

1. Loading images
2. Resizing images
3. Converting images to arrays
4. Normalization

```

from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

import sys
sys.path.append('/drive/MyDrive/Dir/')
%cd drive/MyDrive/Dir
%pwd

/content/drive/MyDrive/Dir

{"type": "string"}

from matplotlib import image
from matplotlib import pyplot
from PIL import Image
import glob
from numpy import asarray
import natsort
folders = ["0010", "0011", "0018", "0024", "0030", "0037", "0050"]
loaded_images = list()
for folder in folders:
    # load all images in a directory
    files = glob.glob(f"Data-YCB/YCB_Video/data/{folder}/*color.png")
    files = (natsort.natsorted(files, reverse=False))
    for filename in files:
        # load image
        img = Image.open(filename)
        # resize image for the input layer
        img = img.resize((32,32))
        # convert to array
        img = asarray(img)
        #normalize
        img = img/255.
        # store loaded image
        loaded_images.append(img)
        # print('> loaded %s %s' % (filename, img.shape))

import numpy as np
data = np.array(loaded_images)
data.shape

(1842, 32, 32, 3)

```

### Labels

```

import pandas as pd
csvs = glob.glob(f"CSV/*.csv")
csvs = (natsort.natsorted(csvs, reverse=False))
print(csvs)
# csvs = ["10", "11", "18", "24", "30", "37", "50"]
labels = pd.DataFrame()

```

```

for csv in csvs:
    label = pd.read_csv(csv, header=None)
    labels = labels.append(label, ignore_index=True)

print(labels.shape)
labels.drop(0, axis = 1, inplace=True)
labels.to_numpy()

['CSV/10.csv', 'CSV/11.csv', 'CSV/18.csv', 'CSV/24.csv', 'CSV/30.csv',
 'CSV/37.csv', 'CSV/50.csv']
(1842, 2)

array([[ 2],
       [ 2],
       [ 2],
       ...,
       [21],
       [21],
       [21]])

```

*Split data into training, testing and validation subset*

```

from sklearn.model_selection import train_test_split

data_train, data_test, labels_train, labels_test =
train_test_split(data, labels, test_size=0.20, random_state=42)

print(data_train.shape)
print(data_test.shape)
print(labels_train.shape)
print(labels_test.shape)

(1473, 32, 32, 3)
(369, 32, 32, 3)
(1473, 1)
(369, 1)

x_train = data_train[:1353, :, :, :]
y_train = labels_train[:1353]
x_val = x_train[-120:, :, :, :]
y_val = y_train[-120:]
x_test = data_test
y_test = labels_test
print('x_train shape:', x_train.shape)
print('y_train shape:', y_train.shape)
print('x_validation shape:', x_val.shape)
print('y_validation shape:', y_val.shape)
print('x_test shape:', x_test.shape)
print('y_test shape:', y_test.shape)

```

*# Convert class vectors to binary class matrices.*

```

y_train = to_categorical(y_train, 24)

```

```

y_val = to_categorical(y_val, 24)
y_test = to_categorical(y_test, 24)

x_train shape: (1353, 32, 32, 3)
y_train shape: (1353, 1)
x_validation shape: (120, 32, 32, 3)
y_validation shape: (120, 1)
x_test shape: (369, 32, 32, 3)
y_test shape: (369, 1)

```

**The previous operations gave the following results:**

The shape of inputs: 32x32x3

Size of datasets:

- training set - 1353 elements
- testing set - 369 elements
- validation set - 120 elements

*Import libraries for model definition*

```

import sys
sys.path.append('/Module/')
%cd Module
%pwd

/content/drive/MyDrive/Dir/Module

{"type": "string"}

import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
import skimage.transform
import time, os, datetime
from Residual_Unit import Residual_Unit
from Attention_Block import Attention_Block

from tensorflow.keras.utils import to_categorical
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam, SGD
from tensorflow.keras.callbacks import ReduceLROnPlateau,
EarlyStopping, TensorBoard, LearningRateScheduler
from tensorflow.keras.models import load_model
from tensorflow.keras.layers import Input
from tensorflow.keras.regularizers import l2
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Activation
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import Flatten

```

```

from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Model

print("TF version: ",tf.__version__)
print("Keras version:",tf.keras.__version__)

TF version: 2.8.2
Keras version: 2.8.0

```

## Data Augmentation

```

# define generators for training and validation data
train_datagen = ImageDataGenerator(
    featurewise_center=True,
    featurewise_std_normalization=True,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    validation_split=0.2)

val_datagen = ImageDataGenerator(
    featurewise_center=True,
    featurewise_std_normalization=True)

test_datagen = ImageDataGenerator(
    featurewise_center=True,
    featurewise_std_normalization=True)

# compute quantities required for feature normalization
train_datagen.fit(x_train)
val_datagen.fit(x_val)
test_datagen.fit(x_test)

```

## Residual Attention Network

All models have been created with the following parameters:

- input shape - 32x32x32
- activation function - Relu
- kernel size - 5
- number of classes - 24
- loss function - categorical cross-entropy
- output Dense layer activation function - softmax
- number of epochs - 10

The Network is built based on following elements:

- Convolutional Layer
- Batch Normalization
- Activation Layer
- Max Pooling
- **Residual Unit**
- **Attention Block**
- Average Pooling
- Flatten Layer
- Dense Layer

```
# define learning rate scheduler
```

```
def lr_schedule(epoch):
```

```
    lr = 1e-4
```

```
    if epoch > 50:
```

```
        lr *= 1e-2
```

```
    elif epoch > 20:
```

```
        lr *= 1e-1
```

```
    print('Learning rate:', lr)
```

```
    return lr
```

```
lr_scheduler = LearningRateScheduler(lr_schedule)
```

```
# Residual Attention Network
```

```
def AttentionResNet56_mini(shape, in_channel, kernel_size, n_classes,
dropout=None, regularization=0.01):
```

```
    """
```

```
        :param shape: The tuple of input data.
```

```
        :param in_channel: The 4-th dimension (channel number) of input
weight matrix. For example, in_channel=3 means the input contains 3
channels.
```

```
        :param kernel_size: Integer. the shape of the kernel. For example,
default kernel_size = 3 means you have a 3*3 kernel.
```

```
        :param n_classes: Integer. The number of target classes. For
example, n_classes = 10 means you have 10 class labels.
```

```
        :param dropout: Float between 0 and 1. Fraction of the input units
to drop.
```

```
        :param regularization: Float. Fraction of the input units to drop.
```

```
    """
```

```
    input_data = Input(shape=shape) # 32x32x32
```

```
    x = Conv2D(in_channel, kernel_size=kernel_size, padding='same')
(input_data) # 32x32x32
```

```
    x = BatchNormalization()(x)
```

```
    x = Activation('relu')(x)
```

```
    x = MaxPooling2D(pool_size=2, padding='same')(x) # 16x16x32
```

```
    out_channel = in_channel * 4
```

```
    x = Residual_Unit(x, in_channel, out_channel) # 16x16x128
```

```
    x = Attention_Block(x, skip=1)
```

```

    in_channel = out_channel // 2
    out_channel = in_channel * 4
    x = Residual_Unit(x, in_channel, out_channel, stride=2) # 8x8x256
    x = Attention_Block(x, skip=1)

    in_channel = out_channel // 2
    out_channel = in_channel * 4
    x = Residual_Unit(x, in_channel, out_channel, stride=1) #
4x4x1024
    x = Residual_Unit(x, in_channel, out_channel)
    x = Residual_Unit(x, in_channel, out_channel)

    x = AveragePooling2D(pool_size=4, strides=1)(x) # 1x1x1024
    x = Flatten()(x)

    output = Dense(n_classes, activation='softmax')(x)
    model = Model(input_data, output)

    return model

def training(model, log_name, batch_size=128, epc=10):
    batch_size = batch_size
    epc = epc

    start = time.time()

    # define training generator
    train_generator = train_datagen.flow(x_train, y_train,
batch_size=batch_size)
    step_size_train = train_generator.n // train_generator.batch_size

    # define validation generator
    val_generator = val_datagen.flow(x_val, y_val,
batch_size=batch_size)
    step_size_val = val_generator.n // val_generator.batch_size

    # define test validation generator
    test_generator = test_datagen.flow(x_test, y_test,
batch_size=batch_size)
    step_size_test = test_generator.n // test_generator.batch_size

    # usefull callbacks
    log_dir='Logs/' + log_name
    tensorboard_callback = TensorBoard(log_dir=log_dir,
histogram_freq=1)
    lr_reducer = ReduceLROnPlateau(monitor='val_accuracy', factor=0.1,
patience=5, verbose=1)
    early_stopper = EarlyStopping(monitor='val_accuracy', patience=15,

```

```

verbose=1)

    model.fit_generator(train_generator,
                        steps_per_epoch = step_size_train,
                        epochs = epc,
                        validation_data = val_generator,
                        validation_steps = step_size_val,
                        callbacks=[tensorboard_callback, lr_reducer,
lr_scheduler, early_stopper])

    end = time.time()
    print("Time taken by above cell is {}".format((end-start)/60))

    # evaluation
    val_scores = model.evaluate_generator(val_generator, verbose=0)
    test_scores = model.evaluate_generator(test_generator, verbose=1)
    print('validation loss:', val_scores[0])
    print('validation accuracy:', val_scores[1])
    print('Test loss:', test_scores[0])
    print('Test accuracy:', test_scores[1])

    return model

```

## First model

The first model consists of the following parameters:

- optimizer - SGD (nesterov)
- no dropout
- no regularization

```

# define model
model = AttentionResNet56_mini(shape=(32,32,3), in_channel=32,
kernel_size=5, n_classes = 24, dropout=None, regularization=None)

# define loss, metrics, optimizer
optimizer = SGD(lr = lr_schedule(0), momentum=0.9, nesterov=True)
# optimizer = Adam(lr = lr_schedule(0) )
model.compile(optimizer, loss="categorical_crossentropy",
metrics=['accuracy'])
model.summary()

```

Learning rate: 0.0001  
Model: "model\_6"

Layer (type) Connected to	Output Shape	Param #
=====		
input_7 (InputLayer)	[(None, 32, 32, 3)]	0
		[]



conv2d_514 (Conv2D) ['input_7[0][0]']	(None, 32, 32, 32)	2432
batch_normalization_367 (Batch ['conv2d_514[0][0]'] Normalization)	(None, 32, 32, 32)	128
activation_381 (Activation) ['batch_normalization_367[0][0]']	(None, 32, 32, 32)	0
max_pooling2d_22 (MaxPooling2D ['activation_381[0][0]'] )	(None, 16, 16, 32)	0
batch_normalization_368 (Batch ['max_pooling2d_22[0][0]'] Normalization)	(None, 16, 16, 32)	128
activation_382 (Activation) ['batch_normalization_368[0][0]']	(None, 16, 16, 32)	0
conv2d_516 (Conv2D) ['activation_382[0][0]']	(None, 16, 16, 32)	1056
batch_normalization_369 (Batch ['conv2d_516[0][0]'] Normalization)	(None, 16, 16, 32)	128
activation_383 (Activation) ['batch_normalization_369[0][0]']	(None, 16, 16, 32)	0
conv2d_517 (Conv2D)	(None, 16, 16, 32)	9248

['activation\_383[0][0]']

batch\_normalization\_370 (Batch Normalization) (None, 16, 16, 32) 128  
['conv2d\_517[0][0]']

activation\_384 (Activation) (None, 16, 16, 32) 0  
['batch\_normalization\_370[0][0]']

conv2d\_518 (Conv2D) (None, 16, 16, 128) 4224  
['activation\_384[0][0]']

conv2d\_515 (Conv2D) (None, 16, 16, 128) 4224  
['max\_pooling2d\_22[0][0]']

add\_122 (Add) (None, 16, 16, 128) 0  
['conv2d\_518[0][0]',  
'conv2d\_515[0][0]']

batch\_normalization\_371 (Batch Normalization) (None, 16, 16, 128) 512  
['add\_122[0][0]']

activation\_385 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_371[0][0]']

conv2d\_520 (Conv2D) (None, 16, 16, 128) 16512  
['activation\_385[0][0]']

batch\_normalization\_372 (Batch Normalization) (None, 16, 16, 128) 512  
['conv2d\_520[0][0]']

activation\_386 (Activation) (None, 16, 16, 128) 0

['batch\_normalization\_372[0][0]']

conv2d\_521 (Conv2D) (None, 16, 16, 128) 147584  
['activation\_386[0][0]']

batch\_normalization\_373 (Batch Normalization) (None, 16, 16, 128) 512  
['conv2d\_521[0][0]']

activation\_387 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_373[0][0]']

conv2d\_522 (Conv2D) (None, 16, 16, 128) 16512  
['activation\_387[0][0]']

conv2d\_519 (Conv2D) (None, 16, 16, 128) 16512  
['add\_122[0][0]']

add\_123 (Add) (None, 16, 16, 128) 0  
['conv2d\_522[0][0]',  
'conv2d\_519[0][0]']

max\_pooling2d\_23 (MaxPooling2D) (None, 8, 8, 128) 0  
['add\_123[0][0]']  
)

batch\_normalization\_380 (Batch Normalization) (None, 8, 8, 128) 512  
['max\_pooling2d\_23[0][0]']

activation\_394 (Activation) (None, 8, 8, 128) 0  
['batch\_normalization\_380[0][0]']

