jyputer_model

June 22, 2019

[1]:
```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from keras.applications import MobileNet
from tensorflow.keras import layers
from keras import regularizers
from keras.layers.convolutional import Convolution2D, MaxPooling2D,UpSampling2D
from keras.layers import Activation, Flatten, Dense, Dropout,Conv2D,␣
 ↪GaussianNoise,LocallyConnected2D
from keras.layers.normalization import BatchNormalization
from keras.layers import Dense,GlobalAveragePooling2D,Input, LeakyReLU
from keras.callbacks import LearningRateScheduler
import keras
import matplotlib.pyplot as plt
from keras.preprocessing.image import ImageDataGenerator


path_var_cache="G:\\CIFAR10\\cifar-10-python\\variable_Cache\\"


def plot_model_history(model_history):
    fig, axs = plt.subplots(1,2,figsize=(15,5))
    # summarize history for accuracy
    axs[0].plot(range(1,len(model_history.history['acc'])+1),model_history.
 ↪history['acc'])
    axs[0].plot(range(1,len(model_history.history['val_acc'])+1),model_history.
 ↪history['val_acc'])
    axs[0].set_title('Model Accuracy')
    axs[0].set_ylabel('Accuracy')
    axs[0].set_xlabel('Epoch')
    axs[0].set_xticks(np.arange(1,len(model_history.
 ↪history['acc'])+1),len(model_history.history['acc'])/10)
    axs[0].legend(['train', 'val'], loc='best')
    # summarize history for loss
    axs[1].plot(range(1,len(model_history.history['loss'])+1),model_history.
 ↪history['loss'])
```

```
    axs[1].plot(range(1,len(model_history.history['val_loss'])+1),model_history.
 ↪history['val_loss'])
    axs[1].set_title('Model Loss')
    axs[1].set_ylabel('Loss')
    axs[1].set_xlabel('Epoch')
    axs[1].set_xticks(np.arange(1,len(model_history.
 ↪history['loss'])+1),len(model_history.history['loss'])/10)
    axs[1].legend(['train', 'val'], loc='best')
    plt.show()


def lr_schedule(epoch):
    lrate = 0.0005
    if epoch > 25:
        lrate = 0.00001
    if epoch > 45:
        lrate = 0.000005
    return lrate

def load_batch(num):
    x=np.load(path_var_cache+'x'+str(num+3)+'.npy')
    y=np.load(path_var_cache+'y'+str(num+3)+'.npy')
    return x,y

x_train=np.load(path_var_cache+'x_train.npy')
y_train=np.load(path_var_cache+'y_train.npy')

x_train=(x_train.astype(np.float32))/255

x_val=np.load(path_var_cache+'x_val.npy')
x_val=(x_val.astype(np.float32))/255

y_val=np.load(path_var_cache+'y_val.npy')

x_test=np.load(path_var_cache+'x_test.npy')
x_test=(x_test.astype(np.float32))/255

y_test=np.load(path_var_cache+'y_test.npy')


def cnn_transfer(weight_decay):

    input_layer=Input(shape=(32,32,3))

    upsample=UpSampling2D((2,2))(input_layer)
```

```
   upsample=Conv2D(3, (5,5), padding='valid', kernel_regularizer=regularizers.
↪l2(weight_decay),activation='relu')(upsample)
   upsample=UpSampling2D((2,2))(upsample)

   upsample=Conv2D(3, (7,7), padding='valid', kernel_regularizer=regularizers.
↪l2(weight_decay),activation='relu')(upsample)
   upsample=UpSampling2D((2,2))(upsample)

   upsample=Conv2D(3, (5,5), padding='valid', kernel_regularizer=regularizers.
↪l2(weight_decay),activation='relu')(upsample)

   upsample=base_model(upsample)

   upsample=Conv2D(512, (3,3), padding='valid',␣
↪kernel_regularizer=regularizers.l2(weight_decay),activation='relu')(upsample)

   upsample=MaxPooling2D((2,2))(upsample)
   upsample=Dropout(0.3)(upsample)

   out=Flatten()(upsample)




   conv_net=Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.
