Q1

Cutout regularization aims to use both the advantages from dropout as well as the advantages of data augmenation with regards to regularization. It is a regularization technique applied to convolutional neural neworks where we remove neighboring parts of input images and also augmente the dataset with obstructed versions of existing samples. The idea is to extend the dropout in the sense that we also apply a spatial prior that is relevant for images data as images share local connectivities. The ultimate goal is to force the learning algorithm to get a global sense of the full image rather than learning specific features that might endanger the predictions on unseen data where those features are not present. Finally, the key difference with dropout is that cutout is applied to the input layer so that visual features are removed directly from the image and no dropout is applied in the hidden layer and therefore the image itself is modified which makes cutout similar to dropout. ¶

In [2]:
```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras import models, layers
import tensorflow.keras as keras
from tensorflow.keras.layers import BatchNormalization, LayerNormalizati
on
import tensorflow as tf
from tensorflow.keras.layers import Dropout
import numpy as np
import matplotlib.pyplot as plt
if tf.test.gpu_device_name():
    print('Default GPU Device: {}'.format(tf.test.gpu_device_name()))
else:
    print("Please install GPU version of TF")
from matplotlib import pyplot
from tensorflow.keras.datasets import cifar10
from numpy import save
from numpy import savetxt
from numpy import loadtxt
from time import time
import pandas as pd



from tensorflow.keras.layers import Dense, Conv2D
from tensorflow.keras.layers import BatchNormalization, Activation
from tensorflow.keras.layers import AveragePooling2D, Input
from tensorflow.keras.layers import Flatten, add
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ModelCheckpoint, LearningRateSche
duler
from tensorflow.keras.callbacks import ReduceLROnPlateau
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.regularizers import l2
from tensorflow.keras.models import Model
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.utils import plot_model
from tensorflow.keras.utils import to_categorical
import numpy as np
import os
import math
import numpy as np
from tensorflow.keras.datasets import cifar10
import matplotlib.pyplot as plt
from numpy import savetxt
from numpy import loadtxt
```

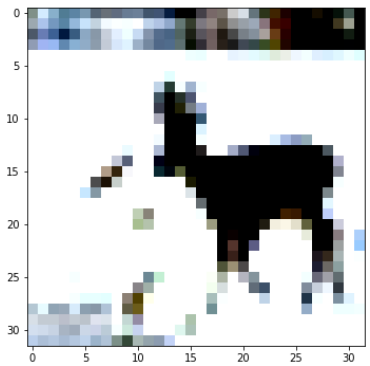
```
/Users/taziy/anaconda3/lib/python3.7/site-packages/tensorboard/compat/t
ensorflow_stub/dtypes.py:541: FutureWarning: Passing (type, 1) or '1typ
e' as a synonym of type is deprecated; in a future version of numpy, it
will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
/Users/taziy/anaconda3/lib/python3.7/site-packages/tensorboard/compat/t
ensorflow_stub/dtypes.py:542: FutureWarning: Passing (type, 1) or '1typ
e' as a synonym of type is deprecated; in a future version of numpy, it
will be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
/Users/taziy/anaconda3/lib/python3.7/site-packages/tensorboard/compat/t
ensorflow_stub/dtypes.py:543: FutureWarning: Passing (type, 1) or '1typ
e' as a synonym of type is deprecated; in a future version of numpy, it
will be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
/Users/taziy/anaconda3/lib/python3.7/site-packages/tensorboard/compat/t
ensorflow_stub/dtypes.py:544: FutureWarning: Passing (type, 1) or '1typ
e' as a synonym of type is deprecated; in a future version of numpy, it
will be understood as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
/Users/taziy/anaconda3/lib/python3.7/site-packages/tensorboard/compat/t
ensorflow_stub/dtypes.py:545: FutureWarning: Passing (type, 1) or '1typ
e' as a synonym of type is deprecated; in a future version of numpy, it
will be understood as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
/Users/taziy/anaconda3/lib/python3.7/site-packages/tensorboard/compat/t
ensorflow_stub/dtypes.py:550: FutureWarning: Passing (type, 1) or '1typ
e' as a synonym of type is deprecated; in a future version of numpy, it
will be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])

Please install GPU version of TF
```

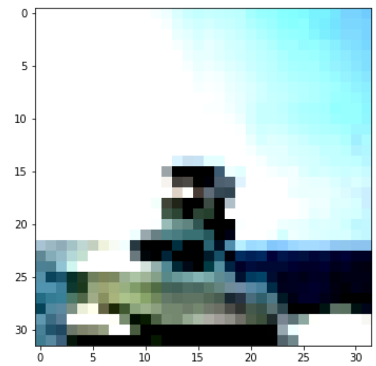
In [ ]:

```
def apply_mask(image, size=12, n_squares=1):
    h, w, channels = image.shape
    new_image = image
    for _ in range(n_squares):
        y = np.random.randint(h)
        x = np.random.randint(w)
        y1 = np.clip(y - size // 2, 0, h)
        y2 = np.clip(y + size // 2, 0, h)
        x1 = np.clip(x - size // 2, 0, w)
        x2 = np.clip(x + size // 2, 0, w)
        new_image[y1:y2,x1:x2,:] = 0
    return new_image
```

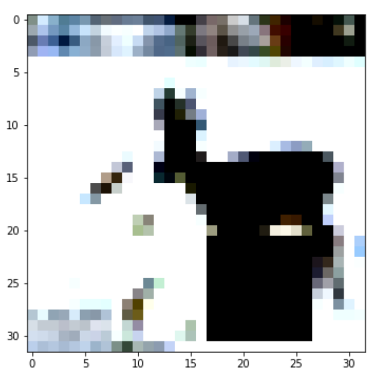
In [ ]:
```
(x_train, y_train), (x_test, y_test) = cifar10.load_data()
# Data normalization
m, std = np.mean(x_train), np.std(x_train)
x_train = (x_train - m)/std
x_test = (x_test - m)/std
y_train = tf.keras.utils.to_categorical(y_train)
y_test = tf.keras.utils.to_categorical(y_test)
plt.figure(figsize=(40,20))
print("Original images:")
images_index = [20,100]
for i in range(2):
    tmp = x_train[images_index[i]]
    plt.subplot(330 + 1 + i)
    plt.imshow(tmp)
plt.show()
print("Images with cutout:")
plt.figure(figsize=(40,20))
for i in range(2):
    tmp = x_train[images_index[i]]
    plt.subplot(330 + 1 + i)
    plt.imshow(apply_mask(tmp,size=10))
plt.show()
```

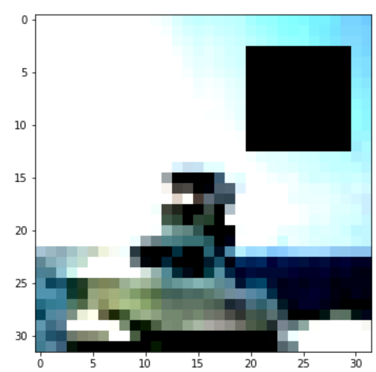
```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170500096/170498071 [==============================] - 13s 0us/step

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

Original images:
```





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

Images with cutout:
```





Q2

In [ ]:
```
# training parameters
batch_size = 64
epochs =100
data_augmentation = True
num_classes = 10

# subtracting pixel mean improves accuracy
subtract_pixel_mean = True
n = 7 # so that we use resnet 44 because 6*n+2
depth = n * 6 + 2