conv2d\_532 (Conv2D) (None, 8, 8, 128) 16512

```
['activation_394[0][0]']
```

```
batch_normalization_381 (Batch Normalization) (None, 8, 8, 128) 512  
['conv2d_532[0][0]']
```

```
activation_395 (Activation) (None, 8, 8, 128) 0  
['batch_normalization_381[0][0]']
```

```
conv2d_533 (Conv2D) (None, 8, 8, 128) 147584  
['activation_395[0][0]']
```

```
batch_normalization_382 (Batch Normalization) (None, 8, 8, 128) 512  
['conv2d_533[0][0]']
```

```
activation_396 (Activation) (None, 8, 8, 128) 0  
['batch_normalization_382[0][0]']
```

```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/  
gradient_descent.py:102: UserWarning: The `lr` argument is deprecated,  
use `learning_rate` instead.
```

```
super(SGD, self).__init__(name, **kwargs)
```

```
conv2d_534 (Conv2D) (None, 8, 8, 128) 16512  
['activation_396[0][0]']
```

```
conv2d_531 (Conv2D) (None, 8, 8, 128) 16512  
['max_pooling2d_23[0][0]']
```

```
add_126 (Add) (None, 8, 8, 128) 0  
['conv2d_534[0][0]',
```

```
'conv2d_531[0][0]']
```

```
batch_normalization_383 (Batch Normalization) (None, 8, 8, 128) 512  
['add_126[0][0]']
```

activation_397 (Activation)	(None, 8, 8, 128)	0
['batch_normalization_383[0][0]']		
conv2d_536 (Conv2D)	(None, 8, 8, 128)	16512
['activation_397[0][0]']		
batch_normalization_384 (Batch Normalization)	(None, 8, 8, 128)	512
['conv2d_536[0][0]']		
activation_398 (Activation)	(None, 8, 8, 128)	0
['batch_normalization_384[0][0]']		
conv2d_537 (Conv2D)	(None, 8, 8, 128)	147584
['activation_398[0][0]']		
batch_normalization_377 (Batch Normalization)	(None, 16, 16, 128)	512
['add_123[0][0]']		
batch_normalization_385 (Batch Normalization)	(None, 8, 8, 128)	512
['conv2d_537[0][0]']		
activation_391 (Activation)	(None, 16, 16, 128)	0
['batch_normalization_377[0][0]']		
activation_399 (Activation)	(None, 8, 8, 128)	0
['batch_normalization_385[0][0]']		
conv2d_528 (Conv2D)	(None, 16, 16, 128)	16512
['activation_391[0][0]']		

conv2d_538 (Conv2D) ['activation_399[0][0]']	(None, 8, 8, 128)	16512
conv2d_535 (Conv2D) ['add_126[0][0]']	(None, 8, 8, 128)	16512
batch_normalization_378 (Batch Normalization) ['conv2d_528[0][0]']	(None, 16, 16, 128)	512
add_127 (Add) ['conv2d_538[0][0]', 'conv2d_535[0][0]']	(None, 8, 8, 128)	0
activation_392 (Activation) ['batch_normalization_378[0][0]']	(None, 16, 16, 128)	0
up_sampling2d_16 (UpSampling2D) ['add_127[0][0]'] )	(None, 16, 16, 128)	0
conv2d_529 (Conv2D) ['activation_392[0][0]']	(None, 16, 16, 128)	147584
conv2d_539 (Conv2D) ['up_sampling2d_16[0][0]']	(None, 16, 16, 128)	16512
batch_normalization_379 (Batch Normalization) ['conv2d_529[0][0]']	(None, 16, 16, 128)	512
conv2d_540 (Conv2D) ['conv2d_539[0][0]']	(None, 16, 16, 128)	16512

activation_393 (Activation)	(None, 16, 16, 128)	0
['batch_normalization_379[0][0]']		
activation_400 (Activation)	(None, 16, 16, 128)	0
['conv2d_540[0][0]']		
conv2d_530 (Conv2D)	(None, 16, 16, 128)	16512
['activation_393[0][0]']		
conv2d_527 (Conv2D)	(None, 16, 16, 128)	16512
['add_123[0][0]']		
lambda_14 (Lambda)	(None, 16, 16, 128)	0
['activation_400[0][0]']		
add_125 (Add)	(None, 16, 16, 128)	0
['conv2d_530[0][0]',		
'conv2d_527[0][0]']		
multiply_14 (Multiply)	(None, 16, 16, 128)	0
['lambda_14[0][0]',		
'add_125[0][0]']		
batch_normalization_386 (Batch Normalization)	(None, 16, 16, 128)	512
['multiply_14[0][0]']		
activation_401 (Activation)	(None, 16, 16, 128)	0
['batch_normalization_386[0][0]']		
conv2d_542 (Conv2D)	(None, 16, 16, 128)	16512
['activation_401[0][0]']		
batch_normalization_387 (Batch Normalization)	(None, 16, 16, 128)	512

['conv2d\_542[0][0]']  
Normalization)

activation\_402 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_387[0][0]']

conv2d\_543 (Conv2D) (None, 16, 16, 128) 147584  
['activation\_402[0][0]']

batch\_normalization\_388 (Batch (None, 16, 16, 128) 512  
['conv2d\_543[0][0]']  
Normalization)

activation\_403 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_388[0][0]']

conv2d\_544 (Conv2D) (None, 16, 16, 128) 16512  
['activation\_403[0][0]']

conv2d\_541 (Conv2D) (None, 16, 16, 128) 16512  
['multiply\_14[0][0]']

add\_128 (Add) (None, 16, 16, 128) 0  
['conv2d\_544[0][0]',  
'conv2d\_541[0][0]']

batch\_normalization\_389 (Batch (None, 16, 16, 128) 512  
['add\_128[0][0]']  
Normalization)

activation\_404 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_389[0][0]']

conv2d\_546 (Conv2D) (None, 16, 16, 64) 8256



['activation\_404[0][0]']

batch\_normalization\_390 (Batch Normalization) (None, 16, 16, 64) 256  
['conv2d\_546[0][0]']

activation\_405 (Activation) (None, 16, 16, 64) 0  
['batch\_normalization\_390[0][0]']

conv2d\_547 (Conv2D) (None, 8, 8, 64) 36928  
['activation\_405[0][0]']

batch\_normalization\_391 (Batch Normalization) (None, 8, 8, 64) 256  
['conv2d\_547[0][0]']

activation\_406 (Activation) (None, 8, 8, 64) 0  
['batch\_normalization\_391[0][0]']

conv2d\_548 (Conv2D) (None, 8, 8, 256) 16640  
['activation\_406[0][0]']

conv2d\_545 (Conv2D) (None, 8, 8, 256) 33024  
['add\_128[0][0]']

add\_129 (Add) (None, 8, 8, 256) 0  
['conv2d\_548[0][0]',  
'conv2d\_545[0][0]']

batch\_normalization\_392 (Batch Normalization) (None, 8, 8, 256) 1024  
['add\_129[0][0]']

activation\_407 (Activation) (None, 8, 8, 256) 0

['batch\_normalization\_392[0][0]']

conv2d_550 (Conv2D)	(None, 8, 8, 256)	65792
['activation_407[0][0]']		

batch_normalization_393 (Batch Normalization)	(None, 8, 8, 256)	1024
['conv2d_550[0][0]']		

activation_408 (Activation)	(None, 8, 8, 256)	0
['batch_normalization_393[0][0]']		

conv2d_551 (Conv2D)	(None, 8, 8, 256)	590080
['activation_408[0][0]']		

batch_normalization_394 (Batch Normalization)	(None, 8, 8, 256)	1024
['conv2d_551[0][0]']		

activation_409 (Activation)	(None, 8, 8, 256)	0
['batch_normalization_394[0][0]']		

conv2d_552 (Conv2D)	(None, 8, 8, 256)	65792
['activation_409[0][0]']		

conv2d_549 (Conv2D)	(None, 8, 8, 256)	65792
['add_129[0][0]']		

add_130 (Add)	(None, 8, 8, 256)	0
['conv2d_552[0][0]',		
'conv2d_549[0][0]']		

max_pooling2d_24 (MaxPooling2D)	(None, 4, 4, 256)	0
['add_130[0][0]']		
)		

batch_normalization_401 (Batch Normalization)	(None, 4, 4, 256)	1024
['max_pooling2d_24[0][0]']		
activation_416 (Activation)	(None, 4, 4, 256)	0
['batch_normalization_401[0][0]']		
conv2d_562 (Conv2D)	(None, 4, 4, 256)	65792
['activation_416[0][0]']		
batch_normalization_402 (Batch Normalization)	(None, 4, 4, 256)	1024
['conv2d_562[0][0]']		
activation_417 (Activation)	(None, 4, 4, 256)	0
['batch_normalization_402[0][0]']		
conv2d_563 (Conv2D)	(None, 4, 4, 256)	590080
['activation_417[0][0]']		
batch_normalization_403 (Batch Normalization)	(None, 4, 4, 256)	1024
['conv2d_563[0][0]']		
activation_418 (Activation)	(None, 4, 4, 256)	0
['batch_normalization_403[0][0]']		
conv2d_564 (Conv2D)	(None, 4, 4, 256)	65792
['activation_418[0][0]']		
conv2d_561 (Conv2D)	(None, 4, 4, 256)	65792
['max_pooling2d_24[0][0]']		

add_133 (Add)	(None, 4, 4, 256)	0
['conv2d_564[0][0]',		
'conv2d_561[0][0]']		
batch_normalization_404 (Batch Normalization)	(None, 4, 4, 256)	1024
['add_133[0][0]']		
activation_419 (Activation)	(None, 4, 4, 256)	0
['batch_normalization_404[0][0]']		
conv2d_566 (Conv2D)	(None, 4, 4, 256)	65792
['activation_419[0][0]']		
batch_normalization_405 (Batch Normalization)	(None, 4, 4, 256)	1024
['conv2d_566[0][0]']		
activation_420 (Activation)	(None, 4, 4, 256)	0
['batch_normalization_405[0][0]']		
conv2d_567 (Conv2D)	(None, 4, 4, 256)	590080
['activation_420[0][0]']		
batch_normalization_398 (Batch Normalization)	(None, 8, 8, 256)	1024
['add_130[0][0]']		
batch_normalization_406 (Batch Normalization)	(None, 4, 4, 256)	1024
['conv2d_567[0][0]']		
activation_413 (Activation)	(None, 8, 8, 256)	0

['batch\_normalization\_398[0][0]']

activation\_421 (Activation) (None, 4, 4, 256) 0  
['batch\_normalization\_406[0][0]']

conv2d\_558 (Conv2D) (None, 8, 8, 256) 65792  
['activation\_413[0][0]']

conv2d\_568 (Conv2D) (None, 4, 4, 256) 65792  
['activation\_421[0][0]']

conv2d\_565 (Conv2D) (None, 4, 4, 256) 65792  
['add\_133[0][0]']

batch\_normalization\_399 (Batch Normalization) (None, 8, 8, 256) 1024  
['conv2d\_558[0][0]']

add\_134 (Add) (None, 4, 4, 256) 0  
['conv2d\_568[0][0]',  
'conv2d\_565[0][0]']

activation\_414 (Activation) (None, 8, 8, 256) 0  
['batch\_normalization\_399[0][0]']

up\_sampling2d\_17 (UpSampling2D) (None, 8, 8, 256) 0  
['add\_134[0][0]']  
)

conv2d\_559 (Conv2D) (None, 8, 8, 256) 590080  
['activation\_414[0][0]']

conv2d\_569 (Conv2D) (None, 8, 8, 256) 65792  
['up\_sampling2d\_17[0][0]']

batch_normalization_400 (Batch Normalization)	(None, 8, 8, 256)	1024
conv2d_570 (Conv2D)	(None, 8, 8, 256)	65792
activation_415 (Activation)	(None, 8, 8, 256)	0
activation_422 (Activation)	(None, 8, 8, 256)	0
conv2d_560 (Conv2D)	(None, 8, 8, 256)	65792
conv2d_557 (Conv2D)	(None, 8, 8, 256)	65792
lambda_15 (Lambda)	(None, 8, 8, 256)	0
add_132 (Add)	(None, 8, 8, 256)	0
multiply_15 (Multiply)	(None, 8, 8, 256)	0
batch_normalization_407 (Batch Normalization)	(None, 8, 8, 256)	1024

activation_423 (Activation) ['batch_normalization_407[0][0]']	(None, 8, 8, 256)	0
conv2d_572 (Conv2D) ['activation_423[0][0]']	(None, 8, 8, 256)	65792
batch_normalization_408 (Batch ['conv2d_572[0][0]'] Normalization)	(None, 8, 8, 256)	1024
activation_424 (Activation) ['batch_normalization_408[0][0]']	(None, 8, 8, 256)	0
conv2d_573 (Conv2D) ['activation_424[0][0]']	(None, 8, 8, 256)	590080
batch_normalization_409 (Batch ['conv2d_573[0][0]'] Normalization)	(None, 8, 8, 256)	1024
activation_425 (Activation) ['batch_normalization_409[0][0]']	(None, 8, 8, 256)	0
conv2d_574 (Conv2D) ['activation_425[0][0]']	(None, 8, 8, 256)	65792
conv2d_571 (Conv2D) ['multiply_15[0][0]']	(None, 8, 8, 256)	65792
add_135 (Add) ['conv2d_574[0][0]', 'conv2d_571[0][0]']	(None, 8, 8, 256)	0
batch_normalization_410 (Batch	(None, 8, 8, 256)	1024

