In [1]:
```
import tensorflow
import keras
from keras.models import Sequential
from keras.layers import Input,Flatten,Conv2D,Dense,MaxPooling2D
```

In [2]:
```
from keras.utils import image_dataset_from_directory
```

```
splitting data into train and test
```

In [21]:
```
import os
import random
import shutil

data_path = r"C:\Users\range\Downloads\archive (21)\Car\auto rickshaw"
train_folder = os.path.join(r"C:\Users\range\Downloads\archive (21)\trn", 'vehicle')
test_folder = os.path.join(r"C:\Users\range\Downloads\archive (21)\tst", 'vehicle')

image_extensions = ['.jpg']

imgs_list = [filename for filename in os.listdir(data_path) if os.path.splitext(filename)[-1]
             in image_extensions]

random.seed(42)

random.shuffle(imgs_list)

train_size = int(len(imgs_list) * 0.7)
test_size = int(len(imgs_list) * 0.3)

for folder_path in [train_folder,test_folder]:
    if not os.path.exists(folder_path):
        os.makedirs(folder_path)

for i, f in enumerate(imgs_list):
    if i < train_size:
        dest_folder = train_folder
    else:
        dest_folder = test_folder
    shutil.copy(os.path.join(data_path, f), os.path.join(dest_folder, f))
```

In [39]:
```
trnn=image_dataset_from_directory(r"C:\Users\range\Downloads\archive (21)\trn",
                                  labels='inferred',label_mode="int",batch_size=32,image_size=(256,256))
```

```
Found 1536 files belonging to 6 classes.
```

In [23]:
```
trn.class_names
```

Out[23]:
```
['bike', 'bycycle', 'car', 'rickshaw', 'taxi', 'truck']
```

In [40]:
```
tstt=image_dataset_from_directory(r"C:\Users\range\Downloads\archive (21)\tst",
                                  labels='inferred',label_mode="int",batch_size=32,image_size=(256,256))
```

```
Found 664 files belonging to 6 classes.
```

inserting the convolutional base with input base as 256x256 the pixel size and 3 for images being in RGB

In [46]:
```
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(256, 256, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
```

selecting 6 at final layers as there are 6 types of vehicles in dataset

In [47]:
```
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(6))
```

In [48]:
```
from keras.callbacks import TensorBoard
```

In [49]:
```
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

history = model.fit(trn, epochs=10,
                    validation_data=tst,callbacks=TB)
```

```
Epoch 1/10
48/48 [==============================] - 102s 2s/step - loss: 1.8377 - accuracy: 0.3092 - val_loss: 2.1007 -
val_accuracy: 0.2123
Epoch 2/10
48/48 [==============================] - 95s 2s/step - loss: 1.3679 - accuracy: 0.4655 - val_loss: 2.2612 -
val_accuracy: 0.2123
Epoch 3/10
48/48 [==============================] - 96s 2s/step - loss: 0.9691 - accuracy: 0.6484 - val_loss: 4.1425 -
val_accuracy: 0.2123
Epoch 4/10
48/48 [==============================] - 97s 2s/step - loss: 0.5423 - accuracy: 0.8034 - val_loss: 5.1054 -
val_accuracy: 0.2123
Epoch 5/10
48/48 [==============================] - 96s 2s/step - loss: 0.2092 - accuracy: 0.9401 - val_loss: 3.4220 -
val_accuracy: 0.2123
Epoch 6/10
48/48 [==============================] - 97s 2s/step - loss: 0.0848 - accuracy: 0.9798 - val_loss: 4.5323 -
val_accuracy: 0.2123
Epoch 7/10
48/48 [==============================] - 96s 2s/step - loss: 0.0805 - accuracy: 0.9811 - val_loss: 6.9762 -
val_accuracy: 0.2123
Epoch 8/10
48/48 [==============================] - 96s 2s/step - loss: 0.0260 - accuracy: 0.9915 - val_loss: 4.9798 -
val_accuracy: 0.2123
Epoch 9/10
48/48 [==============================] - 98s 2s/step - loss: 0.0160 - accuracy: 0.9967 - val_loss: 5.8343 -
val_accuracy: 0.2123
Epoch 10/10
48/48 [==============================] - 97s 2s/step - loss: 0.0062 - accuracy: 0.9987 - val_loss: 6.6756 -
val_accuracy: 0.2123
```

In [36]:
```
TB = TensorBoard(log_dir=r"C:\Users\range\Downloads\archive (21)",histogram_freq=1)
```

In [52]:
```
%reload_ext tensorboard
%tensorboard --logdir "C:\Users\range\Downloads\archive (21)"
```

TensorBoard TIME SERIES SCALARS GRAPHS DISTRIBUTI INACTIVE

Filter runs (regex)

Run

train

validat ion

Filter tags (regex) All Scalars Image Histogram Settings

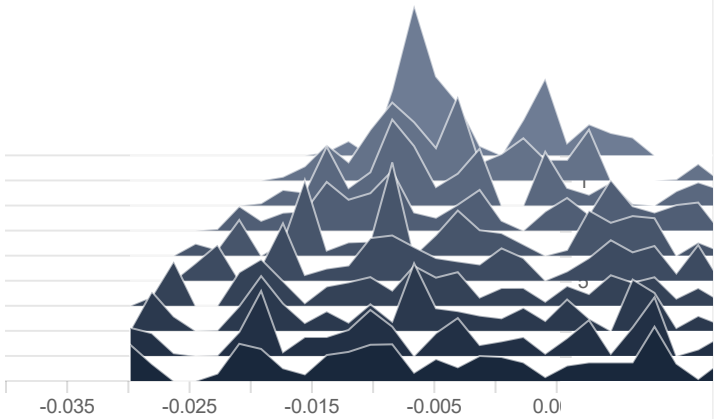
Pinned

Pin cards for a quick view and comparison

conv2d_3 2 cards

conv2d_3/bias_0/histogram

train



conv2d_3/kernel_0/histogram

train

Settings

GENERAL

Horizontal Axis

Step

Enable step selection and data table (Scalars only)

Enable Range Selection

Link by step 9

Card Width

SCALARS

Smoothing

0.7

Tooltip sorting method

Alphabetical

Ignore outliers in chart scaling

Partition non-monotonic X axis

HISTOGRAMS

Mode

Offset

In [53]:
```
%reload_ext tensorboard
%tensorboard --logdir "C:\Users\range\Downloads\archive (21)"
```

```
Reusing TensorBoard on port 6006 (pid 15156), started 0:00:57 ago. (Use '!kill 15156' to kill it.)
```

In [ ]:

TensorBoard TIME SERIES SCALARS GRAPHS DISTRIBUTI INACTIVE

Show data download links

Ignore outliers in chart scalin

Tooltip sorting method: default

Smoothing 0.6

Horizontal Axis

Filter tags (regular expressions supported)

epoch_accuracy

epoch_accuracy
tag: epoch_accuracy