# model name, depth and version
model_type = 'ResNet44'

# load the CIFAR10 data.
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

# input image dimensions.
input_shape = x_train.shape[1:]

# normalize data.
x_train = x_train.astype('float32') / 255
x_test = x_test.astype('float32') / 255

# if subtract pixel mean is enabled
if subtract_pixel_mean:
    x_train_mean = np.mean(x_train, axis=0)
    x_train -= x_train_mean
    x_test -= x_train_mean

y_train = to_categorical(y_train, num_classes)
y_test = to_categorical(y_test, num_classes)


def lr_schedule(epoch):
    lr = 1e-3 if epoch <80 else 1e-4
    return lr


def resnet_layer(inputs,
                 num_filters=16,
                 kernel_size=3,
                 strides=1,
                 activation='relu',
                 batch_normalization=True,
                 conv_first=True):

    conv = Conv2D(num_filters,
                  kernel_size=kernel_size,
                  strides=strides,
                  padding='same',
                  kernel_initializer='he_normal',
                  kernel_regularizer=l2(1e-4))

    x = inputs
```

```
    if conv_first:
        x = conv(x)
        if batch_normalization:
            x = BatchNormalization()(x)
        if activation is not None:
            x = Activation(activation)(x)
    else:
        if batch_normalization:
            x = BatchNormalization()(x)
        if activation is not None:
            x = Activation(activation)(x)
        x = conv(x)
    return x


def resnet_v1(input_shape, depth, num_classes=10):

    if (depth - 2) % 6 != 0:
        raise ValueError('depth should be 6n+2 (eg 20, 32, in [a])')
    num_filters = 16
    num_res_blocks = int((depth - 2) / 6)

    inputs = Input(shape=input_shape)
    x = resnet_layer(inputs=inputs)
    for stack in range(3):
        for res_block in range(num_res_blocks):
            strides = 1
            # first layer but not first stack
            if stack > 0 and res_block == 0:
                strides = 2  # downsample
            y = resnet_layer(inputs=x,
                             num_filters=num_filters,
                             strides=strides)
            y = resnet_layer(inputs=y,
                             num_filters=num_filters,
                             activation=None)
            # first layer but not first stack
            if stack > 0 and res_block == 0:
                # linear projection residual shortcut
                # connection to match changed dims
                x = resnet_layer(inputs=x,
                                 num_filters=num_filters,
                                 kernel_size=1,
                                 strides=strides,
                                 activation=None,
                                 batch_normalization=False)
            x = add([x, y])
            x = Activation('relu')(x)
        num_filters *= 2

    # add classifier on top.
    # v1 does not use BN after last shortcut connection-ReLU
    x = AveragePooling2D(pool_size=8)(x)
    y = Flatten()(x)
    outputs = Dense(num_classes,
                    activation='softmax',
                    kernel_initializer='he_normal')(y)
```

```
    # instantiate model.
    model = Model(inputs=inputs, outputs=outputs)
    return model



model = resnet_v1(input_shape=input_shape, depth=depth)

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


# prepare model model saving directory.
save_dir = os.path.join(os.getcwd(), 'saved_models')
model_name = 'cifar10_%s_model.{epoch:03d}.h5' % model_type
if not os.path.isdir(save_dir):
    os.makedirs(save_dir)
filepath = os.path.join(save_dir, model_name)

# prepare callbacks for model saving and for learning rate adjustment.
checkpoint = ModelCheckpoint(filepath=filepath,
                             monitor='val_acc',
                             verbose=1,
                             save_best_only=True)

lr_scheduler = LearningRateScheduler(lr_schedule)

lr_reducer = ReduceLROnPlateau(factor=np.sqrt(0.1),
                               cooldown=0,
                               patience=5,
                               min_lr=0.5e-6)

callbacks = [checkpoint, lr_reducer, lr_scheduler]

# run training, with or without data augmentation.

print('With data augmentation')
datagen = ImageDataGenerator(featurewise_center=False,samplewise_center=
False,featurewise_std_normalization=False,
                             samplewise_std_normalization=False,zca_whit
ening=False,rotation_range=0,width_shift_range=0.1,height_shift_range=0.
1,horizontal_flip=True,vertical_flip=False)

datagen.fit(x_train)

steps_per_epoch =  math.ceil(len(x_train) / batch_size)
# fit the model on the batches generated by datagen.flow().
history = model.fit(x=datagen.flow(x_train, y_train, batch_size=batch_si
ze),
          verbose=1,
          epochs=epochs,
          validation_data=(x_test, y_test),  ### Testing is done with or
iginal images.
          steps_per_epoch=steps_per_epoch,
```

```
          callbacks=callbacks)


# score trained model
scores = model.evaluate(x_test,
                        y_test,
                        batch_size=batch_size,
                        verbose=0)
print('Validation loss:', scores[0])
print('Test accuracy:', scores[1])
```

```
Model: "functional_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connec
ted to
==================================================================================================
input_1 (InputLayer)            [(None, 32, 32, 3)]  0
__________________________________________________________________________________________________
conv2d (Conv2D)                 (None, 32, 32, 16)   448         input_
1[0][0]
__________________________________________________________________________________________________
batch_normalization (BatchNorma (None, 32, 32, 16)   64          conv2d
[0][0]
__________________________________________________________________________________________________
activation (Activation)         (None, 32, 32, 16)   0           batch_
normalization[0][0]
__________________________________________________________________________________________________
conv2d_1 (Conv2D)               (None, 32, 32, 16)   2320        activa
tion[0][0]
__________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, 32, 32, 16)   64          conv2d
_1[0][0]
__________________________________________________________________________________________________
activation_1 (Activation)       (None, 32, 32, 16)   0           batch_
normalization_1[0][0]
__________________________________________________________________________________________________
conv2d_2 (Conv2D)               (None, 32, 32, 16)   2320        activa
tion_1[0][0]
__________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, 32, 32, 16)   64          conv2d
_2[0][0]
__________________________________________________________________________________________________
add (Add)                       (None, 32, 32, 16)   0           activa
tion[0][0]
                                                                 batch_
normalization_2[0][0]
__________________________________________________________________________________________________
activation_2 (Activation)       (None, 32, 32, 16)   0           add[0]
[0]
__________________________________________________________________________________________________
conv2d_3 (Conv2D)               (None, 32, 32, 16)   2320        activa
tion_2[0][0]
__________________________________________________________________________________________________
batch_normalization_3 (BatchNor (None, 32, 32, 16)   64          conv2d
```

```
_3[0][0]
__________________________________________________________________________________________
activation_3 (Activation)       (None, 32, 32, 16)   0           batch_
normalization_3[0][0]
__________________________________________________________________________________________
conv2d_4 (Conv2D)               (None, 32, 32, 16)   2320        activa
tion_3[0][0]
__________________________________________________________________________________________
batch_normalization_4 (BatchNor (None, 32, 32, 16)   64          conv2d
_4[0][0]
__________________________________________________________________________________________
add_1 (Add)                     (None, 32, 32, 16)   0           activa
tion_2[0][0]
                                                                 batch_
normalization_4[0][0]
__________________________________________________________________________________________
activation_4 (Activation)       (None, 32, 32, 16)   0           add_1
[0][0]
__________________________________________________________________________________________
conv2d_5 (Conv2D)               (None, 32, 32, 16)   2320        activa
tion_4[0][0]
__________________________________________________________________________________________
batch_normalization_5 (BatchNor (None, 32, 32, 16)   64          conv2d
_5[0][0]
__________________________________________________________________________________________
activation_5 (Activation)       (None, 32, 32, 16)   0           batch_
normalization_5[0][0]
__________________________________________________________________________________________
conv2d_6 (Conv2D)               (None, 32, 32, 16)   2320        activa
tion_5[0][0]
__________________________________________________________________________________________
batch_normalization_6 (BatchNor (None, 32, 32, 16)   64          conv2d
_6[0][0]
__________________________________________________________________________________________
add_2 (Add)                     (None, 32, 32, 16)   0           activa
tion_4[0][0]
                                                                 batch_
normalization_6[0][0]
__________________________________________________________________________________________
activation_6 (Activation)       (None, 32, 32, 16)   0           add_2
[0][0]
__________________________________________________________________________________________
conv2d_7 (Conv2D)               (None, 32, 32, 16)   2320        activa
tion_6[0][0]
```