['add\_135[0][0]']  
Normalization)

activation\_426 (Activation) (None, 8, 8, 256) 0  
['batch\_normalization\_410[0][0]']

conv2d\_576 (Conv2D) (None, 8, 8, 128) 32896  
['activation\_426[0][0]']

batch\_normalization\_411 (Batch (None, 8, 8, 128) 512  
['conv2d\_576[0][0]']  
Normalization)

activation\_427 (Activation) (None, 8, 8, 128) 0  
['batch\_normalization\_411[0][0]']

conv2d\_577 (Conv2D) (None, 8, 8, 128) 147584  
['activation\_427[0][0]']

batch\_normalization\_412 (Batch (None, 8, 8, 128) 512  
['conv2d\_577[0][0]']  
Normalization)

activation\_428 (Activation) (None, 8, 8, 128) 0  
['batch\_normalization\_412[0][0]']

conv2d\_578 (Conv2D) (None, 8, 8, 512) 66048  
['activation\_428[0][0]']

conv2d\_575 (Conv2D) (None, 8, 8, 512) 131584  
['add\_135[0][0]']

add\_136 (Add) (None, 8, 8, 512) 0  
['conv2d\_578[0][0]',



'conv2d\_575[0][0]'

batch\_normalization\_413 (Batch Normalization) (None, 8, 8, 512) 2048  
['add\_136[0][0]']

activation\_429 (Activation) (None, 8, 8, 512) 0  
['batch\_normalization\_413[0][0]']

conv2d\_580 (Conv2D) (None, 8, 8, 128) 65664  
['activation\_429[0][0]']

batch\_normalization\_414 (Batch Normalization) (None, 8, 8, 128) 512  
['conv2d\_580[0][0]']

activation\_430 (Activation) (None, 8, 8, 128) 0  
['batch\_normalization\_414[0][0]']

conv2d\_581 (Conv2D) (None, 8, 8, 128) 147584  
['activation\_430[0][0]']

batch\_normalization\_415 (Batch Normalization) (None, 8, 8, 128) 512  
['conv2d\_581[0][0]']

activation\_431 (Activation) (None, 8, 8, 128) 0  
['batch\_normalization\_415[0][0]']

conv2d\_582 (Conv2D) (None, 8, 8, 512) 66048  
['activation\_431[0][0]']

conv2d\_579 (Conv2D) (None, 8, 8, 512) 262656  
['add\_136[0][0]']

add_137 (Add)	(None, 8, 8, 512)	0
['conv2d_582[0][0]',		
'conv2d_579[0][0]']		
batch_normalization_416 (Batch Normalization)	(None, 8, 8, 512)	2048
['add_137[0][0]']		
activation_432 (Activation)	(None, 8, 8, 512)	0
['batch_normalization_416[0][0]']		
conv2d_584 (Conv2D)	(None, 8, 8, 128)	65664
['activation_432[0][0]']		
batch_normalization_417 (Batch Normalization)	(None, 8, 8, 128)	512
['conv2d_584[0][0]']		
activation_433 (Activation)	(None, 8, 8, 128)	0
['batch_normalization_417[0][0]']		
conv2d_585 (Conv2D)	(None, 8, 8, 128)	147584
['activation_433[0][0]']		
batch_normalization_418 (Batch Normalization)	(None, 8, 8, 128)	512
['conv2d_585[0][0]']		
activation_434 (Activation)	(None, 8, 8, 128)	0
['batch_normalization_418[0][0]']		
conv2d_586 (Conv2D)	(None, 8, 8, 512)	66048
['activation_434[0][0]']		

```
conv2d_583 (Conv2D)          (None, 8, 8, 512)    262656
['add_137[0][0]']
```

```
add_138 (Add)                (None, 8, 8, 512)    0
['conv2d_586[0][0]',
'conv2d_583[0][0]']
```

```
average_pooling2d_6 (AveragePo (None, 5, 5, 512)    0
['add_138[0][0]']
oling2D)
```

```
flatten_6 (Flatten)          (None, 12800)         0
['average_pooling2d_6[0][0]']
```

```
dense_6 (Dense)              (None, 24)            307224
['flatten_6[0][0]']
```

```
=====
Total params: 7,005,528
Trainable params: 6,989,144
Non-trainable params: 16,384
```

---

```
# training
model = training(model, '56mini-SGD-Base')
```

```
Learning rate: 0.0001
Epoch 1/10
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:82:
UserWarning: `Model.fit_generator` is deprecated and will be removed
in a future version. Please use `Model.fit`, which supports
generators.
```

```
10/10 [=====] - ETA: 0s - loss: 3.3288 -
accuracy: 0.1233WARNING:tensorflow:Learning rate reduction is
conditioned on metric `val_accuracy` which is not available. Available
metrics are: loss,accuracy,lr
WARNING:tensorflow:Early stopping conditioned on metric `val_accuracy`
which is not available. Available metrics are: loss,accuracy,lr
```

10/10 [=====] - 89s 8s/step - loss: 3.3288 -  
accuracy: 0.1233 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 2/10  
10/10 [=====] - ETA: 0s - loss: 2.5569 -  
accuracy: 0.2147WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 84s 8s/step - loss: 2.5569 -  
accuracy: 0.2147 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 3/10  
10/10 [=====] - ETA: 0s - loss: 2.2407 -  
accuracy: 0.2992WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 87s 9s/step - loss: 2.2407 -  
accuracy: 0.2992 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 4/10  
10/10 [=====] - ETA: 0s - loss: 2.1238 -  
accuracy: 0.3200WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 83s 9s/step - loss: 2.1238 -  
accuracy: 0.3200 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 5/10  
10/10 [=====] - ETA: 0s - loss: 1.9305 -  
accuracy: 0.3829WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 83s 8s/step - loss: 1.9305 -  
accuracy: 0.3829 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 6/10  
10/10 [=====] - ETA: 0s - loss: 1.8059 -  
accuracy: 0.3951WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr

```
10/10 [=====] - 83s 9s/step - loss: 1.8059 -  
accuracy: 0.3951 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 7/10  
10/10 [=====] - ETA: 0s - loss: 1.7566 -  
accuracy: 0.4057WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 83s 8s/step - loss: 1.7566 -  
accuracy: 0.4057 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 8/10  
10/10 [=====] - ETA: 0s - loss: 1.7023 -  
accuracy: 0.4269WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 83s 8s/step - loss: 1.7023 -  
accuracy: 0.4269 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 9/10  
10/10 [=====] - ETA: 0s - loss: 1.6334 -  
accuracy: 0.4506WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 83s 8s/step - loss: 1.6334 -  
accuracy: 0.4506 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 10/10  
10/10 [=====] - ETA: 0s - loss: 1.5503 -  
accuracy: 0.4808WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 83s 8s/step - loss: 1.5503 -  
accuracy: 0.4808 - lr: 1.0000e-04  
Time taken by above cell is 18.884944967428844.
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:88:  
UserWarning: `Model.evaluate_generator` is deprecated and will be  
removed in a future version. Please use `Model.evaluate`, which  
supports generators.  
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:89:  
UserWarning: `Model.evaluate_generator` is deprecated and will be
```

removed in a future version. Please use `Model.evaluate`, which supports generators.

```
3/3 [=====] - 6s 2s/step - loss: 2.5170 -  
accuracy: 0.2358  
validation loss: 2.445267677307129  
validation accuracy: 0.28333333134651184  
Test loss: 2.5169692039489746  
Test accuracy: 0.23577235639095306
```

## Second model

The second model consists of the following parameters:

- optimizer - Adam
- no dropout
- no regularization

```
# define model
```

```
model = AttentionResNet56_mini(shape=(32,32,3), in_channel=32,  
kernel_size=5, n_classes=24, dropout=None, regularization=None)
```

```
# define loss, metrics, optimizer
```

```
optimizer = Adam(lr = lr_schedule(0) )  
model.compile(optimizer, loss='categorical_crossentropy',  
metrics=['accuracy'])
```

Learning rate: 0.0001

```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adam.py:105:  
UserWarning: The `lr` argument is deprecated, use `learning_rate`  
instead.
```

```
super(Adam, self).__init__(name, **kwargs)
```

```
# training
```

```
model = training(model, '56mini-Adam-Base')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:82:  
UserWarning: `Model.fit_generator` is deprecated and will be removed  
in a future version. Please use `Model.fit`, which supports  
generators.
```

Learning rate: 0.0001

Epoch 1/10

```
10/10 [=====] - ETA: 0s - loss: 3.8085 -  
accuracy: 0.1918WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 92s 8s/step - loss: 3.8085 -  
accuracy: 0.1918 - lr: 1.0000e-04
```

Learning rate: 0.0001

Epoch 2/10

10/10 [=====] - ETA: 0s - loss: 2.0231 -

accuracy: 0.3829WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr

WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr

10/10 [=====] - 83s 8s/step - loss: 2.0231 -  
accuracy: 0.3829 - lr: 1.0000e-04

Learning rate: 0.0001

Epoch 3/10

10/10 [=====] - ETA: 0s - loss: 1.5241 -

accuracy: 0.4988WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr

WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr

10/10 [=====] - 83s 9s/step - loss: 1.5241 -  
accuracy: 0.4988 - lr: 1.0000e-04

Learning rate: 0.0001

Epoch 4/10

10/10 [=====] - ETA: 0s - loss: 1.2348 -

accuracy: 0.5878WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr

WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr

10/10 [=====] - 83s 8s/step - loss: 1.2348 -  
accuracy: 0.5878 - lr: 1.0000e-04

Learning rate: 0.0001

Epoch 5/10

10/10 [=====] - ETA: 0s - loss: 1.0614 -

accuracy: 0.6147WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr

WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr

10/10 [=====] - 84s 8s/step - loss: 1.0614 -  
accuracy: 0.6147 - lr: 1.0000e-04

Learning rate: 0.0001

Epoch 6/10

10/10 [=====] - ETA: 0s - loss: 0.9178 -

accuracy: 0.6571WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr

WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr

10/10 [=====] - 83s 8s/step - loss: 0.9178 -  
accuracy: 0.6571 - lr: 1.0000e-04

Learning rate: 0.0001  
Epoch 7/10  
10/10 [=====] - ETA: 0s - loss: 0.8826 -  
accuracy: 0.6969WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 87s 9s/step - loss: 0.8826 -  
accuracy: 0.6969 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 8/10  
10/10 [=====] - ETA: 0s - loss: 0.7821 -  
accuracy: 0.7306WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 83s 8s/step - loss: 0.7821 -  
accuracy: 0.7306 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 9/10  
10/10 [=====] - ETA: 0s - loss: 0.8114 -  
accuracy: 0.7004WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 83s 8s/step - loss: 0.8114 -  
accuracy: 0.7004 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 10/10  
10/10 [=====] - ETA: 0s - loss: 0.7012 -  
accuracy: 0.7396WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 83s 9s/step - loss: 0.7012 -  
accuracy: 0.7396 - lr: 1.0000e-04  
Time taken by above cell is 19.88717257976532.

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:88:  
UserWarning: `Model.evaluate\_generator` is deprecated and will be  
removed in a future version. Please use `Model.evaluate`, which  
supports generators.  
/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:89:  
UserWarning: `Model.evaluate\_generator` is deprecated and will be  
removed in a future version. Please use `Model.evaluate`, which  
supports generators.



```

3/3 [=====] - 6s 2s/step - loss: 3.1991 -
accuracy: 0.1301
validation loss: 3.351363182067871
validation accuracy: 0.0833333358168602
Test loss: 3.199070930480957
Test accuracy: 0.13008129596710205

```

It seems that SGD optimizer gives better results of training. However, during several experiments we decided that in further computations Adam optimizer will be more suitable.

### Including dropout and regularization

The third model was created with the following parameters:

- optimizer - Adam
- regularization = 0.01
- dropout = 0.4

```
def AttentionResNet56_mini(shape, in_channel, kernel_size, n_classes,
dropout=None, regularization=0.01):
```

```

    """
    :param shape: The tuple of input data.
    :param in_channel: The 4-th dimension (channel number) of input
weight matrix. For example, in_channel=3 means the input contains 3
channels.
    :param kernel_size: Integer. the shape of the kernel. For example,
default kernel_size = 3 means you have a 3*3 kernel.
    :param n_classes: Integer. The number of target classes. For
example, n_classes = 10 means you have 10 class labels.
    :param dropout: Float between 0 and 1. Fraction of the input units
to drop.
    :param regularization: Float. Fraction of the input units to drop.
    """

    input_data = Input(shape=shape) # 32x32x32
    x = Conv2D(in_channel, kernel_size=kernel_size, padding='same')
(input_data) # 32x32x32
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = MaxPooling2D(pool_size=2, padding='same')(x) # 16x16x32

    out_channel = in_channel * 4
    x = Residual_Unit(x, in_channel, out_channel) # 16x16x128
    x = Attention_Block(x, skip=1)

    in_channel = out_channel // 2
    out_channel = in_channel * 4
    x = Residual_Unit(x, in_channel, out_channel, stride=2) # 8x8x256
    x = Attention_Block(x, skip=1)

```

```

    in_channel = out_channel // 2
    out_channel = in_channel * 4
    x = Residual_Unit(x, in_channel, out_channel, stride=1) #
4x4x1024
    x = Residual_Unit(x, in_channel, out_channel)
    x = Residual_Unit(x, in_channel, out_channel)

    # add BN and Activation
    x = BatchNormalization()(x) # new
    x = Activation('relu')(x) # new
    x = AveragePooling2D(pool_size=4, strides=1)(x) # 1x1x1024
    x = Flatten()(x)

    if dropout:
        x = Dropout(dropout)(x) # new
        output = Dense(n_classes, kernel_regularizer=l2(regularization),
activation='softmax')(x) # new
        model = Model(input_data, output)

    return model

```

```

# define model
model = AttentionResNet56_mini(shape=(32,32,3), in_channel=32,
kernel_size=5, n_classes=24, dropout=0.4, regularization=0.01)

```

```

# define loss, metrics, optimizer
optimizer = Adam(lr = lr_schedule(0) )
model.compile(optimizer, loss='categorical_crossentropy',
metrics=['accuracy'])
model.summary()

```

Learning rate: 0.0001  
Model: "model\_8"

Layer (type) Connected to	Output Shape	Param #
=====		
input_9 (InputLayer)	[(None, 32, 32, 3)]	0
=====		
conv2d_660 (Conv2D) ['input_9[0][0]']	(None, 32, 32, 32)	2432
=====		
batch_normalization_471 (Batch Normalization) ['conv2d_660[0][0]']	(None, 32, 32, 32)	128

