_error: 0.0036
Epoch 71/140
78/78 [==============================] - 39s 492ms/step - loss: 0.0665 - categorical_acc
uracy: 0.9796 - acc: 0.9796 - mean_squared_error: 0.0032
Epoch 72/140
78/78 [==============================] - 38s 487ms/step - loss: 0.0619 - categorical_acc
uracy: 0.9790 - acc: 0.9790 - mean_squared_error: 0.0031
Epoch 73/140
78/78 [==============================] - 41s 520ms/step - loss: 0.0649 - categorical_acc
uracy: 0.9800 - acc: 0.9800 - mean_squared_error: 0.0031
Epoch 74/140
78/78 [==============================] - 40s 513ms/step - loss: 0.0673 - categorical_acc
uracy: 0.9796 - acc: 0.9796 - mean_squared_error: 0.0033
Epoch 75/140
78/78 [==============================] - 39s 502ms/step - loss: 0.0668 - categorical_acc
uracy: 0.9802 - acc: 0.9802 - mean_squared_error: 0.0032
Epoch 76/140
78/78 [==============================] - 38s 484ms/step - loss: 0.0621 - categorical_acc
uracy: 0.9828 - acc: 0.9828 - mean_squared_error: 0.0029
Epoch 77/140
78/78 [==============================] - 38s 484ms/step - loss: 0.0673 - categorical_acc
uracy: 0.9792 - acc: 0.9792 - mean_squared_error: 0.0033
Epoch 78/140
78/78 [==============================] - 37s 470ms/step - loss: 0.0634 - categorical_acc
uracy: 0.9810 - acc: 0.9810 - mean_squared_error: 0.0030
Epoch 79/140
78/78 [==============================] - 39s 497ms/step - loss: 0.0686 - categorical_acc
uracy: 0.9794 - acc: 0.9794 - mean_squared_error: 0.0033
Epoch 80/140
78/78 [==============================] - 39s 492ms/step - loss: 0.0659 - categorical_acc
uracy: 0.9810 - acc: 0.9810 - mean_squared_error: 0.0031
Epoch 81/140
78/78 [==============================] - 37s 474ms/step - loss: 0.0578 - categorical_acc
uracy: 0.9824 - acc: 0.9824 - mean_squared_error: 0.0029
Epoch 82/140
78/78 [==============================] - 38s 480ms/step - loss: 0.0614 - categorical_acc
uracy: 0.9810 - acc: 0.9810 - mean_squared_error: 0.0030
Epoch 83/140
78/78 [==============================] - 38s 483ms/step - loss: 0.0605 - categorical_acc
uracy: 0.9824 - acc: 0.9824 - mean_squared_error: 0.0029
Epoch 84/140
78/78 [==============================] - 39s 496ms/step - loss: 0.0668 - categorical_acc
uracy: 0.9796 - acc: 0.9796 - mean_squared_error: 0.0033
```