↪l2(weight_decay),activation='relu')(input_layer)
   conv_net=BatchNormalization()(conv_net)

   conv_net=Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.
↪l2(weight_decay),activation='relu')(conv_net)
   conv_net=BatchNormalization()(conv_net)

   conv_net=MaxPooling2D(pool_size=(2,2))(conv_net)
   conv_net=Dropout(0.2)(conv_net)

   conv_net=Conv2D(96, (3,3), padding='same', kernel_regularizer=regularizers.
↪l2(weight_decay),activation='relu')(conv_net)
   conv_net=BatchNormalization()(conv_net)

   conv_net=Conv2D(96, (3,3), padding='same', kernel_regularizer=regularizers.
↪l2(weight_decay),activation='relu')(conv_net)
   conv_net=BatchNormalization()(conv_net)

   conv_net=MaxPooling2D(pool_size=(2,2))(conv_net)
   conv_net=Dropout(0.3)(conv_net)
```

```
    conv_net=Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.
 ↪l2(weight_decay),activation='relu')(conv_net)
    conv_net=BatchNormalization()(conv_net)

    conv_net=Conv2D(128, (3,3), padding='same', kernel_regularizer=regularizers.
 ↪l2(weight_decay),activation='relu')(conv_net)
    conv_net=BatchNormalization()(conv_net)

    conv_net=MaxPooling2D(pool_size=(2,2))(conv_net)
    conv_net=Dropout(0.3)(conv_net)
#
#    conv_net=Conv2D(256, (3,3), padding='same',␣
 ↪kernel_regularizer=regularizers.l2(weight_decay),activation='relu')(conv_net)
#    conv_net=BatchNormalization()(conv_net)




    conv_net=Flatten()(conv_net)

    concat=keras.layers.concatenate([conv_net,out])

    concat=Dense(1024,kernel_regularizer=regularizers.l2(weight_decay))(concat)
    concat=LeakyReLU(0.3)(concat)
    concat=GaussianNoise(0.05)(concat)

    pred=Dense(10,activation='softmax')(concat)

    model=keras.Model(inputs=[input_layer],outputs=[pred])

    return model







def cnn_cifar(weight_decay):


    model = keras.Sequential()

    model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.
 ↪l2(weight_decay), input_shape=x_train.shape[1:]))
    model.add(Activation('relu'))
```

```
    model.add(BatchNormalization())
    model.add(Conv2D(32, (3,3), padding='same', kernel_regularizer=regularizers.
 ↪l2(weight_decay)))
    model.add(Activation('relu'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.2))

    model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.
 ↪l2(weight_decay)))
    model.add(Activation('relu'))
    model.add(BatchNormalization())
    model.add(Conv2D(64, (3,3), padding='same', kernel_regularizer=regularizers.
 ↪l2(weight_decay)))
    model.add(Activation('relu'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.3))

    model.add(Conv2D(128, (3,3), padding='same',␣
 ↪kernel_regularizer=regularizers.l2(weight_decay)))
    model.add(Activation('relu'))
    model.add(BatchNormalization())
    model.add(Conv2D(128, (3,3), padding='same',␣
 ↪kernel_regularizer=regularizers.l2(weight_decay)))
    model.add(Activation('relu'))

    model.add(LocallyConnected2D(128, (3,3), padding='valid',␣
 ↪kernel_regularizer=regularizers.l2(weight_decay)))
    model.add(Activation('relu'))

    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.3))
    model.add(Flatten())
    model.add(Dense(1024, activation='relu', kernel_regularizer=regularizers.