Normalization)

activation\_489 (Activation) (None, 32, 32, 32) 0  
['batch\_normalization\_471[0][0]']

max\_pooling2d\_28 (MaxPooling2D) (None, 16, 16, 32) 0  
['activation\_489[0][0]']  
)

batch\_normalization\_472 (Batch Normalization) (None, 16, 16, 32) 128  
['max\_pooling2d\_28[0][0]']

activation\_490 (Activation) (None, 16, 16, 32) 0  
['batch\_normalization\_472[0][0]']

conv2d\_662 (Conv2D) (None, 16, 16, 32) 1056  
['activation\_490[0][0]']

batch\_normalization\_473 (Batch Normalization) (None, 16, 16, 32) 128  
['conv2d\_662[0][0]']

activation\_491 (Activation) (None, 16, 16, 32) 0  
['batch\_normalization\_473[0][0]']

conv2d\_663 (Conv2D) (None, 16, 16, 32) 9248  
['activation\_491[0][0]']

batch\_normalization\_474 (Batch Normalization) (None, 16, 16, 32) 128  
['conv2d\_663[0][0]']

```

activation_492 (Activation)      (None, 16, 16, 32)    0
['batch_normalization_474[0][0]']

conv2d_664 (Conv2D)              (None, 16, 16, 128)  4224
['activation_492[0][0]']

conv2d_661 (Conv2D)              (None, 16, 16, 128)  4224
['max_pooling2d_28[0][0]']

add_156 (Add)                    (None, 16, 16, 128)  0
['conv2d_664[0][0]',
'conv2d_661[0][0]']

batch_normalization_475 (Batch Normalization) (None, 16, 16, 128)  512
['add_156[0][0]']

activation_493 (Activation)      (None, 16, 16, 128)  0
['batch_normalization_475[0][0]']

conv2d_666 (Conv2D)              (None, 16, 16, 128)  16512
['activation_493[0][0]']

batch_normalization_476 (Batch Normalization) (None, 16, 16, 128)  512
['conv2d_666[0][0]']

activation_494 (Activation)      (None, 16, 16, 128)  0
['batch_normalization_476[0][0]']

conv2d_667 (Conv2D)              (None, 16, 16, 128)  147584
['activation_494[0][0]']

batch_normalization_477 (Batch Normalization) (None, 16, 16, 128)  512
['conv2d_667[0][0]']

```

Normalization)

activation\_495 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_477[0][0]']

conv2d\_668 (Conv2D) (None, 16, 16, 128) 16512  
['activation\_495[0][0]']

conv2d\_665 (Conv2D) (None, 16, 16, 128) 16512  
['add\_156[0][0]']

add\_157 (Add) (None, 16, 16, 128) 0  
['conv2d\_668[0][0]',  
'conv2d\_665[0][0]']

max\_pooling2d\_29 (MaxPooling2D) (None, 8, 8, 128) 0  
['add\_157[0][0]']  
)

batch\_normalization\_484 (Batch Normalization) (None, 8, 8, 128) 512  
['max\_pooling2d\_29[0][0]']

activation\_502 (Activation) (None, 8, 8, 128) 0  
['batch\_normalization\_484[0][0]']

conv2d\_678 (Conv2D) (None, 8, 8, 128) 16512  
['activation\_502[0][0]']

batch\_normalization\_485 (Batch Normalization) (None, 8, 8, 128) 512  
['conv2d\_678[0][0]']

```
activation_503 (Activation)      (None, 8, 8, 128)      0
['batch_normalization_485[0][0]']
```

```
conv2d_679 (Conv2D)             (None, 8, 8, 128)      147584
['activation_503[0][0]']
```

```
batch_normalization_486 (Batch Normalization) (None, 8, 8, 128)      512
['conv2d_679[0][0]']
```

```
activation_504 (Activation)      (None, 8, 8, 128)      0
['batch_normalization_486[0][0]']
```

```
conv2d_680 (Conv2D)             (None, 8, 8, 128)      16512
['activation_504[0][0]']
```

```
conv2d_677 (Conv2D)             (None, 8, 8, 128)      16512
['max_pooling2d_29[0][0]']
```

```
add_160 (Add)                   (None, 8, 8, 128)      0
['conv2d_680[0][0]',
'conv2d_677[0][0]']
```

```
batch_normalization_487 (Batch Normalization) (None, 8, 8, 128)      512
['add_160[0][0]']
```

```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adam.py:105:
UserWarning: The `lr` argument is deprecated, use `learning_rate`
instead.
```

```
    super(Adam, self).__init__(name, **kwargs)
```

```
activation_505 (Activation)      (None, 8, 8, 128)      0
['batch_normalization_487[0][0]']
```

```
conv2d_682 (Conv2D)             (None, 8, 8, 128)      16512
['activation_505[0][0]']
```

batch\_normalization\_488 (Batch Normalization) (None, 8, 8, 128) 512  
['conv2d\_682[0][0]']

activation\_506 (Activation) (None, 8, 8, 128) 0  
['batch\_normalization\_488[0][0]']

conv2d\_683 (Conv2D) (None, 8, 8, 128) 147584  
['activation\_506[0][0]']

batch\_normalization\_481 (Batch Normalization) (None, 16, 16, 128) 512  
['add\_157[0][0]']

batch\_normalization\_489 (Batch Normalization) (None, 8, 8, 128) 512  
['conv2d\_683[0][0]']

activation\_499 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_481[0][0]']

activation\_507 (Activation) (None, 8, 8, 128) 0  
['batch\_normalization\_489[0][0]']

conv2d\_674 (Conv2D) (None, 16, 16, 128) 16512  
['activation\_499[0][0]']

conv2d\_684 (Conv2D) (None, 8, 8, 128) 16512  
['activation\_507[0][0]']

conv2d\_681 (Conv2D) (None, 8, 8, 128) 16512  
['add\_160[0][0]']

batch\_normalization\_482 (Batch Normalization) (None, 16, 16, 128) 512  
['conv2d\_674[0][0]']

add\_161 (Add) (None, 8, 8, 128) 0  
['conv2d\_684[0][0]',  
'conv2d\_681[0][0]']

activation\_500 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_482[0][0]']

up\_sampling2d\_20 (UpSampling2D) (None, 16, 16, 128) 0  
['add\_161[0][0]']  
)

conv2d\_675 (Conv2D) (None, 16, 16, 128) 147584  
['activation\_500[0][0]']

conv2d\_685 (Conv2D) (None, 16, 16, 128) 16512  
['up\_sampling2d\_20[0][0]']

batch\_normalization\_483 (Batch Normalization) (None, 16, 16, 128) 512  
['conv2d\_675[0][0]']

conv2d\_686 (Conv2D) (None, 16, 16, 128) 16512  
['conv2d\_685[0][0]']

activation\_501 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_483[0][0]']

activation\_508 (Activation) (None, 16, 16, 128) 0  
['conv2d\_686[0][0]']



conv2d_676 (Conv2D) ['activation_501[0][0]']	(None, 16, 16, 128)	16512
conv2d_673 (Conv2D) ['add_157[0][0]']	(None, 16, 16, 128)	16512
lambda_18 (Lambda) ['activation_508[0][0]']	(None, 16, 16, 128)	0
add_159 (Add) ['conv2d_676[0][0]', 'conv2d_673[0][0]']	(None, 16, 16, 128)	0
multiply_18 (Multiply) ['lambda_18[0][0]', 'add_159[0][0]']	(None, 16, 16, 128)	0
batch_normalization_490 (Batch Normalization) ['multiply_18[0][0]']	(None, 16, 16, 128)	512
activation_509 (Activation) ['batch_normalization_490[0][0]']	(None, 16, 16, 128)	0
conv2d_688 (Conv2D) ['activation_509[0][0]']	(None, 16, 16, 128)	16512
batch_normalization_491 (Batch Normalization) ['conv2d_688[0][0]']	(None, 16, 16, 128)	512
activation_510 (Activation) ['batch_normalization_491[0][0]']	(None, 16, 16, 128)	0

conv2d\_689 (Conv2D) (None, 16, 16, 128) 147584  
['activation\_510[0][0]']

batch\_normalization\_492 (Batch Normalization) (None, 16, 16, 128) 512  
['conv2d\_689[0][0]']

activation\_511 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_492[0][0]']

conv2d\_690 (Conv2D) (None, 16, 16, 128) 16512  
['activation\_511[0][0]']

conv2d\_687 (Conv2D) (None, 16, 16, 128) 16512  
['multiply\_18[0][0]']

add\_162 (Add) (None, 16, 16, 128) 0  
['conv2d\_690[0][0]',  
'conv2d\_687[0][0]']

batch\_normalization\_493 (Batch Normalization) (None, 16, 16, 128) 512  
['add\_162[0][0]']

activation\_512 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_493[0][0]']

conv2d\_692 (Conv2D) (None, 16, 16, 64) 8256  
['activation\_512[0][0]']

batch\_normalization\_494 (Batch Normalization) (None, 16, 16, 64) 256  
['conv2d\_692[0][0]']

activation_513 (Activation)	(None, 16, 16, 64)	0
['batch_normalization_494[0][0]']		
conv2d_693 (Conv2D)	(None, 8, 8, 64)	36928
['activation_513[0][0]']		
batch_normalization_495 (Batch Normalization)	(None, 8, 8, 64)	256
['conv2d_693[0][0]']		
activation_514 (Activation)	(None, 8, 8, 64)	0
['batch_normalization_495[0][0]']		
conv2d_694 (Conv2D)	(None, 8, 8, 256)	16640
['activation_514[0][0]']		
conv2d_691 (Conv2D)	(None, 8, 8, 256)	33024
['add_162[0][0]']		
add_163 (Add)	(None, 8, 8, 256)	0
['conv2d_694[0][0]',		
'conv2d_691[0][0]']		
batch_normalization_496 (Batch Normalization)	(None, 8, 8, 256)	1024
['add_163[0][0]']		
activation_515 (Activation)	(None, 8, 8, 256)	0
['batch_normalization_496[0][0]']		
conv2d_696 (Conv2D)	(None, 8, 8, 256)	65792
['activation_515[0][0]']		
batch_normalization_497 (Batch Normalization)	(None, 8, 8, 256)	1024
['conv2d_696[0][0]']		

Normalization)

activation_516 (Activation)	(None, 8, 8, 256)	0
['batch_normalization_497[0][0]']		
conv2d_697 (Conv2D)	(None, 8, 8, 256)	590080
['activation_516[0][0]']		
batch_normalization_498 (Batch Normalization)	(None, 8, 8, 256)	1024
['conv2d_697[0][0]']		
activation_517 (Activation)	(None, 8, 8, 256)	0
['batch_normalization_498[0][0]']		
conv2d_698 (Conv2D)	(None, 8, 8, 256)	65792
['activation_517[0][0]']		
conv2d_695 (Conv2D)	(None, 8, 8, 256)	65792
['add_163[0][0]']		
add_164 (Add)	(None, 8, 8, 256)	0
['conv2d_698[0][0]',		
'conv2d_695[0][0]']		
max_pooling2d_30 (MaxPooling2D)	(None, 4, 4, 256)	0
['add_164[0][0]']		
)		
batch_normalization_505 (Batch Normalization)	(None, 4, 4, 256)	1024
['max_pooling2d_30[0][0]']		

activation_524 (Activation) ['batch_normalization_505[0][0]']	(None, 4, 4, 256)	0
conv2d_708 (Conv2D) ['activation_524[0][0]']	(None, 4, 4, 256)	65792
batch_normalization_506 (Batch ['conv2d_708[0][0]'] Normalization)	(None, 4, 4, 256)	1024
activation_525 (Activation) ['batch_normalization_506[0][0]']	(None, 4, 4, 256)	0
conv2d_709 (Conv2D) ['activation_525[0][0]']	(None, 4, 4, 256)	590080
batch_normalization_507 (Batch ['conv2d_709[0][0]'] Normalization)	(None, 4, 4, 256)	1024
activation_526 (Activation) ['batch_normalization_507[0][0]']	(None, 4, 4, 256)	0
conv2d_710 (Conv2D) ['activation_526[0][0]']	(None, 4, 4, 256)	65792
conv2d_707 (Conv2D) ['max_pooling2d_30[0][0]']	(None, 4, 4, 256)	65792
add_167 (Add) ['conv2d_710[0][0]', 'conv2d_707[0][0]']	(None, 4, 4, 256)	0
batch_normalization_508 (Batch ['add_167[0][0]']	(None, 4, 4, 256)	1024