```
Epoch 85/140
78/78 [==============================] - 38s 484ms/step - loss: 0.0521 - categorical_acc
uracy: 0.9839 - acc: 0.9839 - mean_squared_error: 0.0026
Epoch 86/140
78/78 [==============================] - 39s 496ms/step - loss: 0.0608 - categorical_acc
uracy: 0.9812 - acc: 0.9812 - mean_squared_error: 0.0029
Epoch 87/140
78/78 [==============================] - 39s 491ms/step - loss: 0.0574 - categorical_acc
uracy: 0.9826 - acc: 0.9826 - mean_squared_error: 0.0029
Epoch 88/140
78/78 [==============================] - 39s 493ms/step - loss: 0.0592 - categorical_acc
uracy: 0.9818 - acc: 0.9818 - mean_squared_error: 0.0028
Epoch 89/140
78/78 [==============================] - 38s 486ms/step - loss: 0.0568 - categorical_acc
uracy: 0.9837 - acc: 0.9837 - mean_squared_error: 0.0027
Epoch 90/140
78/78 [==============================] - 38s 482ms/step - loss: 0.0510 - categorical_acc
uracy: 0.9857 - acc: 0.9857 - mean_squared_error: 0.0024
Epoch 91/140
78/78 [==============================] - 37s 476ms/step - loss: 0.0559 - categorical_acc
uracy: 0.9818 - acc: 0.9818 - mean_squared_error: 0.0028
Epoch 92/140
78/78 [==============================] - 37s 471ms/step - loss: 0.0618 - categorical_acc
uracy: 0.9814 - acc: 0.9814 - mean_squared_error: 0.0030
Epoch 93/140
78/78 [==============================] - 37s 468ms/step - loss: 0.0548 - categorical_acc
uracy: 0.9826 - acc: 0.9826 - mean_squared_error: 0.0027
Epoch 94/140
78/78 [==============================] - 36s 465ms/step - loss: 0.0555 - categorical_acc
uracy: 0.9828 - acc: 0.9828 - mean_squared_error: 0.0026
Epoch 95/140
78/78 [==============================] - 37s 470ms/step - loss: 0.0500 - categorical_acc
uracy: 0.9855 - acc: 0.9855 - mean_squared_error: 0.0025
Epoch 96/140
78/78 [==============================] - 37s 467ms/step - loss: 0.0479 - categorical_acc
uracy: 0.9849 - acc: 0.9849 - mean_squared_error: 0.0023
Epoch 97/140
78/78 [==============================] - 37s 466ms/step - loss: 0.0453 - categorical_acc
uracy: 0.9879 - acc: 0.9879 - mean_squared_error: 0.0020
Epoch 98/140
78/78 [==============================] - 37s 469ms/step - loss: 0.0482 - categorical_acc
uracy: 0.9859 - acc: 0.9859 - mean_squared_error: 0.0024
Epoch 99/140
78/78 [==============================] - 37s 466ms/step - loss: 0.0480 - categorical_acc
uracy: 0.9843 - acc: 0.9843 - mean_squared_error: 0.0024
Epoch 100/140
78/78 [==============================] - 37s 469ms/step - loss: 0.0495 - categorical_acc
uracy: 0.9847 - acc: 0.9847 - mean_squared_error: 0.0024
Epoch 101/140
78/78 [==============================] - 37s 466ms/step - loss: 0.0460 - categorical_acc
uracy: 0.9853 - acc: 0.9853 - mean_squared_error: 0.0023
Epoch 102/140
78/78 [==============================] - 36s 467ms/step - loss: 0.0503 - categorical_acc
uracy: 0.9843 - acc: 0.9843 - mean_squared_error: 0.0025
Epoch 103/140
78/78 [==============================] - 37s 467ms/step - loss: 0.0526 - categorical_acc
uracy: 0.9839 - acc: 0.9839 - mean_squared_error: 0.0026
Epoch 104/140
78/78 [==============================] - 36s 463ms/step - loss: 0.0462 - categorical_acc
uracy: 0.9859 - acc: 0.9859 - mean_squared_error: 0.0022
```

```
Epoch 105/140
78/78 [==============================] - 37s 465ms/step - loss: 0.0480 - categorical_acc
uracy: 0.9851 - acc: 0.9851 - mean_squared_error: 0.0023
Epoch 106/140
78/78 [==============================] - 37s 472ms/step - loss: 0.0495 - categorical_acc
uracy: 0.9839 - acc: 0.9839 - mean_squared_error: 0.0024
Epoch 107/140
78/78 [==============================] - 37s 467ms/step - loss: 0.0479 - categorical_acc
uracy: 0.9863 - acc: 0.9863 - mean_squared_error: 0.0024
Epoch 107: early stopping
```

In [17]:

```python
# Plot the progression of the acccuracy through the epochs
import matplotlib.pyplot as plt
plt.plot(history.history['acc'])
plt.title('model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



In [179…

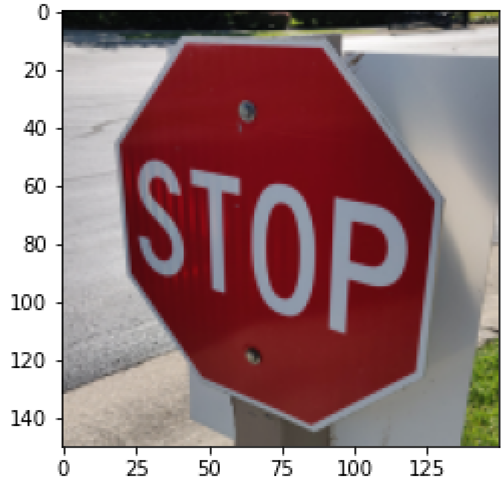
```python
# Demo to randomly pick a sign and demonstrate that it is predicted correctly

from random import randint
test_image = randint(0,len(data_train))
test = np.expand_dims(data_train[test_image], axis=0)
test = tf.keras.applications.inception_v3.preprocess_input(
    test, data_format=None
)
print(one_hot_translator(model.predict(test)))
plt.imshow(data_train[test_image])
```

```
1/1 [==============================] - 0s 37ms/step
Stop
```