```
________________________________________________________________________________________
__________________________
batch_normalization_7 (BatchNor (None, 32, 32, 16)   64          conv2d
_7[0][0]
________________________________________________________________________________________
__________________________
activation_7 (Activation)       (None, 32, 32, 16)   0           batch_
normalization_7[0][0]
________________________________________________________________________________________
__________________________
conv2d_8 (Conv2D)               (None, 32, 32, 16)   2320        activa
tion_7[0][0]
________________________________________________________________________________________
__________________________
batch_normalization_8 (BatchNor (None, 32, 32, 16)   64          conv2d
_8[0][0]
________________________________________________________________________________________
__________________________
add_3 (Add)                     (None, 32, 32, 16)   0           activa
tion_6[0][0]
                                                                 batch_
normalization_8[0][0]
________________________________________________________________________________________
__________________________
activation_8 (Activation)       (None, 32, 32, 16)   0           add_3
[0][0]
________________________________________________________________________________________
__________________________
conv2d_9 (Conv2D)               (None, 32, 32, 16)   2320        activa
tion_8[0][0]
________________________________________________________________________________________
__________________________
batch_normalization_9 (BatchNor (None, 32, 32, 16)   64          conv2d
_9[0][0]
________________________________________________________________________________________
__________________________
activation_9 (Activation)       (None, 32, 32, 16)   0           batch_
normalization_9[0][0]
________________________________________________________________________________________
__________________________
conv2d_10 (Conv2D)              (None, 32, 32, 16)   2320        activa
tion_9[0][0]
________________________________________________________________________________________
__________________________
batch_normalization_10 (BatchNo (None, 32, 32, 16)   64          conv2d
_10[0][0]
________________________________________________________________________________________
__________________________
add_4 (Add)                     (None, 32, 32, 16)   0           activa
tion_8[0][0]
                                                                 batch_
normalization_10[0][0]
________________________________________________________________________________________
__________________________
activation_10 (Activation)      (None, 32, 32, 16)   0           add_4
[0][0]
________________________________________________________________________________________
```

```
________________________________________________________________________________
conv2d_11 (Conv2D)              (None, 32, 32, 16)   2320        activa
tion_10[0][0]
________________________________________________________________________________
batch_normalization_11 (BatchNo (None, 32, 32, 16)   64          conv2d
_11[0][0]
________________________________________________________________________________
activation_11 (Activation)      (None, 32, 32, 16)   0           batch_
normalization_11[0][0]
________________________________________________________________________________
conv2d_12 (Conv2D)              (None, 32, 32, 16)   2320        activa
tion_11[0][0]
________________________________________________________________________________
batch_normalization_12 (BatchNo (None, 32, 32, 16)   64          conv2d
_12[0][0]
________________________________________________________________________________
add_5 (Add)                     (None, 32, 32, 16)   0           activa
tion_10[0][0]
                                                                 batch_
normalization_12[0][0]
________________________________________________________________________________
activation_12 (Activation)      (None, 32, 32, 16)   0           add_5
[0][0]
________________________________________________________________________________
conv2d_13 (Conv2D)              (None, 32, 32, 16)   2320        activa
tion_12[0][0]
________________________________________________________________________________
batch_normalization_13 (BatchNo (None, 32, 32, 16)   64          conv2d
_13[0][0]
________________________________________________________________________________
activation_13 (Activation)      (None, 32, 32, 16)   0           batch_
normalization_13[0][0]
________________________________________________________________________________
conv2d_14 (Conv2D)              (None, 32, 32, 16)   2320        activa
tion_13[0][0]
________________________________________________________________________________
batch_normalization_14 (BatchNo (None, 32, 32, 16)   64          conv2d
_14[0][0]
________________________________________________________________________________
add_6 (Add)                     (None, 32, 32, 16)   0           activa
tion_12[0][0]
                                                                 batch_
normalization_14[0][0]
________________________________________________________________________________
```

```
activation_14 (Activation)      (None, 32, 32, 16)   0           add_6
[0][0]
__________________________________________________________________________________________________
conv2d_15 (Conv2D)              (None, 16, 16, 32)   4640        activa
tion_14[0][0]
__________________________________________________________________________________________________
batch_normalization_15 (BatchNo (None, 16, 16, 32)   128         conv2d
_15[0][0]
__________________________________________________________________________________________________
activation_15 (Activation)      (None, 16, 16, 32)   0           batch_
normalization_15[0][0]
__________________________________________________________________________________________________
conv2d_16 (Conv2D)              (None, 16, 16, 32)   9248        activa
tion_15[0][0]
__________________________________________________________________________________________________
conv2d_17 (Conv2D)              (None, 16, 16, 32)   544         activa
tion_14[0][0]
__________________________________________________________________________________________________
batch_normalization_16 (BatchNo (None, 16, 16, 32)   128         conv2d
_16[0][0]
__________________________________________________________________________________________________
add_7 (Add)                     (None, 16, 16, 32)   0           conv2d
_17[0][0]
                                                                 batch_
normalization_16[0][0]
__________________________________________________________________________________________________
activation_16 (Activation)      (None, 16, 16, 32)   0           add_7
[0][0]
__________________________________________________________________________________________________
conv2d_18 (Conv2D)              (None, 16, 16, 32)   9248        activa
tion_16[0][0]
__________________________________________________________________________________________________
batch_normalization_17 (BatchNo (None, 16, 16, 32)   128         conv2d
_18[0][0]
__________________________________________________________________________________________________
activation_17 (Activation)      (None, 16, 16, 32)   0           batch_
normalization_17[0][0]
__________________________________________________________________________________________________
conv2d_19 (Conv2D)              (None, 16, 16, 32)   9248        activa
tion_17[0][0]
__________________________________________________________________________________________________
batch_normalization_18 (BatchNo (None, 16, 16, 32)   128         conv2d
_19[0][0]
__________________________________________________________________________________________________
```