Normalization)

activation_527 (Activation)	(None, 4, 4, 256)	0
['batch_normalization_508[0][0]']		
conv2d_712 (Conv2D)	(None, 4, 4, 256)	65792
['activation_527[0][0]']		
batch_normalization_509 (Batch Normalization)	(None, 4, 4, 256)	1024
['conv2d_712[0][0]']		
activation_528 (Activation)	(None, 4, 4, 256)	0
['batch_normalization_509[0][0]']		
conv2d_713 (Conv2D)	(None, 4, 4, 256)	590080
['activation_528[0][0]']		
batch_normalization_502 (Batch Normalization)	(None, 8, 8, 256)	1024
['add_164[0][0]']		
batch_normalization_510 (Batch Normalization)	(None, 4, 4, 256)	1024
['conv2d_713[0][0]']		
activation_521 (Activation)	(None, 8, 8, 256)	0
['batch_normalization_502[0][0]']		
activation_529 (Activation)	(None, 4, 4, 256)	0
['batch_normalization_510[0][0]']		
conv2d_704 (Conv2D)	(None, 8, 8, 256)	65792
['activation_521[0][0]']		

conv2d_714 (Conv2D) ['activation_529[0][0]']	(None, 4, 4, 256)	65792
conv2d_711 (Conv2D) ['add_167[0][0]']	(None, 4, 4, 256)	65792
batch_normalization_503 (Batch Normalization) ['conv2d_704[0][0]']	(None, 8, 8, 256)	1024
add_168 (Add) ['conv2d_714[0][0]', 'conv2d_711[0][0]']	(None, 4, 4, 256)	0
activation_522 (Activation) ['batch_normalization_503[0][0]']	(None, 8, 8, 256)	0
up_sampling2d_21 (UpSampling2D) ['add_168[0][0]'] )	(None, 8, 8, 256)	0
conv2d_705 (Conv2D) ['activation_522[0][0]']	(None, 8, 8, 256)	590080
conv2d_715 (Conv2D) ['up_sampling2d_21[0][0]']	(None, 8, 8, 256)	65792
batch_normalization_504 (Batch Normalization) ['conv2d_705[0][0]']	(None, 8, 8, 256)	1024
conv2d_716 (Conv2D) ['conv2d_715[0][0]']	(None, 8, 8, 256)	65792

activation_523 (Activation) ['batch_normalization_504[0][0]']	(None, 8, 8, 256)	0
activation_530 (Activation) ['conv2d_716[0][0]']	(None, 8, 8, 256)	0
conv2d_706 (Conv2D) ['activation_523[0][0]']	(None, 8, 8, 256)	65792
conv2d_703 (Conv2D) ['add_164[0][0]']	(None, 8, 8, 256)	65792
lambda_19 (Lambda) ['activation_530[0][0]']	(None, 8, 8, 256)	0
add_166 (Add) ['conv2d_706[0][0]', 'conv2d_703[0][0]']	(None, 8, 8, 256)	0
multiply_19 (Multiply) ['lambda_19[0][0]', 'add_166[0][0]']	(None, 8, 8, 256)	0
batch_normalization_511 (Batch Normalization) ['multiply_19[0][0]']	(None, 8, 8, 256)	1024
activation_531 (Activation) ['batch_normalization_511[0][0]']	(None, 8, 8, 256)	0
conv2d_718 (Conv2D) ['activation_531[0][0]']	(None, 8, 8, 256)	65792



batch_normalization_512 (Batch Normalization)	(None, 8, 8, 256)	1024
['conv2d_718[0][0]']		
activation_532 (Activation)	(None, 8, 8, 256)	0
['batch_normalization_512[0][0]']		
conv2d_719 (Conv2D)	(None, 8, 8, 256)	590080
['activation_532[0][0]']		
batch_normalization_513 (Batch Normalization)	(None, 8, 8, 256)	1024
['conv2d_719[0][0]']		
activation_533 (Activation)	(None, 8, 8, 256)	0
['batch_normalization_513[0][0]']		
conv2d_720 (Conv2D)	(None, 8, 8, 256)	65792
['activation_533[0][0]']		
conv2d_717 (Conv2D)	(None, 8, 8, 256)	65792
['multiply_19[0][0]']		
add_169 (Add)	(None, 8, 8, 256)	0
['conv2d_720[0][0]',		
'conv2d_717[0][0]']		
batch_normalization_514 (Batch Normalization)	(None, 8, 8, 256)	1024
['add_169[0][0]']		
activation_534 (Activation)	(None, 8, 8, 256)	0
['batch_normalization_514[0][0]']		

conv2d_722 (Conv2D) ['activation_534[0][0]']	(None, 8, 8, 128)	32896
batch_normalization_515 (Batch Normalization) ['conv2d_722[0][0]']	(None, 8, 8, 128)	512
activation_535 (Activation) ['batch_normalization_515[0][0]']	(None, 8, 8, 128)	0
conv2d_723 (Conv2D) ['activation_535[0][0]']	(None, 8, 8, 128)	147584
batch_normalization_516 (Batch Normalization) ['conv2d_723[0][0]']	(None, 8, 8, 128)	512
activation_536 (Activation) ['batch_normalization_516[0][0]']	(None, 8, 8, 128)	0
conv2d_724 (Conv2D) ['activation_536[0][0]']	(None, 8, 8, 512)	66048
conv2d_721 (Conv2D) ['add_169[0][0]']	(None, 8, 8, 512)	131584
add_170 (Add) ['conv2d_724[0][0]', 'conv2d_721[0][0]']	(None, 8, 8, 512)	0
batch_normalization_517 (Batch Normalization) ['add_170[0][0]']	(None, 8, 8, 512)	2048

activation_537 (Activation)	(None, 8, 8, 512)	0
['batch_normalization_517[0][0]']		
conv2d_726 (Conv2D)	(None, 8, 8, 128)	65664
['activation_537[0][0]']		
batch_normalization_518 (Batch Normalization)	(None, 8, 8, 128)	512
['conv2d_726[0][0]']		
activation_538 (Activation)	(None, 8, 8, 128)	0
['batch_normalization_518[0][0]']		
conv2d_727 (Conv2D)	(None, 8, 8, 128)	147584
['activation_538[0][0]']		
batch_normalization_519 (Batch Normalization)	(None, 8, 8, 128)	512
['conv2d_727[0][0]']		
activation_539 (Activation)	(None, 8, 8, 128)	0
['batch_normalization_519[0][0]']		
conv2d_728 (Conv2D)	(None, 8, 8, 512)	66048
['activation_539[0][0]']		
conv2d_725 (Conv2D)	(None, 8, 8, 512)	262656
['add_170[0][0]']		
add_171 (Add)	(None, 8, 8, 512)	0
['conv2d_728[0][0]',		
'conv2d_725[0][0]']		
batch_normalization_520 (Batch Normalization)	(None, 8, 8, 512)	2048
['add_171[0][0]']		

Normalization)

activation_540 (Activation)	(None, 8, 8, 512)	0
['batch_normalization_520[0][0]']		
conv2d_730 (Conv2D)	(None, 8, 8, 128)	65664
['activation_540[0][0]']		
batch_normalization_521 (Batch Normalization)	(None, 8, 8, 128)	512
['conv2d_730[0][0]']		
activation_541 (Activation)	(None, 8, 8, 128)	0
['batch_normalization_521[0][0]']		
conv2d_731 (Conv2D)	(None, 8, 8, 128)	147584
['activation_541[0][0]']		
batch_normalization_522 (Batch Normalization)	(None, 8, 8, 128)	512
['conv2d_731[0][0]']		
activation_542 (Activation)	(None, 8, 8, 128)	0
['batch_normalization_522[0][0]']		
conv2d_732 (Conv2D)	(None, 8, 8, 512)	66048
['activation_542[0][0]']		
conv2d_729 (Conv2D)	(None, 8, 8, 512)	262656
['add_171[0][0]']		
add_172 (Add)	(None, 8, 8, 512)	0
['conv2d_732[0][0]',		
'conv2d_729[0][0]']		

```
batch_normalization_523 (Batch Normalization) (None, 8, 8, 512) 2048
['add_172[0][0]']
```

```
activation_543 (Activation) (None, 8, 8, 512) 0
['batch_normalization_523[0][0]']
```

```
average_pooling2d_8 (AveragePooling2D) (None, 5, 5, 512) 0
['activation_543[0][0]']
```

```
flatten_8 (Flatten) (None, 12800) 0
['average_pooling2d_8[0][0]']
```

```
dropout_2 (Dropout) (None, 12800) 0
['flatten_8[0][0]']
```

```
dense_8 (Dense) (None, 24) 307224
['dropout_2[0][0]']
```

```
=====
Total params: 7,007,576
Trainable params: 6,990,168
Non-trainable params: 17,408
=====
```

---

```
# training
```

```
model = training(model, '56mini-regularization')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:82:
UserWarning: `Model.fit_generator` is deprecated and will be removed
in a future version. Please use `Model.fit`, which supports
generators.
```

```
Learning rate: 0.0001
```

```
Epoch 1/10
```

```
10/10 [=====] - ETA: 0s - loss: 3.1317 -
```

accuracy: 0.2000WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 92s 8s/step - loss: 3.1317 -  
accuracy: 0.2000 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 2/10  
10/10 [=====] - ETA: 0s - loss: 2.5366 -  
accuracy: 0.3453WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 84s 9s/step - loss: 2.5366 -  
accuracy: 0.3453 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 3/10  
10/10 [=====] - ETA: 0s - loss: 2.2207 -  
accuracy: 0.3853WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 84s 8s/step - loss: 2.2207 -  
accuracy: 0.3853 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 4/10  
10/10 [=====] - ETA: 0s - loss: 2.0464 -  
accuracy: 0.4392WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 84s 8s/step - loss: 2.0464 -  
accuracy: 0.4392 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 5/10  
10/10 [=====] - ETA: 0s - loss: 1.8332 -  
accuracy: 0.5314WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 84s 8s/step - loss: 1.8332 -  
accuracy: 0.5314 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 6/10  
10/10 [=====] - ETA: 0s - loss: 1.7484 -

accuracy: 0.5576WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 84s 8s/step - loss: 1.7484 -  
accuracy: 0.5576 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 7/10  
10/10 [=====] - ETA: 0s - loss: 1.5822 -  
accuracy: 0.6049WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 84s 8s/step - loss: 1.5822 -  
accuracy: 0.6049 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 8/10  
10/10 [=====] - ETA: 0s - loss: 1.4687 -  
accuracy: 0.6392WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 84s 8s/step - loss: 1.4687 -  
accuracy: 0.6392 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 9/10  
10/10 [=====] - ETA: 0s - loss: 1.3706 -  
accuracy: 0.6735WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 84s 9s/step - loss: 1.3706 -  
accuracy: 0.6735 - lr: 1.0000e-04  
Learning rate: 0.0001  
Epoch 10/10  
10/10 [=====] - ETA: 0s - loss: 1.2912 -  
accuracy: 0.6898WARNING:tensorflow:Learning rate reduction is  
conditioned on metric `val\_accuracy` which is not available. Available  
metrics are: loss,accuracy,lr  
WARNING:tensorflow:Early stopping conditioned on metric `val\_accuracy`  
which is not available. Available metrics are: loss,accuracy,lr  
10/10 [=====] - 84s 8s/step - loss: 1.2912 -  
accuracy: 0.6898 - lr: 1.0000e-04  
Time taken by above cell is 21.86872197786967.

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:88:
UserWarning: `Model.evaluate_generator` is deprecated and will be
removed in a future version. Please use `Model.evaluate`, which
supports generators.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:89:
UserWarning: `Model.evaluate_generator` is deprecated and will be
removed in a future version. Please use `Model.evaluate`, which
supports generators.
```

```
3/3 [=====] - 6s 2s/step - loss: 3.6200 -
accuracy: 0.0840
validation loss: 3.573878288269043
validation accuracy: 0.125
Test loss: 3.620004415512085
Test accuracy: 0.08401083946228027
```

It seems that in this configuration test accuracy is very low.

### Including different batch sizes and increasing number of epochs

The fourth model was created using the following parameters:

- optimizer - Adam
- regularization = 0.001
- dropout = 0.4
- number of epochs = 15
- batch size - 64

```
def AttentionResNet56_mini(shape, in_channel, kernel_size, n_classes,
dropout=None, regularization=0.01):
```

```
    """
    :param shape: The tuple of input data.
    :param in_channel: The 4-th dimension (channel number) of input
weight matrix. For example, in_channel=3 means the input contains 3
channels.
    :param kernel_size: Integer. the shape of the kernel. For example,
default kernel_size = 3 means you have a 3*3 kernel.
    :param n_classes: Integer. The number of target classes. For
example, n_classes = 10 means you have 10 class labels.
    :param dropout: Float between 0 and 1. Fraction of the input units
to drop.
    :param regularization: Float. Fraction of the input units to drop.
    """

    input_data = Input(shape=shape) # 32x32x32
    x = Conv2D(in_channel, kernel_size=kernel_size, padding='same')
(input_data) # 32x32x32
    x = MaxPooling2D(pool_size=2, padding='same')(x) # 16x16x32

    out_channel = in_channel * 4
```



```

x = Residual_Unit(x, in_channel, out_channel) # 16x16x128
x = Attention_Block(x, skip=2)

in_channel = out_channel // 2
out_channel = in_channel * 4
x = Residual_Unit(x, in_channel, out_channel, stride=2) # 8x8x256
x = Attention_Block(x, skip=1)
x = Attention_Block(x, skip=1)

in_channel = out_channel // 2
out_channel = in_channel * 4
x = Residual_Unit(x, in_channel, out_channel, stride=1) #
4x4x1024
x = Residual_Unit(x, in_channel, out_channel)
x = Residual_Unit(x, in_channel, out_channel)

# add BN and Activation
x = BatchNormalization()(x) # new
x = Activation('relu')(x) # new
x = AveragePooling2D(pool_size=4, strides=1)(x) # 1x1x1024
x = Flatten()(x)

if dropout:
    x = Dropout(dropout)(x) # new
    output = Dense(n_classes, kernel_regularizer=l2(regularization),
activation='softmax')(x) # new
    model = Model(input_data, output)

return model

# define model
model = AttentionResNet56_mini(shape=(32,32,3), in_channel=32,
kernel_size=5, n_classes=24, dropout=0.4, regularization=0.001)

# define loss, metrics, optimizer
optimizer = Adam(lr = lr_schedule(0) )
model.compile(optimizer, loss='categorical_crossentropy',
metrics=['accuracy'])
model.summary()