Out[179…

```
<matplotlib.image.AxesImage at 0x248214161c0>
```



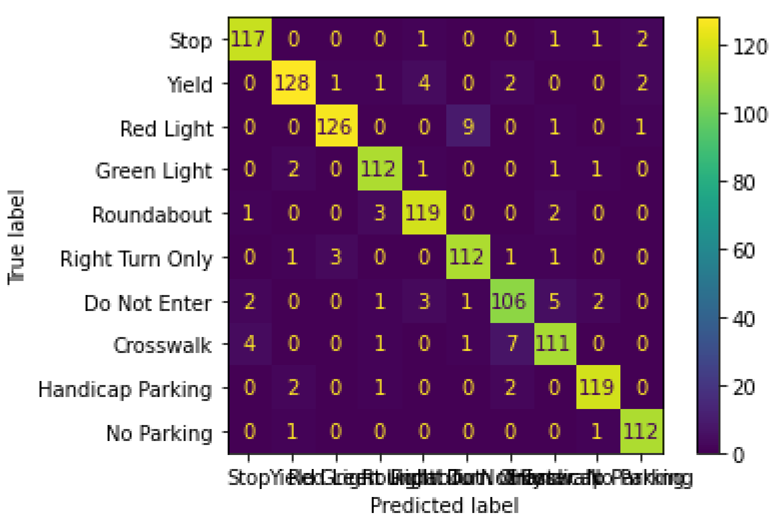
In [19]:
```
# Calculate the predictions for the test values
predictions = model.predict(x_test)
```

```
39/39 [==============================] - 9s 231ms/step
```

In [20]:
```
evaluation = model.evaluate(x_test, t_test)
print("Test run accuracy is {}".format(evaluation[-1]))
```

```
39/39 [==============================] - 9s 228ms/step - loss: 0.3067 - categorical_accu
racy: 0.9379 - acc: 0.9379 - mean_squared_error: 0.0100
Test run accuracy is 0.010019068606197834
```

In [21]:
```
# Create a Confusion Matrix to show the weakness in the model
predicted_values = translate_all(predictions)
real_values = translate_all(t_test)
from sklearn.metrics import confusion_matrix
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cfm = confusion_matrix(real_values, predicted_values)
disp = ConfusionMatrixDisplay(confusion_matrix=cfm, display_labels=['Stop','Yield','Red
disp.plot()
plt.show()
```



In [22]:
```
# Test Run Data
```

```
# Testing optimizers
# NADAM=                                          times = [275,272,286], loss after 3 runs = [
# ADAM=                                           times = [190,186,190], loss after 3 runs = [
# SGD(.001, Nesterov=No, Momentum = No),    times = [182,180,181], loss after 3 runs = [
# SGD(.01, Nesterov=No, Momentum = No),     times = [180,180,179], loss after 3 runs = [
# SGD(.01, Nesterov=Yes, Momentum = 0.25), times = [183,183,183], loss after 3 runs = [
# SGD(.01, Nesterov=Yes, Momentum = 0.5),  times = [189,183,183], loss after 3 runs = [
# SGD(.01, Nesterov=Yes, Momentum = 0.75), times = [185,185,184], loss after 3 runs = [
```

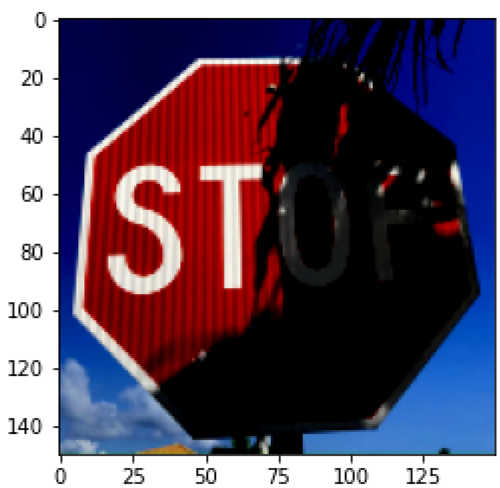
In [97]:

```
# This method is used to show an example of the post processed test data
from random import randint
import matplotlib.pyplot as plt
test_image = randint(0,len(x_train))
plt.imshow(x_train[test_image])
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or
[0..255] for integers).
```