 ↪l2(weight_decay)))
    model.add(GaussianNoise(0.05))
#     model.add(Dense(1024, activation='relu'))
    model.add(Dense(10, activation='softmax'))


    return model

base_model=MobileNet(input_shape=(224,224,3),include_top=False,weights='imagenet')
```

```
#model=cnn_cifar(1e-4)
#print(model.summary())


#
model=cnn_transfer(1e-4)
print(model.summary())


mean = np.mean(x_train,axis=(0,1,2,3))
std = np.std(x_train,axis=(0,1,2,3))
x_train = (x_train-mean)/(std)
x_test = (x_test-mean)/(std)
x_val = (x_val-mean)/std



optimizer=keras.optimizers.rmsprop(lr=0.0005,decay=1e-6)
model.compile(optimizer=optimizer,loss=keras.losses.
 ↪categorical_crossentropy,metrics=['acc'])



#model.load_weights('akshit')

epochs=125
datagen = ImageDataGenerator(rotation_range=30,width_shift_range=0.
 ↪2,height_shift_range=0.2,horizontal_flip=True)
datagen.fit(x_train)

model_info=model.fit_generator(datagen.flow(x_train, y_train,␣
 ↪batch_size=32),validation_data=(x_val,y_val),steps_per_epoch=len(x_train)/
 ↪32, epochs=75,callbacks=[LearningRateScheduler(lr_schedule)])

plot_model_history(model_info)

print('Testing data')
acc_test=model.evaluate(x_test,y_test)
print('Test ACC = '+str(acc_test))
```

```
Using TensorFlow backend.

WARNING:tensorflow:From C:\Users\Akshit\Anaconda3\envs\akshit\lib\site-
packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from
```

```
tensorflow.python.framework.ops) is deprecated and will be removed in a future
version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From C:\Users\Akshit\Anaconda3\envs\akshit\lib\site-
packages\keras\backend\tensorflow_backend.py:3445: calling dropout (from
tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed
in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 -
keep_prob`.
_________________________________________________________________________________
__________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================
==================
input_2 (InputLayer)         (None, 32, 32, 3)         0
_________________________________________________________________________________
__________________
conv2d_5 (Conv2D)            (None, 32, 32, 64)        1792        input_2[0][0]
_________________________________________________________________________________
__________________
batch_normalization_1 (BatchNor (None, 32, 32, 64)     256         conv2d_5[0][0]
_________________________________________________________________________________
__________________
conv2d_6 (Conv2D)            (None, 32, 32, 64)        36928
batch_normalization_1[0][0]
_________________________________________________________________________________
__________________
batch_normalization_2 (BatchNor (None, 32, 32, 64)     256         conv2d_6[0][0]
_________________________________________________________________________________
__________________
max_pooling2d_2 (MaxPooling2D)  (None, 16, 16, 64)     0
batch_normalization_2[0][0]
_________________________________________________________________________________
__________________
dropout_2 (Dropout)          (None, 16, 16, 64)        0
max_pooling2d_2[0][0]
_________________________________________________________________________________
__________________
conv2d_7 (Conv2D)            (None, 16, 16, 96)        55392       dropout_2[0][0]
_________________________________________________________________________________
__________________
batch_normalization_3 (BatchNor (None, 16, 16, 96)     384         conv2d_7[0][0]
_________________________________________________________________________________
__________________
conv2d_8 (Conv2D)            (None, 16, 16, 96)        83040
batch_normalization_3[0][0]
```

```
____________________________________________________________________________________________________
up_sampling2d_1 (UpSampling2D)  (None, 64, 64, 3)    0           input_2[0][0]
____________________________________________________________________________________________________
batch_normalization_4 (BatchNor (None, 16, 16, 96)   384         conv2d_8[0][0]
____________________________________________________________________________________________________
conv2d_1 (Conv2D)               (None, 60, 60, 3)    228
up_sampling2d_1[0][0]
____________________________________________________________________________________________________
max_pooling2d_3 (MaxPooling2D)  (None, 8, 8, 96)     0
batch_normalization_4[0][0]
____________________________________________________________________________________________________
up_sampling2d_2 (UpSampling2D)  (None, 120, 120, 3)  0           conv2d_1[0][0]