Learning rate: 0.0001
Model: "model_10"

```

---

Layer (type) Connected to	Output Shape	Param #
input_11 (InputLayer)	[(None, 32, 32, 3)]	0

---

conv2d\_844 (Conv2D) (None, 32, 32, 32) 2432  
['input\_11[0][0]']

max\_pooling2d\_36 (MaxPooling2D) (None, 16, 16, 32) 0  
['conv2d\_844[0][0]']  
)

batch\_normalization\_603 (Batch Normalization) (None, 16, 16, 32) 128  
['max\_pooling2d\_36[0][0]']

activation\_626 (Activation) (None, 16, 16, 32) 0  
['batch\_normalization\_603[0][0]']

conv2d\_846 (Conv2D) (None, 16, 16, 32) 1056  
['activation\_626[0][0]']

batch\_normalization\_604 (Batch Normalization) (None, 16, 16, 32) 128  
['conv2d\_846[0][0]']

activation\_627 (Activation) (None, 16, 16, 32) 0  
['batch\_normalization\_604[0][0]']

conv2d\_847 (Conv2D) (None, 16, 16, 32) 9248  
['activation\_627[0][0]']

batch\_normalization\_605 (Batch Normalization) (None, 16, 16, 32) 128  
['conv2d\_847[0][0]']

activation\_628 (Activation) (None, 16, 16, 32) 0  
['batch\_normalization\_605[0][0]']

conv2d\_848 (Conv2D) (None, 16, 16, 128) 4224  
['activation\_628[0][0]']

conv2d\_845 (Conv2D) (None, 16, 16, 128) 4224  
['max\_pooling2d\_36[0][0]']

add\_200 (Add) (None, 16, 16, 128) 0  
['conv2d\_848[0][0]',  
'conv2d\_845[0][0]']

batch\_normalization\_606 (Batch Normalization) (None, 16, 16, 128) 512  
['add\_200[0][0]']

activation\_629 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_606[0][0]']

conv2d\_850 (Conv2D) (None, 16, 16, 128) 16512  
['activation\_629[0][0]']

batch\_normalization\_607 (Batch Normalization) (None, 16, 16, 128) 512  
['conv2d\_850[0][0]']

activation\_630 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_607[0][0]']

conv2d\_851 (Conv2D) (None, 16, 16, 128) 147584  
['activation\_630[0][0]']

batch\_normalization\_608 (Batch Normalization) (None, 16, 16, 128) 512  
['conv2d\_851[0][0]']

activation_631 (Activation)	(None, 16, 16, 128)	0
['batch_normalization_608[0][0]']		
conv2d_852 (Conv2D)	(None, 16, 16, 128)	16512
['activation_631[0][0]']		
conv2d_849 (Conv2D)	(None, 16, 16, 128)	16512
['add_200[0][0]']		
add_201 (Add)	(None, 16, 16, 128)	0
['conv2d_852[0][0]',		
'conv2d_849[0][0]']		
max_pooling2d_37 (MaxPooling2D)	(None, 8, 8, 128)	0
['add_201[0][0]']		
)		
batch_normalization_615 (Batch Normalization)	(None, 8, 8, 128)	512
['max_pooling2d_37[0][0]']		
activation_638 (Activation)	(None, 8, 8, 128)	0
['batch_normalization_615[0][0]']		
conv2d_862 (Conv2D)	(None, 8, 8, 128)	16512
['activation_638[0][0]']		
batch_normalization_616 (Batch Normalization)	(None, 8, 8, 128)	512
['conv2d_862[0][0]']		
activation_639 (Activation)	(None, 8, 8, 128)	0
['batch_normalization_616[0][0]']		

conv2d_863 (Conv2D) ['activation_639[0][0]']	(None, 8, 8, 128)	147584
batch_normalization_617 (Batch Normalization) ['conv2d_863[0][0]']	(None, 8, 8, 128)	512
activation_640 (Activation) ['batch_normalization_617[0][0]']	(None, 8, 8, 128)	0
conv2d_864 (Conv2D) ['activation_640[0][0]']	(None, 8, 8, 128)	16512
conv2d_861 (Conv2D) ['max_pooling2d_37[0][0]']	(None, 8, 8, 128)	16512
add_204 (Add) ['conv2d_864[0][0]', 'conv2d_861[0][0]']	(None, 8, 8, 128)	0
max_pooling2d_38 (MaxPooling2D) ['add_204[0][0]'] )	(None, 4, 4, 128)	0
batch_normalization_621 (Batch Normalization) ['max_pooling2d_38[0][0]']	(None, 4, 4, 128)	512
activation_644 (Activation) ['batch_normalization_621[0][0]']	(None, 4, 4, 128)	0
conv2d_870 (Conv2D) ['activation_644[0][0]']	(None, 4, 4, 128)	16512

```
batch_normalization_622 (Batch Normalization) (None, 4, 4, 128) 512
['conv2d_870[0][0]']
```

```
activation_645 (Activation) (None, 4, 4, 128) 0
['batch_normalization_622[0][0]']
```

```
conv2d_871 (Conv2D) (None, 4, 4, 128) 147584
['activation_645[0][0]']
```

```
batch_normalization_623 (Batch Normalization) (None, 4, 4, 128) 512
['conv2d_871[0][0]']
```

```
activation_646 (Activation) (None, 4, 4, 128) 0
['batch_normalization_623[0][0]']
```

```
conv2d_872 (Conv2D) (None, 4, 4, 128) 16512
['activation_646[0][0]']
```

```
conv2d_869 (Conv2D) (None, 4, 4, 128) 16512
['max_pooling2d_38[0][0]']
```

```
add_206 (Add) (None, 4, 4, 128) 0
['conv2d_872[0][0]',
 'conv2d_869[0][0]']
```

```
batch_normalization_624 (Batch Normalization) (None, 4, 4, 128) 512
['add_206[0][0]']
```

```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adam.py:105:
UserWarning: The `lr` argument is deprecated, use `learning_rate`
instead.
```

```
super(Adam, self).__init__(name, **kwargs)
```

Normalization)

activation\_647 (Activation) (None, 4, 4, 128) 0  
['batch\_normalization\_624[0][0]']

batch\_normalization\_618 (Batch (None, 8, 8, 128) 512  
['add\_204[0][0]']  
Normalization)

conv2d\_874 (Conv2D) (None, 4, 4, 128) 16512  
['activation\_647[0][0]']

activation\_641 (Activation) (None, 8, 8, 128) 0  
['batch\_normalization\_618[0][0]']

batch\_normalization\_625 (Batch (None, 4, 4, 128) 512  
['conv2d\_874[0][0]']  
Normalization)

conv2d\_866 (Conv2D) (None, 8, 8, 128) 16512  
['activation\_641[0][0]']

activation\_648 (Activation) (None, 4, 4, 128) 0  
['batch\_normalization\_625[0][0]']

batch\_normalization\_619 (Batch (None, 8, 8, 128) 512  
['conv2d\_866[0][0]']  
Normalization)

conv2d\_875 (Conv2D) (None, 4, 4, 128) 147584  
['activation\_648[0][0]']

activation\_642 (Activation) (None, 8, 8, 128) 0  
['batch\_normalization\_619[0][0]']

batch_normalization_626 (Batch Normalization)	(None, 4, 4, 128)	512
conv2d_867 (Conv2D)	(None, 8, 8, 128)	147584
activation_649 (Activation)	(None, 4, 4, 128)	0
batch_normalization_620 (Batch Normalization)	(None, 8, 8, 128)	512
conv2d_876 (Conv2D)	(None, 4, 4, 128)	16512
conv2d_873 (Conv2D)	(None, 4, 4, 128)	16512
activation_643 (Activation)	(None, 8, 8, 128)	0
add_207 (Add)	(None, 4, 4, 128)	0
conv2d_868 (Conv2D)	(None, 8, 8, 128)	16512
conv2d_865 (Conv2D)	(None, 8, 8, 128)	16512



```

up_sampling2d_26 (UpSampling2D (None, 8, 8, 128) 0
['add_207[0][0]']
)

add_205 (Add) (None, 8, 8, 128) 0
['conv2d_868[0][0]',
'conv2d_865[0][0]']

add_208 (Add) (None, 8, 8, 128) 0
['up_sampling2d_26[0][0]',
'add_205[0][0]']

batch_normalization_627 (Batch (None, 8, 8, 128) 512
['add_208[0][0]']
Normalization)

activation_650 (Activation) (None, 8, 8, 128) 0
['batch_normalization_627[0][0]']

conv2d_878 (Conv2D) (None, 8, 8, 128) 16512
['activation_650[0][0]']

batch_normalization_628 (Batch (None, 8, 8, 128) 512
['conv2d_878[0][0]']
Normalization)

activation_651 (Activation) (None, 8, 8, 128) 0
['batch_normalization_628[0][0]']

conv2d_879 (Conv2D) (None, 8, 8, 128) 147584
['activation_651[0][0]']

batch_normalization_612 (Batch (None, 16, 16, 128) 512
['add_201[0][0]']

```

Normalization)

batch\_normalization\_629 (Batch Normalization) (None, 8, 8, 128) 512  
['conv2d\_879[0][0]']

activation\_635 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_612[0][0]']

activation\_652 (Activation) (None, 8, 8, 128) 0  
['batch\_normalization\_629[0][0]']

conv2d\_858 (Conv2D) (None, 16, 16, 128) 16512  
['activation\_635[0][0]']

conv2d\_880 (Conv2D) (None, 8, 8, 128) 16512  
['activation\_652[0][0]']

conv2d\_877 (Conv2D) (None, 8, 8, 128) 16512  
['add\_208[0][0]']

batch\_normalization\_613 (Batch Normalization) (None, 16, 16, 128) 512  
['conv2d\_858[0][0]']

add\_209 (Add) (None, 8, 8, 128) 0  
['conv2d\_880[0][0]',  
'conv2d\_877[0][0]']

activation\_636 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_613[0][0]']

up\_sampling2d\_27 (UpSampling2D) (None, 16, 16, 128) 0  
['add\_209[0][0]']

)

conv2d\_859 (Conv2D) (None, 16, 16, 128) 147584  
['activation\_636[0][0]']

conv2d\_881 (Conv2D) (None, 16, 16, 128) 16512  
['up\_sampling2d\_27[0][0]']

batch\_normalization\_614 (Batch Normalization) (None, 16, 16, 128) 512  
['conv2d\_859[0][0]']

conv2d\_882 (Conv2D) (None, 16, 16, 128) 16512  
['conv2d\_881[0][0]']

activation\_637 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_614[0][0]']

activation\_653 (Activation) (None, 16, 16, 128) 0  
['conv2d\_882[0][0]']

conv2d\_860 (Conv2D) (None, 16, 16, 128) 16512  
['activation\_637[0][0]']

conv2d\_857 (Conv2D) (None, 16, 16, 128) 16512  
['add\_201[0][0]']

lambda\_23 (Lambda) (None, 16, 16, 128) 0  
['activation\_653[0][0]']

add\_203 (Add) (None, 16, 16, 128) 0  
['conv2d\_860[0][0]',  
'conv2d\_857[0][0]']

```
multiply_23 (Multiply)          (None, 16, 16, 128)  0
['lambda_23[0][0]',
'add_203[0][0]']
```

```
batch_normalization_630 (Batch Normalization) (None, 16, 16, 128)  512
['multiply_23[0][0]']
```

```
activation_654 (Activation)      (None, 16, 16, 128)  0
['batch_normalization_630[0][0]']
```

```
conv2d_884 (Conv2D)              (None, 16, 16, 128) 16512
['activation_654[0][0]']
```

```
batch_normalization_631 (Batch Normalization) (None, 16, 16, 128)  512
['conv2d_884[0][0]']
```

```
activation_655 (Activation)      (None, 16, 16, 128)  0
['batch_normalization_631[0][0]']
```

```
conv2d_885 (Conv2D)              (None, 16, 16, 128) 147584
['activation_655[0][0]']
```

```
batch_normalization_632 (Batch Normalization) (None, 16, 16, 128)  512
['conv2d_885[0][0]']
```

```
activation_656 (Activation)      (None, 16, 16, 128)  0
['batch_normalization_632[0][0]']
```