```
add_8 (Add)                     (None, 16, 16, 32)   0           activa
tion_16[0][0]
                                                                 batch_
normalization_18[0][0]
__________________________________________________________________________________________________
activation_18 (Activation)      (None, 16, 16, 32)   0           add_8
[0][0]
__________________________________________________________________________________________________
conv2d_20 (Conv2D)              (None, 16, 16, 32)   9248        activa
tion_18[0][0]
__________________________________________________________________________________________________
batch_normalization_19 (BatchNo (None, 16, 16, 32)   128         conv2d
_20[0][0]
__________________________________________________________________________________________________
activation_19 (Activation)      (None, 16, 16, 32)   0           batch_
normalization_19[0][0]
__________________________________________________________________________________________________
conv2d_21 (Conv2D)              (None, 16, 16, 32)   9248        activa
tion_19[0][0]
__________________________________________________________________________________________________
batch_normalization_20 (BatchNo (None, 16, 16, 32)   128         conv2d
_21[0][0]
__________________________________________________________________________________________________
add_9 (Add)                     (None, 16, 16, 32)   0           activa
tion_18[0][0]
                                                                 batch_
normalization_20[0][0]
__________________________________________________________________________________________________
activation_20 (Activation)      (None, 16, 16, 32)   0           add_9
[0][0]
__________________________________________________________________________________________________
conv2d_22 (Conv2D)              (None, 16, 16, 32)   9248        activa
tion_20[0][0]
__________________________________________________________________________________________________
batch_normalization_21 (BatchNo (None, 16, 16, 32)   128         conv2d
_22[0][0]
__________________________________________________________________________________________________
activation_21 (Activation)      (None, 16, 16, 32)   0           batch_
normalization_21[0][0]
__________________________________________________________________________________________________
conv2d_23 (Conv2D)              (None, 16, 16, 32)   9248        activa
tion_21[0][0]
__________________________________________________________________________________________________
```

```
batch_normalization_22 (BatchNo (None, 16, 16, 32)   128         conv2d
_23[0][0]
__________________________________________________________________________________________________
add_10 (Add)                    (None, 16, 16, 32)   0           activa
tion_20[0][0]
                                                                 batch_
normalization_22[0][0]
__________________________________________________________________________________________________
activation_22 (Activation)      (None, 16, 16, 32)   0           add_10
[0][0]
__________________________________________________________________________________________________
conv2d_24 (Conv2D)              (None, 16, 16, 32)   9248        activa
tion_22[0][0]
__________________________________________________________________________________________________
batch_normalization_23 (BatchNo (None, 16, 16, 32)   128         conv2d
_24[0][0]
__________________________________________________________________________________________________
activation_23 (Activation)      (None, 16, 16, 32)   0           batch_
normalization_23[0][0]
__________________________________________________________________________________________________
conv2d_25 (Conv2D)              (None, 16, 16, 32)   9248        activa
tion_23[0][0]
__________________________________________________________________________________________________
batch_normalization_24 (BatchNo (None, 16, 16, 32)   128         conv2d
_25[0][0]
__________________________________________________________________________________________________
add_11 (Add)                    (None, 16, 16, 32)   0           activa
tion_22[0][0]
                                                                 batch_
normalization_24[0][0]
__________________________________________________________________________________________________
activation_24 (Activation)      (None, 16, 16, 32)   0           add_11
[0][0]
__________________________________________________________________________________________________
conv2d_26 (Conv2D)              (None, 16, 16, 32)   9248        activa
tion_24[0][0]
__________________________________________________________________________________________________
batch_normalization_25 (BatchNo (None, 16, 16, 32)   128         conv2d
_26[0][0]
__________________________________________________________________________________________________
activation_25 (Activation)      (None, 16, 16, 32)   0           batch_
normalization_25[0][0]
__________________________________________________________________________________________________
conv2d_27 (Conv2D)              (None, 16, 16, 32)   9248        activa
```

```
tion_25[0][0]
__________________________________________________________________________________________________
batch_normalization_26 (BatchNo (None, 16, 16, 32)   128         conv2d_27[0][0]
__________________________________________________________________________________________________
add_12 (Add)                    (None, 16, 16, 32)   0           activation_24[0][0]
                                                                 batch_normalization_26[0][0]
__________________________________________________________________________________________________
activation_26 (Activation)      (None, 16, 16, 32)   0           add_12[0][0]
__________________________________________________________________________________________________
conv2d_28 (Conv2D)              (None, 16, 16, 32)   9248        activation_26[0][0]
__________________________________________________________________________________________________
batch_normalization_27 (BatchNo (None, 16, 16, 32)   128         conv2d_28[0][0]
__________________________________________________________________________________________________
activation_27 (Activation)      (None, 16, 16, 32)   0           batch_normalization_27[0][0]
__________________________________________________________________________________________________
conv2d_29 (Conv2D)              (None, 16, 16, 32)   9248        activation_27[0][0]
__________________________________________________________________________________________________
batch_normalization_28 (BatchNo (None, 16, 16, 32)   128         conv2d_29[0][0]
__________________________________________________________________________________________________
add_13 (Add)                    (None, 16, 16, 32)   0           activation_26[0][0]
                                                                 batch_normalization_28[0][0]
__________________________________________________________________________________________________
activation_28 (Activation)      (None, 16, 16, 32)   0           add_13[0][0]
__________________________________________________________________________________________________
conv2d_30 (Conv2D)              (None, 8, 8, 64)     18496       activation_28[0][0]
__________________________________________________________________________________________________
batch_normalization_29 (BatchNo (None, 8, 8, 64)     256         conv2d_30[0][0]
__________________________________________________________________________________________________
activation_29 (Activation)      (None, 8, 8, 64)     0           batch_normalization_29[0][0]
```

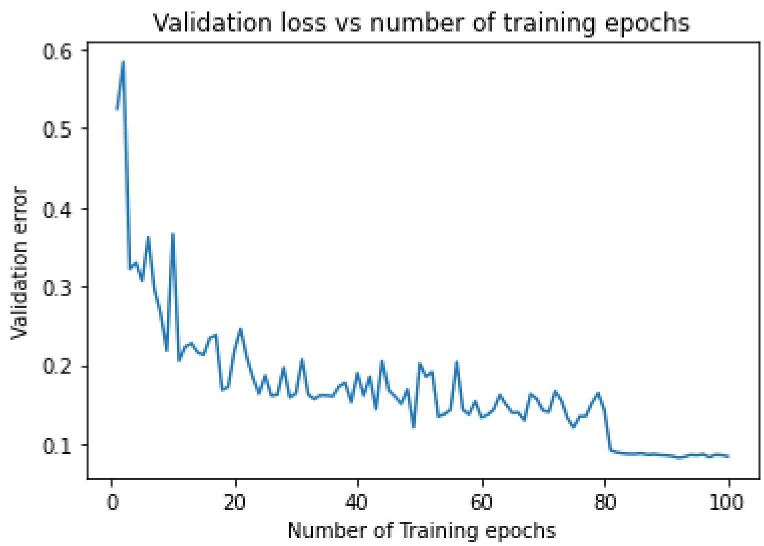
```
__________________________________________________________________________________________________
conv2d_31 (Conv2D)              (None, 8, 8, 64)     36928       activa
tion_29[0][0]
__________________________________________________________________________________________________
conv2d_32 (Conv2D)              (None, 8, 8, 64)     2112        activa
tion_28[0][0]
__________________________________________________________________________________________________
batch_normalization_30 (BatchNo (None, 8, 8, 64)     256         conv2d
_31[0][0]
__________________________________________________________________________________________________
add_14 (Add)                    (None, 8, 8, 64)     0           conv2d
_32[0][0]
                                                                 batch_
normalization_30[0][0]
__________________________________________________________________________________________________
activation_30 (Activation)      (None, 8, 8, 64)     0           add_14
[0][0]
__________________________________________________________________________________________________
conv2d_33 (Conv2D)              (None, 8, 8, 64)     36928       activa
tion_30[0][0]
__________________________________________________________________________________________________
batch_normalization_31 (BatchNo (None, 8, 8, 64)     256         conv2d
_33[0][0]
__________________________________________________________________________________________________
activation_31 (Activation)      (None, 8, 8, 64)     0           batch_
normalization_31[0][0]
__________________________________________________________________________________________________
conv2d_34 (Conv2D)              (None, 8, 8, 64)     36928       activa
tion_31[0][0]
__________________________________________________________________________________________________
batch_normalization_32 (BatchNo (None, 8, 8, 64)     256         conv2d
_34[0][0]
__________________________________________________________________________________________________
add_15 (Add)                    (None, 8, 8, 64)     0           activa
tion_30[0][0]
                                                                 batch_
normalization_32[0][0]
__________________________________________________________________________________________________
activation_32 (Activation)      (None, 8, 8, 64)     0           add_15
[0][0]
__________________________________________________________________________________________________
conv2d_35 (Conv2D)              (None, 8, 8, 64)     36928       activa
tion_32[0][0]
__________________________________________________________________________________________________
```