____________________________________________________________________________________________________
dropout_3 (Dropout)             (None, 8, 8, 96)     0
max_pooling2d_3[0][0]
____________________________________________________________________________________________________
conv2d_2 (Conv2D)               (None, 114, 114, 3)  444
up_sampling2d_2[0][0]
____________________________________________________________________________________________________
conv2d_9 (Conv2D)               (None, 8, 8, 128)    110720      dropout_3[0][0]
____________________________________________________________________________________________________
up_sampling2d_3 (UpSampling2D)  (None, 228, 228, 3)  0           conv2d_2[0][0]
____________________________________________________________________________________________________
batch_normalization_5 (BatchNor (None, 8, 8, 128)    512         conv2d_9[0][0]
____________________________________________________________________________________________________
conv2d_3 (Conv2D)               (None, 224, 224, 3)  228
up_sampling2d_3[0][0]
____________________________________________________________________________________________________
conv2d_10 (Conv2D)              (None, 8, 8, 128)    147584
batch_normalization_5[0][0]
____________________________________________________________________________________________________
mobilenet_1.00_224 (Model)      (None, 7, 7, 1024)   3228864     conv2d_3[0][0]
____________________________________________________________________________________________________
batch_normalization_6 (BatchNor (None, 8, 8, 128)    512         conv2d_10[0][0]
```

```
__________________________________________________________________________________
__________________
conv2d_4 (Conv2D)               (None, 5, 5, 512)    4719104
mobilenet_1.00_224[1][0]
__________________________________________________________________________________
__________________
max_pooling2d_4 (MaxPooling2D)  (None, 4, 4, 128)    0
batch_normalization_6[0][0]
__________________________________________________________________________________
__________________
max_pooling2d_1 (MaxPooling2D)  (None, 2, 2, 512)    0           conv2d_4[0][0]
__________________________________________________________________________________
__________________
dropout_4 (Dropout)             (None, 4, 4, 128)    0
max_pooling2d_4[0][0]
__________________________________________________________________________________
__________________
dropout_1 (Dropout)             (None, 2, 2, 512)    0
max_pooling2d_1[0][0]
__________________________________________________________________________________
__________________
flatten_2 (Flatten)             (None, 2048)         0           dropout_4[0][0]
__________________________________________________________________________________
__________________
flatten_1 (Flatten)             (None, 2048)         0           dropout_1[0][0]
__________________________________________________________________________________
__________________
concatenate_1 (Concatenate)     (None, 4096)         0           flatten_2[0][0]
                                                                 flatten_1[0][0]
__________________________________________________________________________________
__________________
dense_1 (Dense)                 (None, 1024)         4195328
concatenate_1[0][0]
__________________________________________________________________________________
__________________
leaky_re_lu_1 (LeakyReLU)       (None, 1024)         0           dense_1[0][0]
__________________________________________________________________________________
__________________
gaussian_noise_1 (GaussianNoise (None, 1024)         0
leaky_re_lu_1[0][0]
__________________________________________________________________________________
__________________
dense_2 (Dense)                 (None, 10)           10250
gaussian_noise_1[0][0]
==================================================================================
==================
Total params: 12,592,206
Trainable params: 12,569,166
```

```
Non-trainable params: 23,040
_________________________________________________________________________________
_________________
None
WARNING:tensorflow:From C:\Users\Akshit\Anaconda3\envs\akshit\lib\site-
packages\tensorflow\python\ops\math_ops.py:3066: to_int32 (from
tensorflow.python.ops.math_ops) is deprecated and will be removed in a future
version.
Instructions for updating:
Use tf.cast instead.