```
conv2d_886 (Conv2D)              (None, 16, 16, 128) 16512
['activation_656[0][0]']
```

conv2d\_883 (Conv2D) (None, 16, 16, 128) 16512  
['multiply\_23[0][0]']

add\_210 (Add) (None, 16, 16, 128) 0  
['conv2d\_886[0][0]',  
'conv2d\_883[0][0]']

batch\_normalization\_633 (Batch Normalization) (None, 16, 16, 128) 512  
['add\_210[0][0]']

activation\_657 (Activation) (None, 16, 16, 128) 0  
['batch\_normalization\_633[0][0]']

conv2d\_888 (Conv2D) (None, 16, 16, 64) 8256  
['activation\_657[0][0]']

batch\_normalization\_634 (Batch Normalization) (None, 16, 16, 64) 256  
['conv2d\_888[0][0]']

activation\_658 (Activation) (None, 16, 16, 64) 0  
['batch\_normalization\_634[0][0]']

conv2d\_889 (Conv2D) (None, 8, 8, 64) 36928  
['activation\_658[0][0]']

batch\_normalization\_635 (Batch Normalization) (None, 8, 8, 64) 256  
['conv2d\_889[0][0]']

activation\_659 (Activation) (None, 8, 8, 64) 0  
['batch\_normalization\_635[0][0]']

conv2d_890 (Conv2D) ['activation_659[0][0]']	(None, 8, 8, 256)	16640
conv2d_887 (Conv2D) ['add_210[0][0]']	(None, 8, 8, 256)	33024
add_211 (Add) ['conv2d_890[0][0]', 'conv2d_887[0][0]']	(None, 8, 8, 256)	0
batch_normalization_636 (Batch Normalization) ['add_211[0][0]']	(None, 8, 8, 256)	1024
activation_660 (Activation) ['batch_normalization_636[0][0]']	(None, 8, 8, 256)	0
conv2d_892 (Conv2D) ['activation_660[0][0]']	(None, 8, 8, 256)	65792
batch_normalization_637 (Batch Normalization) ['conv2d_892[0][0]']	(None, 8, 8, 256)	1024
activation_661 (Activation) ['batch_normalization_637[0][0]']	(None, 8, 8, 256)	0
conv2d_893 (Conv2D) ['activation_661[0][0]']	(None, 8, 8, 256)	590080
batch_normalization_638 (Batch Normalization) ['conv2d_893[0][0]']	(None, 8, 8, 256)	1024

activation_662 (Activation) ['batch_normalization_638[0][0]']	(None, 8, 8, 256)	0
conv2d_894 (Conv2D) ['activation_662[0][0]']	(None, 8, 8, 256)	65792
conv2d_891 (Conv2D) ['add_211[0][0]']	(None, 8, 8, 256)	65792
add_212 (Add) ['conv2d_894[0][0]', 'conv2d_891[0][0]']	(None, 8, 8, 256)	0
max_pooling2d_39 (MaxPooling2D) ['add_212[0][0]'] )	(None, 4, 4, 256)	0
batch_normalization_645 (Batch Normalization) ['max_pooling2d_39[0][0]']	(None, 4, 4, 256)	1024
activation_669 (Activation) ['batch_normalization_645[0][0]']	(None, 4, 4, 256)	0
conv2d_904 (Conv2D) ['activation_669[0][0]']	(None, 4, 4, 256)	65792
batch_normalization_646 (Batch Normalization) ['conv2d_904[0][0]']	(None, 4, 4, 256)	1024
activation_670 (Activation) ['batch_normalization_646[0][0]']	(None, 4, 4, 256)	0

conv2d_905 (Conv2D) ['activation_670[0][0]']	(None, 4, 4, 256)	590080
batch_normalization_647 (Batch Normalization) ['conv2d_905[0][0]']	(None, 4, 4, 256)	1024
activation_671 (Activation) ['batch_normalization_647[0][0]']	(None, 4, 4, 256)	0
conv2d_906 (Conv2D) ['activation_671[0][0]']	(None, 4, 4, 256)	65792
conv2d_903 (Conv2D) ['max_pooling2d_39[0][0]']	(None, 4, 4, 256)	65792
add_215 (Add) ['conv2d_906[0][0]', 'conv2d_903[0][0]']	(None, 4, 4, 256)	0
batch_normalization_648 (Batch Normalization) ['add_215[0][0]']	(None, 4, 4, 256)	1024
activation_672 (Activation) ['batch_normalization_648[0][0]']	(None, 4, 4, 256)	0
conv2d_908 (Conv2D) ['activation_672[0][0]']	(None, 4, 4, 256)	65792
batch_normalization_649 (Batch Normalization) ['conv2d_908[0][0]']	(None, 4, 4, 256)	1024



activation_673 (Activation) ['batch_normalization_649[0][0]']	(None, 4, 4, 256)	0
conv2d_909 (Conv2D) ['activation_673[0][0]']	(None, 4, 4, 256)	590080
batch_normalization_642 (Batch ['add_212[0][0]'] Normalization)	(None, 8, 8, 256)	1024
batch_normalization_650 (Batch ['conv2d_909[0][0]'] Normalization)	(None, 4, 4, 256)	1024
activation_666 (Activation) ['batch_normalization_642[0][0]']	(None, 8, 8, 256)	0
activation_674 (Activation) ['batch_normalization_650[0][0]']	(None, 4, 4, 256)	0
conv2d_900 (Conv2D) ['activation_666[0][0]']	(None, 8, 8, 256)	65792
conv2d_910 (Conv2D) ['activation_674[0][0]']	(None, 4, 4, 256)	65792
conv2d_907 (Conv2D) ['add_215[0][0]']	(None, 4, 4, 256)	65792
batch_normalization_643 (Batch ['conv2d_900[0][0]'] Normalization)	(None, 8, 8, 256)	1024
add_216 (Add) ['conv2d_910[0][0]',	(None, 4, 4, 256)	0

'conv2d\_907[0][0]']

activation\_667 (Activation) (None, 8, 8, 256) 0  
['batch\_normalization\_643[0][0]']

up\_sampling2d\_28 (UpSampling2D) (None, 8, 8, 256) 0  
['add\_216[0][0]']  
)

conv2d\_901 (Conv2D) (None, 8, 8, 256) 590080  
['activation\_667[0][0]']

conv2d\_911 (Conv2D) (None, 8, 8, 256) 65792  
['up\_sampling2d\_28[0][0]']

batch\_normalization\_644 (Batch Normalization) (None, 8, 8, 256) 1024  
['conv2d\_901[0][0]']

conv2d\_912 (Conv2D) (None, 8, 8, 256) 65792  
['conv2d\_911[0][0]']

activation\_668 (Activation) (None, 8, 8, 256) 0  
['batch\_normalization\_644[0][0]']

activation\_675 (Activation) (None, 8, 8, 256) 0  
['conv2d\_912[0][0]']

conv2d\_902 (Conv2D) (None, 8, 8, 256) 65792  
['activation\_668[0][0]']

conv2d\_899 (Conv2D) (None, 8, 8, 256) 65792  
['add\_212[0][0]']

lambda_24 (Lambda) ['activation_675[0][0]']	(None, 8, 8, 256)	0
add_214 (Add) ['conv2d_902[0][0]', 'conv2d_899[0][0]']	(None, 8, 8, 256)	0
multiply_24 (Multiply) ['lambda_24[0][0]', 'add_214[0][0]']	(None, 8, 8, 256)	0
batch_normalization_651 (Batch Normalization) ['multiply_24[0][0]']	(None, 8, 8, 256)	1024
activation_676 (Activation) ['batch_normalization_651[0][0]']	(None, 8, 8, 256)	0
conv2d_914 (Conv2D) ['activation_676[0][0]']	(None, 8, 8, 256)	65792
batch_normalization_652 (Batch Normalization) ['conv2d_914[0][0]']	(None, 8, 8, 256)	1024
activation_677 (Activation) ['batch_normalization_652[0][0]']	(None, 8, 8, 256)	0
conv2d_915 (Conv2D) ['activation_677[0][0]']	(None, 8, 8, 256)	590080
batch_normalization_653 (Batch Normalization) ['conv2d_915[0][0]']	(None, 8, 8, 256)	1024

activation_678 (Activation) ['batch_normalization_653[0][0]']	(None, 8, 8, 256)	0
conv2d_916 (Conv2D) ['activation_678[0][0]']	(None, 8, 8, 256)	65792
conv2d_913 (Conv2D) ['multiply_24[0][0]']	(None, 8, 8, 256)	65792
add_217 (Add) ['conv2d_916[0][0]', 'conv2d_913[0][0]']	(None, 8, 8, 256)	0
batch_normalization_654 (Batch Normalization) ['add_217[0][0]']	(None, 8, 8, 256)	1024
activation_679 (Activation) ['batch_normalization_654[0][0]']	(None, 8, 8, 256)	0
conv2d_918 (Conv2D) ['activation_679[0][0]']	(None, 8, 8, 256)	65792
batch_normalization_655 (Batch Normalization) ['conv2d_918[0][0]']	(None, 8, 8, 256)	1024
activation_680 (Activation) ['batch_normalization_655[0][0]']	(None, 8, 8, 256)	0
conv2d_919 (Conv2D) ['activation_680[0][0]']	(None, 8, 8, 256)	590080

batch_normalization_656 (Batch Normalization)	(None, 8, 8, 256)	1024
['conv2d_919[0][0]']		
activation_681 (Activation)	(None, 8, 8, 256)	0
['batch_normalization_656[0][0]']		
conv2d_920 (Conv2D)	(None, 8, 8, 256)	65792
['activation_681[0][0]']		
conv2d_917 (Conv2D)	(None, 8, 8, 256)	65792
['add_217[0][0]']		
add_218 (Add)	(None, 8, 8, 256)	0
['conv2d_920[0][0]',		
'conv2d_917[0][0]']		
max_pooling2d_40 (MaxPooling2D)	(None, 4, 4, 256)	0
['add_218[0][0]']		
)		
batch_normalization_663 (Batch Normalization)	(None, 4, 4, 256)	1024
['max_pooling2d_40[0][0]']		
activation_688 (Activation)	(None, 4, 4, 256)	0
['batch_normalization_663[0][0]']		
conv2d_930 (Conv2D)	(None, 4, 4, 256)	65792
['activation_688[0][0]']		
batch_normalization_664 (Batch Normalization)	(None, 4, 4, 256)	1024
['conv2d_930[0][0]']		

activation_689 (Activation) ['batch_normalization_664[0][0]']	(None, 4, 4, 256)	0
conv2d_931 (Conv2D) ['activation_689[0][0]']	(None, 4, 4, 256)	590080
batch_normalization_665 (Batch ['conv2d_931[0][0]'] Normalization)	(None, 4, 4, 256)	1024
activation_690 (Activation) ['batch_normalization_665[0][0]']	(None, 4, 4, 256)	0
conv2d_932 (Conv2D) ['activation_690[0][0]']	(None, 4, 4, 256)	65792
conv2d_929 (Conv2D) ['max_pooling2d_40[0][0]']	(None, 4, 4, 256)	65792
add_221 (Add) ['conv2d_932[0][0]', 'conv2d_929[0][0]']	(None, 4, 4, 256)	0
batch_normalization_666 (Batch ['add_221[0][0]'] Normalization)	(None, 4, 4, 256)	1024
activation_691 (Activation) ['batch_normalization_666[0][0]']	(None, 4, 4, 256)	0
conv2d_934 (Conv2D) ['activation_691[0][0]']	(None, 4, 4, 256)	65792

batch_normalization_667 (Batch Normalization)	(None, 4, 4, 256)	1024
['conv2d_934[0][0]']		
activation_692 (Activation)	(None, 4, 4, 256)	0
['batch_normalization_667[0][0]']		
conv2d_935 (Conv2D)	(None, 4, 4, 256)	590080
['activation_692[0][0]']		
batch_normalization_660 (Batch Normalization)	(None, 8, 8, 256)	1024
['add_218[0][0]']		
batch_normalization_668 (Batch Normalization)	(None, 4, 4, 256)	1024
['conv2d_935[0][0]']		
activation_685 (Activation)	(None, 8, 8, 256)	0
['batch_normalization_660[0][0]']		
activation_693 (Activation)	(None, 4, 4, 256)	0
['batch_normalization_668[0][0]']		
conv2d_926 (Conv2D)	(None, 8, 8, 256)	65792
['activation_685[0][0]']		
conv2d_936 (Conv2D)	(None, 4, 4, 256)	65792
['activation_693[0][0]']		
conv2d_933 (Conv2D)	(None, 4, 4, 256)	65792
['add_221[0][0]']		
batch_normalization_661 (Batch Normalization)	(None, 8, 8, 256)	1024
['conv2d_926[0][0]']		

Normalization)

add_222 (Add) ['conv2d_936[0][0]', 'conv2d_933[0][0]']	(None, 4, 4, 256)	0
activation_686 (Activation) ['batch_normalization_661[0][0]']	(None, 8, 8, 256)	0
up_sampling2d_29 (UpSampling2D) ['add_222[0][0]'] )	(None, 8, 8, 256)	0
conv2d_927 (Conv2D) ['activation_686[0][0]']	(None, 8, 8, 256)	590080
conv2d_937 (Conv2D) ['up_sampling2d_29[0][0]']	(None, 8, 8, 256)	65792
batch_normalization_662 (Batch ['conv2d_927[0][0]'] Normalization)	(None, 8, 8, 256)	1024
conv2d_938 (Conv2D) ['conv2d_937[0][0]']	(None, 8, 8, 256)	65792
activation_687 (Activation) ['batch_normalization_662[0][0]']	(None, 8, 8, 256)	0
activation_694 (Activation) ['conv2d_938[0][0]']	(None, 8, 8, 256)	0
conv2d_928 (Conv2D) ['activation_687[0][0]']	(None, 8, 8, 256)	65792



conv2d_925 (Conv2D) ['add_218[0][0]']	(None, 8, 8, 256)	65792
lambda_25 (Lambda) ['activation_694[0][0]']	(None, 8, 8, 256)	0
add_220 (Add) ['conv2d_928[0][0]', 'conv2d_925[0][0]']	(None, 8, 8, 256)	0
multiply_25 (Multiply) ['lambda_25[0][0]', 'add_220[0][0]']	(None, 8, 8, 256)	0
batch_normalization_669 (Batch Normalization) ['multiply_25[0][0]']	(None, 8, 8, 256)	1024
activation_695 (Activation) ['batch_normalization_669[0][0]']	(None, 8, 8, 256)	0
conv2d_940 (Conv2D) ['activation_695[0][0]']	(None, 8, 8, 256)	65792
batch_normalization_670 (Batch Normalization) ['conv2d_940[0][0]']	(None, 8, 8, 256)	1024
activation_696 (Activation) ['batch_normalization_670[0][0]']	(None, 8, 8, 256)	0
conv2d_941 (Conv2D) ['activation_696[0][0]']	(None, 8, 8, 256)	590080

batch_normalization_671 (Batch Normalization)	(None, 8, 8, 256)	1024
['conv2d_941[0][0]']		
activation_697 (Activation)	(None, 8, 8, 256)	0
['batch_normalization_671[0][0]']		
conv2d_942 (Conv2D)	(None, 8, 8, 256)	65792
['activation_697[0][0]']		
conv2d_939 (Conv2D)	(None, 8, 8, 256)	65792
['multiply_25[0][0]']		
add_223 (Add)	(None, 8, 8, 256)	0
['conv2d_942[0][0]',		
'conv2d_939[0][0]']		
batch_normalization_672 (Batch Normalization)	(None, 8, 8, 256)	1024
['add_223[0][0]']		
activation_698 (Activation)	(None, 8, 8, 256)	0
['batch_normalization_672[0][0]']		
conv2d_944 (Conv2D)	(None, 8, 8, 128)	32896
['activation_698[0][0]']		
batch_normalization_673 (Batch Normalization)	(None, 8, 8, 128)	512
['conv2d_944[0][0]']		
activation_699 (Activation)	(None, 8, 8, 128)	0
['batch_normalization_673[0][0]']		

conv2d_945 (Conv2D) ['activation_699[0][0]']	(None, 8, 8, 128)	147584
batch_normalization_674 (Batch Normalization) ['conv2d_945[0][0]']	(None, 8, 8, 128)	512
activation_700 (Activation) ['batch_normalization_674[0][0]']	(None, 8, 8, 128)	0
conv2d_946 (Conv2D) ['activation_700[0][0]']	(None, 8, 8, 512)	66048
conv2d_943 (Conv2D) ['add_223[0][0]']	(None, 8, 8, 512)	131584
add_224 (Add) ['conv2d_946[0][0]', 'conv2d_943[0][0]']	(None, 8, 8, 512)	0
batch_normalization_675 (Batch Normalization) ['add_224[0][0]']	(None, 8, 8, 512)	2048
activation_701 (Activation) ['batch_normalization_675[0][0]']	(None, 8, 8, 512)	0
conv2d_948 (Conv2D) ['activation_701[0][0]']	(None, 8, 8, 128)	65664
batch_normalization_676 (Batch Normalization) ['conv2d_948[0][0]']	(None, 8, 8, 128)	512

activation_702 (Activation) ['batch_normalization_676[0][0]']	(None, 8, 8, 128)	0
conv2d_949 (Conv2D) ['activation_702[0][0]']	(None, 8, 8, 128)	147584
batch_normalization_677 (Batch ['conv2d_949[0][0]'] Normalization)	(None, 8, 8, 128)	512
activation_703 (Activation) ['batch_normalization_677[0][0]']	(None, 8, 8, 128)	0
conv2d_950 (Conv2D) ['activation_703[0][0]']	(None, 8, 8, 512)	66048
conv2d_947 (Conv2D) ['add_224[0][0]']	(None, 8, 8, 512)	262656
add_225 (Add) ['conv2d_950[0][0]', 'conv2d_947[0][0]']	(None, 8, 8, 512)	0
batch_normalization_678 (Batch ['add_225[0][0]'] Normalization)	(None, 8, 8, 512)	2048
activation_704 (Activation) ['batch_normalization_678[0][0]']	(None, 8, 8, 512)	0
conv2d_952 (Conv2D) ['activation_704[0][0]']	(None, 8, 8, 128)	65664

batch_normalization_679 (Batch Normalization)	(None, 8, 8, 128)	512
['conv2d_952[0][0]']		
activation_705 (Activation)	(None, 8, 8, 128)	0
['batch_normalization_679[0][0]']		
conv2d_953 (Conv2D)	(None, 8, 8, 128)	147584
['activation_705[0][0]']		
batch_normalization_680 (Batch Normalization)	(None, 8, 8, 128)	512
['conv2d_953[0][0]']		
activation_706 (Activation)	(None, 8, 8, 128)	0
['batch_normalization_680[0][0]']		
conv2d_954 (Conv2D)	(None, 8, 8, 512)	66048
['activation_706[0][0]']		
conv2d_951 (Conv2D)	(None, 8, 8, 512)	262656
['add_225[0][0]']		
add_226 (Add)	(None, 8, 8, 512)	0
['conv2d_954[0][0]',		
'conv2d_951[0][0]']		
batch_normalization_681 (Batch Normalization)	(None, 8, 8, 512)	2048
['add_226[0][0]']		
activation_707 (Activation)	(None, 8, 8, 512)	0
['batch_normalization_681[0][0]']		

average_pooling2d_10 (AverageP ['activation_707[0][0]'] ooling2D)	(None, 5, 5, 512)	0
flatten_10 (Flatten) ['average_pooling2d_10[0][0]']	(None, 12800)	0
dropout_4 (Dropout) ['flatten_10[0][0]']	(None, 12800)	0
dense_10 (Dense) ['dropout_4[0][0]']	(None, 24)	307224