```
________________________________________________________________________________
batch_normalization_33 (BatchNo (None, 8, 8, 64)     256         conv2d
_35[0][0]

________________________________________________________________________________
activation_33 (Activation)      (None, 8, 8, 64)     0           batch_
normalization_33[0][0]

________________________________________________________________________________
conv2d_36 (Conv2D)              (None, 8, 8, 64)     36928       activa
tion_33[0][0]

________________________________________________________________________________
batch_normalization_34 (BatchNo (None, 8, 8, 64)     256         conv2d
_36[0][0]

________________________________________________________________________________
add_16 (Add)                    (None, 8, 8, 64)     0           activa
tion_32[0][0]
                                                                 batch_
normalization_34[0][0]

________________________________________________________________________________
activation_34 (Activation)      (None, 8, 8, 64)     0           add_16
[0][0]

________________________________________________________________________________
conv2d_37 (Conv2D)              (None, 8, 8, 64)     36928       activa
tion_34[0][0]

________________________________________________________________________________
batch_normalization_35 (BatchNo (None, 8, 8, 64)     256         conv2d
_37[0][0]

________________________________________________________________________________
activation_35 (Activation)      (None, 8, 8, 64)     0           batch_
normalization_35[0][0]

________________________________________________________________________________
conv2d_38 (Conv2D)              (None, 8, 8, 64)     36928       activa
tion_35[0][0]

________________________________________________________________________________
batch_normalization_36 (BatchNo (None, 8, 8, 64)     256         conv2d
_38[0][0]

________________________________________________________________________________
add_17 (Add)                    (None, 8, 8, 64)     0           activa
tion_34[0][0]
                                                                 batch_
normalization_36[0][0]

________________________________________________________________________________
activation_36 (Activation)      (None, 8, 8, 64)     0           add_17
[0][0]

________________________________________________________________________________
________________________________________________________________________________
```

```
conv2d_39 (Conv2D)              (None, 8, 8, 64)     36928       activa
tion_36[0][0]
__________________________________________________________________________________________________
batch_normalization_37 (BatchNo (None, 8, 8, 64)     256         conv2d
_39[0][0]
__________________________________________________________________________________________________
activation_37 (Activation)      (None, 8, 8, 64)     0           batch_
normalization_37[0][0]
__________________________________________________________________________________________________
conv2d_40 (Conv2D)              (None, 8, 8, 64)     36928       activa
tion_37[0][0]
__________________________________________________________________________________________________
batch_normalization_38 (BatchNo (None, 8, 8, 64)     256         conv2d
_40[0][0]
__________________________________________________________________________________________________
add_18 (Add)                    (None, 8, 8, 64)     0           activa
tion_36[0][0]
                                                                 batch_
normalization_38[0][0]
__________________________________________________________________________________________________
activation_38 (Activation)      (None, 8, 8, 64)     0           add_18
[0][0]
__________________________________________________________________________________________________
conv2d_41 (Conv2D)              (None, 8, 8, 64)     36928       activa
tion_38[0][0]
__________________________________________________________________________________________________
batch_normalization_39 (BatchNo (None, 8, 8, 64)     256         conv2d
_41[0][0]
__________________________________________________________________________________________________
activation_39 (Activation)      (None, 8, 8, 64)     0           batch_
normalization_39[0][0]
__________________________________________________________________________________________________
conv2d_42 (Conv2D)              (None, 8, 8, 64)     36928       activa
tion_39[0][0]
__________________________________________________________________________________________________
batch_normalization_40 (BatchNo (None, 8, 8, 64)     256         conv2d
_42[0][0]
__________________________________________________________________________________________________
add_19 (Add)                    (None, 8, 8, 64)     0           activa
tion_38[0][0]
                                                                 batch_
normalization_40[0][0]
__________________________________________________________________________________________________
activation_40 (Activation)      (None, 8, 8, 64)     0           add_19
```

```
[0][0]
__________________________________________________________________________________________________
conv2d_43 (Conv2D)              (None, 8, 8, 64)     36928       activa
tion_40[0][0]
__________________________________________________________________________________________________
batch_normalization_41 (BatchNo (None, 8, 8, 64)     256         conv2d
_43[0][0]
__________________________________________________________________________________________________
activation_41 (Activation)      (None, 8, 8, 64)     0           batch_
normalization_41[0][0]
__________________________________________________________________________________________________
conv2d_44 (Conv2D)              (None, 8, 8, 64)     36928       activa
tion_41[0][0]
__________________________________________________________________________________________________
batch_normalization_42 (BatchNo (None, 8, 8, 64)     256         conv2d
_44[0][0]
__________________________________________________________________________________________________
add_20 (Add)                    (None, 8, 8, 64)     0           activa
tion_40[0][0]
                                                                 batch_
normalization_42[0][0]
__________________________________________________________________________________________________
activation_42 (Activation)      (None, 8, 8, 64)     0           add_20
[0][0]
__________________________________________________________________________________________________
average_pooling2d (AveragePooli (None, 1, 1, 64)     0           activa
tion_42[0][0]
__________________________________________________________________________________________________
flatten (Flatten)               (None, 64)           0           averag
e_pooling2d[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 10)           650         flatte
n[0][0]
==================================================================================================
Total params: 665,994
Trainable params: 662,826
Non-trainable params: 3,168
__________________________________________________________________________________________________
With data augmentation
```

In [ ]:
```
plt.plot(range(1,101),history.history['val_loss'])[1-el for el in l]
plt.plot(range(1,101),[1-val_acc for val_acc in history.history['val_ac
c']])
plt.xlabel("Number of Training epochs")
plt.ylabel("Validation error")
plt.title("Validation loss vs number of training epochs")
plt.savefig("val_error_without_cutout.png")
plt.show()
```



Q3

In [ ]:
```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

In [ ]:
```
def batch_generator(x, y, epochs, m, batch_size, augment=None):
    for _ in range(epochs):
        n = x.shape[0]
        reorder = np.random.permutation(n)
        cursor = 0
        while cursor + batch_size < x.shape[0]:
            x_batch = x[reorder[cursor:cursor+batch_size]]
            y_batch = y[reorder[cursor:cursor+batch_size]]
            if augment != None:
                yield np.array([augment(xx) for xx in x_batch for rep in
range(m)]), np.array([yy for yy in y_batch for rep in range(m)])
            else:
                yield x_batch, y_batch
            cursor += batch_size
```

In [ ]:
```
val_acc_cutout_32 = []
epochs = 50
durations_32 = []
for m in [32]:#,4,8,16,32]:
    print("M = ",m)
    model = resnet_v1(
        input_shape=x_train.shape[1:],
        depth=44
    )
    model.compile(
        loss='categorical_crossentropy',
        optimizer=tf.keras.optimizers.RMSprop(),
        metrics=['accuracy']
    )
    duration = time()
    hist = model.fit_generator(
        batch_generator(
            x_train,
            y_train,
            m=m,
            batch_size=64,
            epochs=epochs,
            augment=apply_mask
        ),
        epochs=epochs,
        validation_data=(x_test,y_test),
        steps_per_epoch=np.floor(x_train.shape[0]/64.0),
        verbose=1,
        callbacks=[lr_scheduler]
    )
    durations_32.append(time()-duration)
    val_acc_cutout_32.append(hist.history['val_accuracy'])
    savetxt(F"/content/gdrive/My Drive/val_acc_cutout_32.csv", val_acc_c
utout_32, delimiter=',')
    savetxt(F"/content/gdrive/My Drive/durations_32.csv", durations_32,
delimiter=',')
```

```
M =  32
WARNING:tensorflow:From <ipython-input-8-fc7985454fd1>:29: Model.fit_ge
nerator (from tensorflow.python.keras.engine.training) is deprecated an
d will be removed in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.
Epoch 1/50
781/781 [==============================] - 829s 1s/step - loss: 2.4523
- accuracy: 0.1838 - val_loss: 8.8941 - val_accuracy: 0.1322
Epoch 2/50
781/781 [==============================] - 830s 1s/step - loss: 2.2384
- accuracy: 0.2145 - val_loss: 4.9183 - val_accuracy: 0.1734
Epoch 3/50
781/781 [==============================] - 829s 1s/step - loss: 2.1764
- accuracy: 0.2276 - val_loss: 6.6778 - val_accuracy: 0.2332
Epoch 4/50
781/781 [==============================] - 829s 1s/step - loss: 2.1453
- accuracy: 0.2322 - val_loss: 2.5713 - val_accuracy: 0.2711
Epoch 5/50
781/781 [==============================] - 829s 1s/step - loss: 2.1269
- accuracy: 0.2397 - val_loss: 2.1092 - val_accuracy: 0.3123
Epoch 6/50
781/781 [==============================] - 828s 1s/step - loss: 2.1124
- accuracy: 0.2445 - val_loss: 2.4620 - val_accuracy: 0.3150
Epoch 7/50
781/781 [==============================] - 828s 1s/step - loss: 2.0997
- accuracy: 0.2490 - val_loss: 2.3577 - val_accuracy: 0.2605
Epoch 8/50
781/781 [==============================] - 827s 1s/step - loss: 2.0891
- accuracy: 0.2493 - val_loss: 5.9087 - val_accuracy: 0.2397
Epoch 9/50
781/781 [==============================] - 826s 1s/step - loss: 2.0797
- accuracy: 0.2545 - val_loss: 2.0752 - val_accuracy: 0.3292
Epoch 10/50
781/781 [==============================] - 825s 1s/step - loss: 2.0687
- accuracy: 0.2575 - val_loss: 2.8031 - val_accuracy: 0.3276
Epoch 11/50
781/781 [==============================] - 825s 1s/step - loss: 2.0631
- accuracy: 0.2596 - val_loss: 2.5918 - val_accuracy: 0.3336
Epoch 12/50
781/781 [==============================] - 824s 1s/step - loss: 2.0532
- accuracy: 0.2662 - val_loss: 2.5565 - val_accuracy: 0.2961
Epoch 13/50
781/781 [==============================] - 824s 1s/step - loss: 2.0461
- accuracy: 0.2653 - val_loss: 2.6348 - val_accuracy: 0.2966
Epoch 14/50
781/781 [==============================] - 824s 1s/step - loss: 2.0459
- accuracy: 0.2670 - val_loss: 2.5302 - val_accuracy: 0.2901
Epoch 15/50
781/781 [==============================] - 823s 1s/step - loss: 2.0346
- accuracy: 0.2690 - val_loss: 1.9890 - val_accuracy: 0.3520
Epoch 16/50
781/781 [==============================] - 823s 1s/step - loss: 2.0318
- accuracy: 0.2731 - val_loss: 1.9361 - val_accuracy: 0.3740
Epoch 17/50
781/781 [==============================] - 822s 1s/step - loss: 2.0247
- accuracy: 0.2735 - val_loss: 2.2907 - val_accuracy: 0.3718
```

```
Epoch 18/50
781/781 [==============================] - 822s 1s/step - loss: 2.0208
- accuracy: 0.2789 - val_loss: 1.9534 - val_accuracy: 0.3929
Epoch 19/50
781/781 [==============================] - 822s 1s/step - loss: 2.0144
- accuracy: 0.2809 - val_loss: 2.1458 - val_accuracy: 0.3469
Epoch 20/50
781/781 [==============================] - 821s 1s/step - loss: 2.0109
- accuracy: 0.2830 - val_loss: 2.5252 - val_accuracy: 0.3701
Epoch 21/50
781/781 [==============================] - 820s 1s/step - loss: 2.0052
- accuracy: 0.2815 - val_loss: 2.0140 - val_accuracy: 0.3871
Epoch 22/50
781/781 [==============================] - 820s 1s/step - loss: 2.0030
- accuracy: 0.2841 - val_loss: 2.1059 - val_accuracy: 0.3527
Epoch 23/50
781/781 [==============================] - 819s 1s/step - loss: 2.0019
- accuracy: 0.2860 - val_loss: 1.9128 - val_accuracy: 0.4005
Epoch 24/50
781/781 [==============================] - 819s 1s/step - loss: 1.9959
- accuracy: 0.2885 - val_loss: 1.9944 - val_accuracy: 0.3907
Epoch 25/50
781/781 [==============================] - 819s 1s/step - loss: 1.9912
- accuracy: 0.2949 - val_loss: 1.9979 - val_accuracy: 0.3631
Epoch 26/50
781/781 [==============================] - 819s 1s/step - loss: 1.9926
- accuracy: 0.2893 - val_loss: 1.9422 - val_accuracy: 0.4047
Epoch 27/50
781/781 [==============================] - 819s 1s/step - loss: 1.9837
- accuracy: 0.2917 - val_loss: 2.8312 - val_accuracy: 0.3334
Epoch 28/50
781/781 [==============================] - 819s 1s/step - loss: 1.9794
- accuracy: 0.2952 - val_loss: 2.4772 - val_accuracy: 0.3163
Epoch 29/50
781/781 [==============================] - 818s 1s/step - loss: 1.9823
- accuracy: 0.2914 - val_loss: 2.1226 - val_accuracy: 0.3424
Epoch 30/50
781/781 [==============================] - 817s 1s/step - loss: 1.9852
- accuracy: 0.2925 - val_loss: 2.5434 - val_accuracy: 0.3065
Epoch 31/50
781/781 [==============================] - 818s 1s/step - loss: 1.9797
- accuracy: 0.2943 - val_loss: 2.1222 - val_accuracy: 0.3573
Epoch 32/50
781/781 [==============================] - 817s 1s/step - loss: 1.9781
- accuracy: 0.2923 - val_loss: 1.8393 - val_accuracy: 0.4118
Epoch 33/50
781/781 [==============================] - 817s 1s/step - loss: 1.9791
- accuracy: 0.2973 - val_loss: 1.9869 - val_accuracy: 0.3959
Epoch 34/50
781/781 [==============================] - 818s 1s/step - loss: 1.9749
- accuracy: 0.2982 - val_loss: 1.7663 - val_accuracy: 0.4138
Epoch 35/50
781/781 [==============================] - 817s 1s/step - loss: 1.9733
- accuracy: 0.2968 - val_loss: 3.0175 - val_accuracy: 0.3076
Epoch 36/50
781/781 [==============================] - 817s 1s/step - loss: 1.9675
- accuracy: 0.2999 - val_loss: 2.0879 - val_accuracy: 0.3621
```

```
Epoch 37/50
781/781 [==============================] - 816s 1s/step - loss: 1.9696
- accuracy: 0.2984 - val_loss: 1.8978 - val_accuracy: 0.3913
Epoch 38/50
781/781 [==============================] - 816s 1s/step - loss: 1.9722
- accuracy: 0.2989 - val_loss: 2.2136 - val_accuracy: 0.3519
Epoch 39/50
781/781 [==============================] - 816s 1s/step - loss: 1.9663
- accuracy: 0.3034 - val_loss: 2.0892 - val_accuracy: 0.3659
Epoch 40/50
781/781 [==============================] - 815s 1s/step - loss: 1.9643
- accuracy: 0.3006 - val_loss: 2.0164 - val_accuracy: 0.3960
Epoch 41/50
781/781 [==============================] - 816s 1s/step - loss: 1.9652
- accuracy: 0.2995 - val_loss: 1.9512 - val_accuracy: 0.3843
Epoch 42/50
781/781 [==============================] - 815s 1s/step - loss: 1.9642
- accuracy: 0.3009 - val_loss: 2.1837 - val_accuracy: 0.3738
Epoch 43/50
781/781 [==============================] - 816s 1s/step - loss: 1.9620
- accuracy: 0.3045 - val_loss: 1.9665 - val_accuracy: 0.4092
Epoch 44/50
626/781 [=======================>......] - ETA: 2:41 - loss: 1.9645 - a
ccuracy: 0.3018
```

In [ ]:
```
val_acc_cutout_16 = []
epochs = 50
durations_16 = []
for m in [16]:#,4,8,16,32]:
    print("M = ",m)
    model = resnet_v1(
        input_shape=x_train.shape[1:],
        depth=44
    )
    model.compile(
        loss='categorical_crossentropy',
        optimizer=tf.keras.optimizers.RMSprop(),
        metrics=['accuracy']
    )
    duration = time()
    hist = model.fit_generator(
        batch_generator(
            x_train,
            y_train,
            m=m,
            batch_size=64,
            epochs=epochs,
            augment=apply_mask
        ),
        epochs=epochs,
        validation_data=(x_test,y_test),
        steps_per_epoch=np.floor(x_train.shape[0]/64.0),
        verbose=1,
        callbacks=[lr_scheduler]
    )
    durations_16.append(time()-duration)
    val_acc_cutout_16.append(hist.history['val_accuracy'])
    savetxt(F"/content/gdrive/My Drive/val_acc_cutout_16.csv", val_acc_c
utout_16, delimiter=',')
    savetxt(F"/content/gdrive/My Drive/durations_16.csv", durations_16,
delimiter=',')
```

```
M =  16
Epoch 1/50
781/781 [==============================] - 351s 449ms/step - loss: 2.32
46 - accuracy: 0.2588 - val_loss: 2.2616 - val_accuracy: 0.3234
Epoch 2/50
781/781 [==============================] - 350s 449ms/step - loss: 2.04
90 - accuracy: 0.3229 - val_loss: 2.1283 - val_accuracy: 0.3616
Epoch 3/50
781/781 [==============================] - 350s 449ms/step - loss: 1.94
08 - accuracy: 0.3497 - val_loss: 2.0421 - val_accuracy: 0.3785
Epoch 4/50
781/781 [==============================] - 350s 448ms/step - loss: 1.86
49 - accuracy: 0.3729 - val_loss: 2.7860 - val_accuracy: 0.3220
Epoch 5/50
781/781 [==============================] - 350s 448ms/step - loss: 1.82
32 - accuracy: 0.3830 - val_loss: 2.3621 - val_accuracy: 0.3577
Epoch 6/50
781/781 [==============================] - 350s 448ms/step - loss: 1.78
54 - accuracy: 0.3984 - val_loss: 1.6221 - val_accuracy: 0.4752
Epoch 7/50
781/781 [==============================] - 350s 448ms/step - loss: 1.76
37 - accuracy: 0.4010 - val_loss: 1.6932 - val_accuracy: 0.4996
Epoch 8/50
781/781 [==============================] - 349s 447ms/step - loss: 1.73
94 - accuracy: 0.4123 - val_loss: 1.7524 - val_accuracy: 0.4478
Epoch 9/50
781/781 [==============================] - 349s 447ms/step - loss: 1.71
35 - accuracy: 0.4251 - val_loss: 1.6910 - val_accuracy: 0.5094
Epoch 10/50
781/781 [==============================] - 349s 447ms/step - loss: 1.70
26 - accuracy: 0.4257 - val_loss: 1.5577 - val_accuracy: 0.5254
Epoch 11/50
781/781 [==============================] - 348s 446ms/step - loss: 1.68
17 - accuracy: 0.4315 - val_loss: 1.8206 - val_accuracy: 0.4660
Epoch 12/50
781/781 [==============================] - 349s 446ms/step - loss: 1.66
57 - accuracy: 0.4375 - val_loss: 1.5431 - val_accuracy: 0.5336
Epoch 13/50
781/781 [==============================] - 349s 447ms/step - loss: 1.65
12 - accuracy: 0.4481 - val_loss: 1.4617 - val_accuracy: 0.5694
Epoch 14/50
781/781 [==============================] - 349s 446ms/step - loss: 1.63
95 - accuracy: 0.4476 - val_loss: 1.5714 - val_accuracy: 0.5230
Epoch 15/50
781/781 [==============================] - 348s 446ms/step - loss: 1.63
19 - accuracy: 0.4519 - val_loss: 1.5047 - val_accuracy: 0.5651
Epoch 16/50
781/781 [==============================] - 348s 446ms/step - loss: 1.61
73 - accuracy: 0.4575 - val_loss: 1.6193 - val_accuracy: 0.5502
Epoch 17/50
781/781 [==============================] - 348s 446ms/step - loss: 1.60
67 - accuracy: 0.4612 - val_loss: 1.5089 - val_accuracy: 0.5654
Epoch 18/50
781/781 [==============================] - 348s 445ms/step - loss: 1.60
18 - accuracy: 0.4649 - val_loss: 1.5367 - val_accuracy: 0.5813
Epoch 19/50
781/781 [==============================] - 348s 445ms/step - loss: 1.58
```

```
78 - accuracy: 0.4711 - val_loss: 1.6740 - val_accuracy: 0.5736
Epoch 20/50
781/781 [==============================] - 348s 445ms/step - loss: 1.58
46 - accuracy: 0.4719 - val_loss: 1.6035 - val_accuracy: 0.5435
Epoch 21/50
781/781 [==============================] - 348s 445ms/step - loss: 1.57
36 - accuracy: 0.4766 - val_loss: 1.1968 - val_accuracy: 0.6278
Epoch 22/50
781/781 [==============================] - 347s 445ms/step - loss: 1.56
89 - accuracy: 0.4775 - val_loss: 1.4297 - val_accuracy: 0.5711
Epoch 23/50
781/781 [==============================] - 347s 444ms/step - loss: 1.56
27 - accuracy: 0.4785 - val_loss: 1.7105 - val_accuracy: 0.5409
Epoch 24/50
781/781 [==============================] - 347s 444ms/step - loss: 1.55
19 - accuracy: 0.4854 - val_loss: 1.3900 - val_accuracy: 0.6067
Epoch 25/50
781/781 [==============================] - 347s 445ms/step - loss: 1.54
58 - accuracy: 0.4877 - val_loss: 1.9125 - val_accuracy: 0.4994
Epoch 26/50
781/781 [==============================] - 347s 444ms/step - loss: 1.54
20 - accuracy: 0.4887 - val_loss: 1.4997 - val_accuracy: 0.5893
Epoch 27/50
781/781 [==============================] - 347s 444ms/step - loss: 1.53
69 - accuracy: 0.4927 - val_loss: 1.4546 - val_accuracy: 0.5820
Epoch 28/50
781/781 [==============================] - 347s 444ms/step - loss: 1.53
09 - accuracy: 0.4955 - val_loss: 1.6628 - val_accuracy: 0.5456
Epoch 29/50
781/781 [==============================] - 347s 444ms/step - loss: 1.53
04 - accuracy: 0.4930 - val_loss: 1.2704 - val_accuracy: 0.6084
Epoch 30/50
781/781 [==============================] - 347s 444ms/step - loss: 1.52
04 - accuracy: 0.4970 - val_loss: 1.1398 - val_accuracy: 0.6452
Epoch 31/50
781/781 [==============================] - 347s 445ms/step - loss: 1.51
85 - accuracy: 0.4994 - val_loss: 1.5247 - val_accuracy: 0.6070
Epoch 32/50
781/781 [==============================] - 347s 444ms/step - loss: 1.51
22 - accuracy: 0.5016 - val_loss: 1.4050 - val_accuracy: 0.6199
Epoch 33/50
781/781 [==============================] - 346s 443ms/step - loss: 1.50
90 - accuracy: 0.5023 - val_loss: 1.3357 - val_accuracy: 0.6294
Epoch 34/50
781/781 [==============================] - 346s 443ms/step - loss: 1.50
56 - accuracy: 0.5081 - val_loss: 1.1745 - val_accuracy: 0.6539
Epoch 35/50
781/781 [==============================] - 346s 443ms/step - loss: 1.49
39 - accuracy: 0.5117 - val_loss: 1.3223 - val_accuracy: 0.6173
Epoch 36/50
781/781 [==============================] - 346s 443ms/step - loss: 1.49
47 - accuracy: 0.5100 - val_loss: 1.2596 - val_accuracy: 0.6382
Epoch 37/50
781/781 [==============================] - 346s 443ms/step - loss: 1.49
38 - accuracy: 0.5102 - val_loss: 1.2548 - val_accuracy: 0.6551
Epoch 38/50
781/781 [==============================] - 346s 443ms/step - loss: 1.48
```

```
31 - accuracy: 0.5154 - val_loss: 1.1847 - val_accuracy: 0.6446
Epoch 39/50
781/781 [==============================] - 345s 442ms/step - loss: 1.47
91 - accuracy: 0.5155 - val_loss: 1.4447 - val_accuracy: 0.5813
Epoch 40/50
781/781 [==============================] - 346s 443ms/step - loss: 1.47
77 - accuracy: 0.5169 - val_loss: 1.4787 - val_accuracy: 0.6207
Epoch 41/50
781/781 [==============================] - 346s 443ms/step - loss: 1.47
78 - accuracy: 0.5175 - val_loss: 1.3247 - val_accuracy: 0.6319
Epoch 42/50
781/781 [==============================] - 346s 443ms/step - loss: 1.47
60 - accuracy: 0.5188 - val_loss: 1.4403 - val_accuracy: 0.5875
Epoch 43/50
781/781 [==============================] - 347s 444ms/step - loss: 1.47
48 - accuracy: 0.5198 - val_loss: 1.4437 - val_accuracy: 0.6266
Epoch 44/50
781/781 [==============================] - 346s 443ms/step - loss: 1.46
64 - accuracy: 0.5228 - val_loss: 1.1758 - val_accuracy: 0.6656
Epoch 45/50
781/781 [==============================] - 346s 443ms/step - loss: 1.46
53 - accuracy: 0.5227 - val_loss: 1.6002 - val_accuracy: 0.5752
Epoch 46/50
781/781 [==============================] - 346s 443ms/step - loss: 1.45
52 - accuracy: 0.5282 - val_loss: 1.3545 - val_accuracy: 0.6496
Epoch 47/50
781/781 [==============================] - 346s 442ms/step - loss: 1.45
92 - accuracy: 0.5273 - val_loss: 1.1042 - val_accuracy: 0.6927
Epoch 48/50
781/781 [==============================] - 345s 441ms/step - loss: 1.45
77 - accuracy: 0.5270 - val_loss: 1.2462 - val_accuracy: 0.6345
Epoch 49/50
781/781 [==============================] - 345s 441ms/step - loss: 1.45
10 - accuracy: 0.5285 - val_loss: 1.0942 - val_accuracy: 0.6727
Epoch 50/50
781/781 [==============================] - 345s 441ms/step - loss: 1.45
32 - accuracy: 0.5278 - val_loss: 1.3246 - val_accuracy: 0.6311
```

In [14]:

```
cutout2_data = loadtxt("val_acc_cutout_2.csv",delimiter=",")
cutout4_data = loadtxt("val_acc_cutout_4.csv",delimiter=",")
cutout8_data = loadtxt("val_acc_cutout_8.csv",delimiter=",")
cutout16_data = loadtxt("val_acc_cutout_16.csv",delimiter=",")
cutout32_data = loadtxt("val_acc_cutout_32.csv",delimiter=",")

durations2_data = loadtxt("durations_2.csv",delimiter=",")
durations4_data = loadtxt("durations_4.csv",delimiter=",")
durations8_data = loadtxt("durations_8.csv",delimiter=",")
durations16_data = loadtxt("durations_16.csv",delimiter=",")
durations32_data = loadtxt("durations_32.csv",delimiter=",")
```

In [13]:
```
def val_error(val):
    return [1-v for v in val]

plt.plot(range(1,51),val_error(cutout2_data),"b-")
plt.plot(range(1,51),val_error(cutout4_data),"c-")
plt.plot(range(1,51),val_error(cutout8_data),"g-")
plt.plot(range(1,51),val_error(cutout16_data),"r-")
plt.plot(range(1,36),val_error(cutout32_data),"m-")
plt.xlabel("Number of training epochs")
plt.ylabel("Validation error")
plt.legend(["M=2","M=4","M=8","M=16","M=32"])
plt.plot(np.linspace(0,50,10000),[0.06]*10000,"k-")
plt.title("Validation error for M instances with cutout generated from e
ach input")
#plt.savefig("val_error_with_cutout.png")
plt.show()
```

In [20]:
```
print (" M=2 Time to train 50 epochs :", durations2_data,"\n")
print (" M=4 Time to train 50 epochs :", durations4_data,"\n")
print (" M=8 Time to train 50 epochs :", durations8_data,"\n")
print (" M=16 Time to train 50 epochs :", durations16_data,"\n")
print (" M=32 Time to train 35 epochs :", durations32_data,"\n")
```

```
 M=2 Time to train 50 epochs : 2766.2118268013

 M=4 Time to train 50 epochs : 4897.306997060776

 M=8 Time to train 50 epochs : 9115.753979206085

 M=16 Time to train 50 epochs : 17399.752579927444

 M=32 Time to train 35 epochs : 23586.50715637207
```

I could not train them for 100 epochs as I got runtime error on both colab and Jupytherhub but I trained them for 50 epochs for M=2 to 16 and 35 epochs for M=32.

In [ ]:

```

```