Epoch 1/75
1532/1531 [==============================] - 760s 496ms/step - loss: 3.6575 -
acc: 0.4268 - val_loss: 1.7260 - val_acc: 0.5450
Epoch 2/75
1532/1531 [==============================] - 772s 504ms/step - loss: 1.0963 -
acc: 0.6629 - val_loss: 1.1442 - val_acc: 0.6600
Epoch 3/75
1532/1531 [==============================] - 788s 514ms/step - loss: 0.8706 -
acc: 0.7311 - val_loss: 0.8082 - val_acc: 0.7710
Epoch 4/75
1532/1531 [==============================] - 798s 521ms/step - loss: 0.7499 -
acc: 0.7693 - val_loss: 0.5504 - val_acc: 0.8520
Epoch 5/75
1532/1531 [==============================] - 782s 511ms/step - loss: 0.6737 -
acc: 0.7928 - val_loss: 0.5809 - val_acc: 0.8160
Epoch 6/75
1532/1531 [==============================] - 775s 506ms/step - loss: 0.6203 -
acc: 0.8119 - val_loss: 0.5088 - val_acc: 0.8450
Epoch 7/75
1532/1531 [==============================] - 776s 506ms/step - loss: 0.5739 -
acc: 0.8259 - val_loss: 0.5397 - val_acc: 0.8390
Epoch 8/75
1532/1531 [==============================] - 775s 506ms/step - loss: 0.5388 -
acc: 0.8373 - val_loss: 0.5195 - val_acc: 0.8630
Epoch 9/75
1532/1531 [==============================] - 777s 507ms/step - loss: 0.5083 -
acc: 0.8457 - val_loss: 0.4759 - val_acc: 0.8750
Epoch 10/75
1532/1531 [==============================] - 777s 507ms/step - loss: 0.4840 -
acc: 0.8545 - val_loss: 0.4747 - val_acc: 0.8580
Epoch 11/75
1532/1531 [==============================] - 782s 511ms/step - loss: 0.4590 -
acc: 0.8635 - val_loss: 0.4260 - val_acc: 0.8940
Epoch 12/75
1532/1531 [==============================] - 789s 515ms/step - loss: 0.4416 -
acc: 0.8684 - val_loss: 0.5681 - val_acc: 0.8430
Epoch 13/75
1532/1531 [==============================] - 784s 512ms/step - loss: 0.4268 -
```

```
acc: 0.8732 - val_loss: 0.4399 - val_acc: 0.8800
Epoch 14/75
1532/1531 [==============================] - 779s 508ms/step - loss: 0.4099 -
acc: 0.8782 - val_loss: 0.4796 - val_acc: 0.8830
Epoch 15/75
1532/1531 [==============================] - 778s 508ms/step - loss: 0.3913 -
acc: 0.8850 - val_loss: 0.5043 - val_acc: 0.8750
Epoch 16/75
1532/1531 [==============================] - 781s 510ms/step - loss: 0.3804 -
acc: 0.8870 - val_loss: 0.5651 - val_acc: 0.8620
Epoch 17/75
1532/1531 [==============================] - 781s 510ms/step - loss: 0.3713 -
acc: 0.8875 - val_loss: 0.4343 - val_acc: 0.8990
Epoch 18/75
1532/1531 [==============================] - 777s 507ms/step - loss: 0.3579 -
acc: 0.8959 - val_loss: 0.3981 - val_acc: 0.8860
Epoch 19/75
1532/1531 [==============================] - 767s 501ms/step - loss: 0.3551 -
acc: 0.8973 - val_loss: 0.4650 - val_acc: 0.8830
Epoch 20/75
1532/1531 [==============================] - 768s 502ms/step - loss: 0.3459 -
acc: 0.8989 - val_loss: 0.3714 - val_acc: 0.8990
Epoch 21/75
1532/1531 [==============================] - 771s 503ms/step - loss: 0.3385 -
acc: 0.9019 - val_loss: 0.4310 - val_acc: 0.8920
Epoch 22/75
1532/1531 [==============================] - 773s 504ms/step - loss: 0.3251 -
acc: 0.9059 - val_loss: 0.3170 - val_acc: 0.9130
Epoch 23/75
1532/1531 [==============================] - 774s 505ms/step - loss: 0.3227 -
acc: 0.9062 - val_loss: 0.4064 - val_acc: 0.8940
Epoch 24/75
1532/1531 [==============================] - 772s 504ms/step - loss: 0.3182 -
acc: 0.9064 - val_loss: 0.3911 - val_acc: 0.8930
Epoch 25/75
1532/1531 [==============================] - 773s 505ms/step - loss: 0.3042 -
acc: 0.9120 - val_loss: 0.4549 - val_acc: 0.8870
Epoch 26/75
1532/1531 [==============================] - 772s 504ms/step - loss: 0.2994 -
acc: 0.9133 - val_loss: 0.8009 - val_acc: 0.8160
Epoch 27/75
1532/1531 [==============================] - 770s 502ms/step - loss: 0.2535 -
acc: 0.9263 - val_loss: 0.3163 - val_acc: 0.9170
Epoch 28/75
1532/1531 [==============================] - 773s 504ms/step - loss: 0.2244 -
acc: 0.9357 - val_loss: 0.3052 - val_acc: 0.9180
Epoch 29/75
1532/1531 [==============================] - 776s 506ms/step - loss: 0.2095 -
```

```
acc: 0.9395 - val_loss: 0.3036 - val_acc: 0.9200
Epoch 30/75
1532/1531 [==============================] - 777s 507ms/step - loss: 0.2068 -
acc: 0.9400 - val_loss: 0.2985 - val_acc: 0.9210
Epoch 31/75
1532/1531 [==============================] - 776s 507ms/step - loss: 0.1952 -
acc: 0.9433 - val_loss: 0.2946 - val_acc: 0.9210
Epoch 32/75
1532/1531 [==============================] - 767s 501ms/step - loss: 0.1978 -
acc: 0.9433 - val_loss: 0.2889 - val_acc: 0.9250
Epoch 33/75
1532/1531 [==============================] - 766s 500ms/step - loss: 0.1912 -
acc: 0.9453 - val_loss: 0.2834 - val_acc: 0.9260
Epoch 34/75
1532/1531 [==============================] - 765s 499ms/step - loss: 0.1882 -
acc: 0.9461 - val_loss: 0.2790 - val_acc: 0.9300
Epoch 35/75
1532/1531 [==============================] - 769s 502ms/step - loss: 0.1858 -
acc: 0.9467 - val_loss: 0.2835 - val_acc: 0.9250
Epoch 36/75
1532/1531 [==============================] - 781s 510ms/step - loss: 0.1848 -
acc: 0.9464 - val_loss: 0.2749 - val_acc: 0.9260
Epoch 37/75
1532/1531 [==============================] - 799s 521ms/step - loss: 0.1869 -
acc: 0.9473 - val_loss: 0.2761 - val_acc: 0.9280
Epoch 38/75
1532/1531 [==============================] - 804s 525ms/step - loss: 0.1811 -
acc: 0.9482 - val_loss: 0.2839 - val_acc: 0.9250
Epoch 39/75
1532/1531 [==============================] - 808s 527ms/step - loss: 0.1783 -
acc: 0.9477 - val_loss: 0.2746 - val_acc: 0.9280
Epoch 40/75
1532/1531 [==============================] - 810s 528ms/step - loss: 0.1758 -
acc: 0.9507 - val_loss: 0.2812 - val_acc: 0.9280
Epoch 41/75
1532/1531 [==============================] - 783s 511ms/step - loss: 0.1727 -
acc: 0.9505 - val_loss: 0.2773 - val_acc: 0.9260
Epoch 42/75
1532/1531 [==============================] - 778s 508ms/step - loss: 0.1770 -
acc: 0.9502 - val_loss: 0.2681 - val_acc: 0.9300
Epoch 43/75
1532/1531 [==============================] - 783s 511ms/step - loss: 0.1739 -
acc: 0.9511 - val_loss: 0.2697 - val_acc: 0.9280
Epoch 44/75
1532/1531 [==============================] - 775s 506ms/step - loss: 0.1721 -
acc: 0.9508 - val_loss: 0.2707 - val_acc: 0.9280
Epoch 45/75
1532/1531 [==============================] - 771s 503ms/step - loss: 0.1721 -
```

```
acc: 0.9514 - val_loss: 0.2712 - val_acc: 0.9290
Epoch 46/75
1532/1531 [==============================] - 769s 502ms/step - loss: 0.1675 -
acc: 0.9529 - val_loss: 0.2702 - val_acc: 0.9260
Epoch 47/75
1532/1531 [==============================] - 772s 504ms/step - loss: 0.1646 -
acc: 0.9534 - val_loss: 0.2709 - val_acc: 0.9270
Epoch 48/75
1532/1531 [==============================] - 771s 503ms/step - loss: 0.1679 -
acc: 0.9520 - val_loss: 0.2724 - val_acc: 0.9310
Epoch 49/75
1532/1531 [==============================] - 776s 507ms/step - loss: 0.1631 -
acc: 0.9534 - val_loss: 0.2718 - val_acc: 0.9270
Epoch 50/75
1532/1531 [==============================] - 783s 511ms/step - loss: 0.1669 -
acc: 0.9529 - val_loss: 0.2748 - val_acc: 0.9280
Epoch 51/75
1532/1531 [==============================] - 783s 511ms/step - loss: 0.1623 -
acc: 0.9539 - val_loss: 0.2796 - val_acc: 0.9290
Epoch 52/75
1532/1531 [==============================] - 784s 512ms/step - loss: 0.1623 -
acc: 0.9538 - val_loss: 0.2687 - val_acc: 0.9320
Epoch 53/75
1532/1531 [==============================] - 784s 512ms/step - loss: 0.1638 -
acc: 0.9526 - val_loss: 0.2708 - val_acc: 0.9320
Epoch 54/75
1532/1531 [==============================] - 784s 512ms/step - loss: 0.1611 -
acc: 0.9544 - val_loss: 0.2702 - val_acc: 0.9310
Epoch 55/75
1532/1531 [==============================] - 780s 509ms/step - loss: 0.1648 -
acc: 0.9533 - val_loss: 0.2700 - val_acc: 0.9330
Epoch 56/75
1532/1531 [==============================] - 774s 505ms/step - loss: 0.1608 -
acc: 0.9539 - val_loss: 0.2716 - val_acc: 0.9290
Epoch 57/75
1532/1531 [==============================] - 783s 511ms/step - loss: 0.1614 -
acc: 0.9542 - val_loss: 0.2734 - val_acc: 0.9270
Epoch 58/75
1532/1531 [==============================] - 785s 512ms/step - loss: 0.1606 -
acc: 0.9535 - val_loss: 0.2707 - val_acc: 0.9330
Epoch 59/75
1532/1531 [==============================] - 783s 511ms/step - loss: 0.1613 -
acc: 0.9549 - val_loss: 0.2711 - val_acc: 0.9350
Epoch 60/75
1532/1531 [==============================] - 781s 510ms/step - loss: 0.1610 -
acc: 0.9546 - val_loss: 0.2739 - val_acc: 0.9300
Epoch 61/75
1532/1531 [==============================] - 776s 506ms/step - loss: 0.1596 -
```

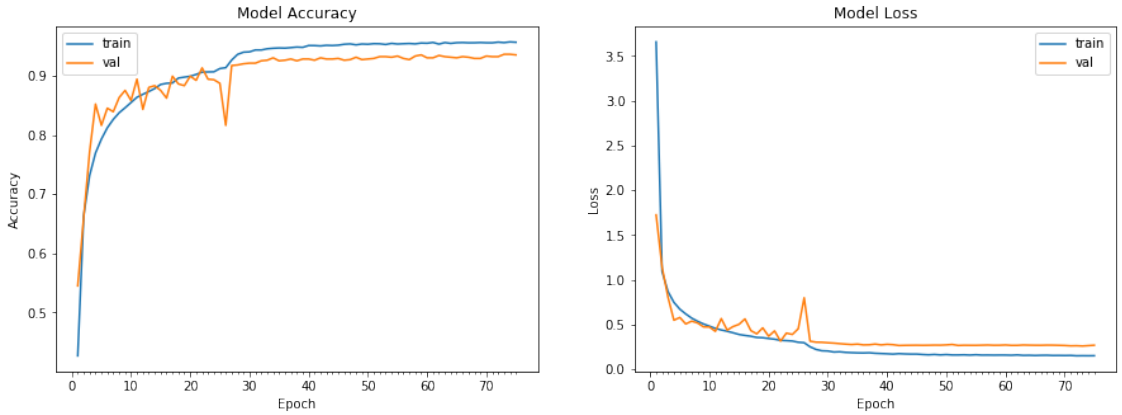
```
acc: 0.9559 - val_loss: 0.2694 - val_acc: 0.9300
Epoch 62/75
1532/1531 [==============================] - 779s 508ms/step - loss: 0.1630 -
acc: 0.9531 - val_loss: 0.2697 - val_acc: 0.9340
Epoch 63/75
1532/1531 [==============================] - 779s 508ms/step - loss: 0.1583 -
acc: 0.9557 - val_loss: 0.2736 - val_acc: 0.9320
Epoch 64/75
1532/1531 [==============================] - 779s 509ms/step - loss: 0.1590 -
acc: 0.9543 - val_loss: 0.2719 - val_acc: 0.9310
Epoch 65/75
1532/1531 [==============================] - 779s 509ms/step - loss: 0.1570 -
acc: 0.9555 - val_loss: 0.2709 - val_acc: 0.9300
Epoch 66/75
1532/1531 [==============================] - 780s 509ms/step - loss: 0.1585 -
acc: 0.9558 - val_loss: 0.2706 - val_acc: 0.9320
Epoch 67/75
1532/1531 [==============================] - 779s 509ms/step - loss: 0.1595 -
acc: 0.9553 - val_loss: 0.2722 - val_acc: 0.9310
Epoch 68/75
1532/1531 [==============================] - 779s 508ms/step - loss: 0.1566 -
acc: 0.9555 - val_loss: 0.2718 - val_acc: 0.9290
Epoch 69/75
1532/1531 [==============================] - 778s 508ms/step - loss: 0.1571 -
acc: 0.9559 - val_loss: 0.2701 - val_acc: 0.9290
Epoch 70/75
1532/1531 [==============================] - 778s 508ms/step - loss: 0.1561 -
acc: 0.9555 - val_loss: 0.2683 - val_acc: 0.9330
Epoch 71/75
1532/1531 [==============================] - 775s 506ms/step - loss: 0.1575 -
acc: 0.9553 - val_loss: 0.2645 - val_acc: 0.9320
Epoch 72/75
1532/1531 [==============================] - 782s 510ms/step - loss: 0.1534 -
acc: 0.9565 - val_loss: 0.2656 - val_acc: 0.9320
Epoch 73/75
1532/1531 [==============================] - 779s 509ms/step - loss: 0.1540 -
acc: 0.9557 - val_loss: 0.2625 - val_acc: 0.9360
Epoch 74/75
1532/1531 [==============================] - 781s 510ms/step - loss: 0.1534 -
acc: 0.9569 - val_loss: 0.2668 - val_acc: 0.9360
Epoch 75/75
1532/1531 [==============================] - 780s 509ms/step - loss: 0.1541 -
acc: 0.9563 - val_loss: 0.2711 - val_acc: 0.9350

<Figure size 1500x500 with 2 Axes>


Testing data
```

```
10000/10000 [==============================] - 40s 4ms/step
Test ACC = [0.31100748220682145, 0.931]
```

[3]:
```
plot_model_history(model_info)
```



[5]:
```
import h5py
model.save_weights('akshit_verma.h5')
```

[ ]:
```
```