```
=====
Total params: 11,687,640
Trainable params: 11,660,312
Non-trainable params: 27,328
```

---

#### *# training*

```
model = training(model, '56paper', batch_size=64, epc=15)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:82:
UserWarning: `Model.fit_generator` is deprecated and will be removed
in a future version. Please use `Model.fit`, which supports
generators.
```

```
Learning rate: 0.0001
```

```
Epoch 1/15
```

```
21/21 [=====] - 138s 6s/step - loss: 2.5464 -
accuracy: 0.2444 - val_loss: 2.4484 - val_accuracy: 0.1406 - lr:
1.0000e-04
```

```
Learning rate: 0.0001
```

```
Epoch 2/15
```

```
21/21 [=====] - 124s 6s/step - loss: 2.0333 -
accuracy: 0.3569 - val_loss: 2.3308 - val_accuracy: 0.3594 - lr:
1.0000e-04
```

```
Learning rate: 0.0001
```

```
Epoch 3/15
```

```
21/21 [=====] - 125s 6s/step - loss: 1.7319 -
accuracy: 0.4352 - val_loss: 3.0002 - val_accuracy: 0.0938 - lr:
1.0000e-04
```

```
Learning rate: 0.0001
```

Epoch 4/15  
21/21 [=====] - 124s 6s/step - loss: 1.5344 -  
accuracy: 0.4717 - val\_loss: 3.1513 - val\_accuracy: 0.1406 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 5/15  
21/21 [=====] - 124s 6s/step - loss: 1.2931 -  
accuracy: 0.5617 - val\_loss: 3.3637 - val\_accuracy: 0.1250 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 6/15  
21/21 [=====] - 124s 6s/step - loss: 1.1914 -  
accuracy: 0.5826 - val\_loss: 3.7710 - val\_accuracy: 0.0781 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 7/15  
21/21 [=====] - ETA: 0s - loss: 1.1127 -  
accuracy: 0.6059  
Epoch 7: ReduceLROnPlateau reducing learning rate to  
9.999999747378752e-06.  
21/21 [=====] - 125s 6s/step - loss: 1.1127 -  
accuracy: 0.6059 - val\_loss: 3.7169 - val\_accuracy: 0.1406 - lr:  
1.0000e-05  
Learning rate: 0.0001  
Epoch 8/15  
21/21 [=====] - 125s 6s/step - loss: 0.9960 -  
accuracy: 0.6610 - val\_loss: 3.3175 - val\_accuracy: 0.1719 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 9/15  
21/21 [=====] - 125s 6s/step - loss: 0.8725 -  
accuracy: 0.6998 - val\_loss: 2.8973 - val\_accuracy: 0.2188 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 10/15  
21/21 [=====] - 125s 6s/step - loss: 0.9759 -  
accuracy: 0.6757 - val\_loss: 2.6949 - val\_accuracy: 0.4062 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 11/15  
21/21 [=====] - 125s 6s/step - loss: 0.9082 -  
accuracy: 0.6912 - val\_loss: 2.8680 - val\_accuracy: 0.2656 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 12/15  
21/21 [=====] - 125s 6s/step - loss: 0.7696 -  
accuracy: 0.7347 - val\_loss: 2.2214 - val\_accuracy: 0.4844 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 13/15

```
21/21 [=====] - 125s 6s/step - loss: 0.6911 -  
accuracy: 0.7657 - val_loss: 1.9289 - val_accuracy: 0.4375 - lr:  
1.0000e-04
```

Learning rate: 0.0001

Epoch 14/15

```
21/21 [=====] - 125s 6s/step - loss: 0.7060 -  
accuracy: 0.7548 - val_loss: 1.6799 - val_accuracy: 0.5000 - lr:  
1.0000e-04
```

Learning rate: 0.0001

Epoch 15/15

```
21/21 [=====] - 125s 6s/step - loss: 0.6785 -  
accuracy: 0.7789 - val_loss: 1.8607 - val_accuracy: 0.5781 - lr:  
1.0000e-04
```

Time taken by above cell is 33.45548965533575.

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:88:  
UserWarning: `Model.evaluate_generator` is deprecated and will be  
removed in a future version. Please use `Model.evaluate`, which  
supports generators.
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:89:  
UserWarning: `Model.evaluate_generator` is deprecated and will be  
removed in a future version. Please use `Model.evaluate`, which  
supports generators.
```

```
6/6 [=====] - 8s 1s/step - loss: 1.8706 -  
accuracy: 0.5339  
validation loss: 1.880006194114685  
validation accuracy: 0.5666666626930237  
Test loss: 1.870613694190979  
Test accuracy: 0.5338753461837769
```

In comparison previous models, introducing batch sizes and longer training resulted in increase of test accuracy. However, the value of 54% is still non-satisfying. Therefore, in next model batch size will be reduced from 64 to 32.

*# training*

```
model2 = training(model, '56paper2', batch_size=32, epc=15)
```

Learning rate: 0.0001

Epoch 1/15

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:82:  
UserWarning: `Model.fit_generator` is deprecated and will be removed  
in a future version. Please use `Model.fit`, which supports  
generators.
```

```
42/42 [=====] - 130s 3s/step - loss: 0.7617 -  
accuracy: 0.7366 - val_loss: 0.7088 - val_accuracy: 0.7396 - lr:  
1.0000e-04
```

Learning rate: 0.0001

Epoch 2/15



42/42 [=====] - 130s 3s/step - loss: 0.7444 -  
accuracy: 0.7403 - val\_loss: 0.5493 - val\_accuracy: 0.8021 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 3/15  
42/42 [=====] - 129s 3s/step - loss: 0.6762 -  
accuracy: 0.7820 - val\_loss: 0.3894 - val\_accuracy: 0.8750 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 4/15  
42/42 [=====] - 129s 3s/step - loss: 0.6689 -  
accuracy: 0.7759 - val\_loss: 0.3714 - val\_accuracy: 0.8542 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 5/15  
42/42 [=====] - 130s 3s/step - loss: 0.6087 -  
accuracy: 0.8070 - val\_loss: 0.4287 - val\_accuracy: 0.8750 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 6/15  
42/42 [=====] - 130s 3s/step - loss: 0.6219 -  
accuracy: 0.7964 - val\_loss: 0.7706 - val\_accuracy: 0.7812 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 7/15  
42/42 [=====] - 130s 3s/step - loss: 0.5533 -  
accuracy: 0.8221 - val\_loss: 0.7607 - val\_accuracy: 0.7604 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 8/15  
42/42 [=====] - 130s 3s/step - loss: 0.5305 -  
accuracy: 0.8244 - val\_loss: 0.3299 - val\_accuracy: 0.9062 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 9/15  
42/42 [=====] - 130s 3s/step - loss: 0.5261 -  
accuracy: 0.8266 - val\_loss: 0.7646 - val\_accuracy: 0.7604 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 10/15  
42/42 [=====] - 130s 3s/step - loss: 0.4781 -  
accuracy: 0.8342 - val\_loss: 0.3258 - val\_accuracy: 0.8750 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 11/15  
42/42 [=====] - 130s 3s/step - loss: 0.4569 -  
accuracy: 0.8554 - val\_loss: 0.6177 - val\_accuracy: 0.8125 - lr:  
1.0000e-04  
Learning rate: 0.0001  
Epoch 12/15

```

42/42 [=====] - 129s 3s/step - loss: 0.4945 -
accuracy: 0.8433 - val_loss: 1.2858 - val_accuracy: 0.6667 - lr:
1.0000e-04
Learning rate: 0.0001
Epoch 13/15
42/42 [=====] - ETA: 0s - loss: 0.4788 -
accuracy: 0.8471
Epoch 13: ReduceLROnPlateau reducing learning rate to
9.999999747378752e-06.
42/42 [=====] - 130s 3s/step - loss: 0.4788 -
accuracy: 0.8471 - val_loss: 0.4296 - val_accuracy: 0.8542 - lr:
1.0000e-05
Learning rate: 0.0001
Epoch 14/15
42/42 [=====] - 130s 3s/step - loss: 0.4559 -
accuracy: 0.8554 - val_loss: 0.5791 - val_accuracy: 0.8646 - lr:
1.0000e-04
Learning rate: 0.0001
Epoch 15/15
42/42 [=====] - 130s 3s/step - loss: 0.4432 -
accuracy: 0.8607 - val_loss: 0.1723 - val_accuracy: 0.9688 - lr:
1.0000e-04
Time taken by above cell is 34.04429015715917.

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:88:
UserWarning: `Model.evaluate_generator` is deprecated and will be
removed in a future version. Please use `Model.evaluate`, which
supports generators.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:89:
UserWarning: `Model.evaluate_generator` is deprecated and will be
removed in a future version. Please use `Model.evaluate`, which
supports generators.

```

```

12/12 [=====] - 8s 691ms/step - loss: 0.4355
- accuracy: 0.8482
validation loss: 0.2170329988002777
validation accuracy: 0.949999988079071
Test loss: 0.43545252084732056
Test accuracy: 0.848238468170166

```

The last configuration finally gave satisfying results in case of test accuracy.

## Final remarks

It can be seen that sometimes validation accuracy is greater than training accuracy. This might happen due to the fact that the dataset class distribution is imbalanced - out of 24 possible classes not all exist in all subsets (split function uses random state method).

Moreover, using dropout some information of the model is lost during training, while in validation all pixels are present - dropout somehow makes it harder for the network to give right answers during training.

Finally, the dataset should consist of more images, of which equal number would belong to each class.

### Performance Analysis

```
%load_ext tensorboard
```

```
%tensorboard --logdir /content/drive/MyDrive/Dir/Module/Logs/56paper2/
```

Reusing TensorBoard on port 6006 (pid 11431), started 0:09:32 ago.  
(Use '!kill 11431' to kill it.)

```
<IPython.core.display.Javascript object>
```