Out[97]:

```
<matplotlib.image.AxesImage at 0x24821a2caf0>
```



In [16]:

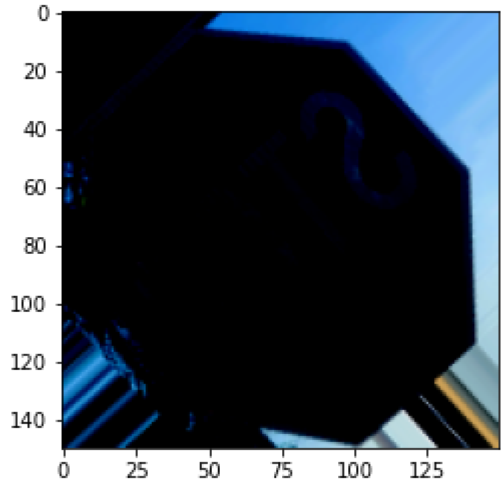
```
# Original 140 Epoch run was done with .25-.75 brightness range
```

In [37]:

```
# This method is used to show an example of the post processed data
from random import randint
x,y = next(train_generator)

plt.imshow(x[randint(0,63)])
plt.show()
```

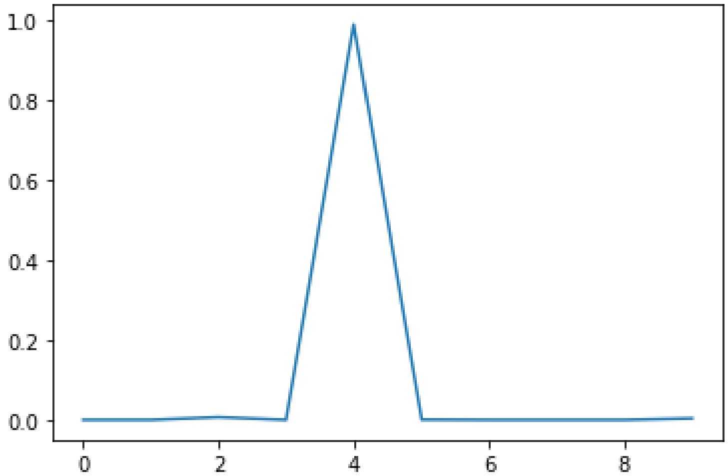
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or
[0..255] for integers).
```



In [253…

```
Load_image = tf.keras.preprocessing.image.load_img('New.jpg')
og_image = Load_image.copy()
Load_image = np.array(tf.cast(tf.image.resize(Load_image,(150,150)), np.uint8))
print(Load_image.shape)
Load_image = tf.keras.applications.inception_v3.preprocess_input(
    Load_image, data_format=None
)
Load_image = tf.expand_dims(Load_image,0)
prediction = model.predict(Load_image)
print(prediction)
print(one_hot_translator(prediction))
plt.imshow(og_image)
calculate_likelyhood(prediction)
plt.plot([0,1,2,3,4,5,6,7,8,9],prediction[0])
plt.show()
```

```
(150, 150, 3)
1/1 [==============================] - 0s 38ms/step
[[2.3534343e-05 1.6287031e-05 6.8655545e-03 3.1475363e-06 9.8857528e-01
  6.3804933e-04 1.9053403e-05 1.5509835e-05 2.7066142e-07 3.8431643e-03]]
Roundabout
```



In [250…

```
def calculate_std(np_array):
    translated_values = []
    for i in np_array:
        translated_values.append((np.argmax(i))/sum(i))
    return np.array(translated_values)
```

In [251…

```
x = calculate_std(predictions)
print(x)
print(np.argmax(x))
```

```
[2.00000003 7.00000062 6.99999951 ... 2.00000004 5.99999922 6.99999949]
422
```

In [ ]: