Load data and pre-processing

In [74]:

```
###Setting environment
import pandas as pd
import numpy as np
import sklearn
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
%matplotlib inline
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from tensorflow.keras.datasets import cifar100
from tensorflow.keras import regularizers
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.optimizers import Adam
import pickle
np.random.seed(123)
tf.random.set_seed(123)

import ssl

ssl._create_default_https_context = ssl._create_unverified_context
```

In [75]:

```
import tensorflow as tf
print("Num GPUs Available: ", len(tf.config.experimental.list_physical_devices('GPU')))
```

```
Num GPUs Available:  1
```

In [76]:

```
# transform labels to categorical
(x_train_original, y_train_original), (x_test, y_test) = cifar100.load_data()
y_train_original = keras.utils.to_categorical(y_train_original, 100)
y_test = keras.utils.to_categorical(y_test, 100)
```

In [77]:

```
print(x_train_original.shape)
print(y_train_original.shape)
print(x_test.shape)
print(y_test.shape)
```

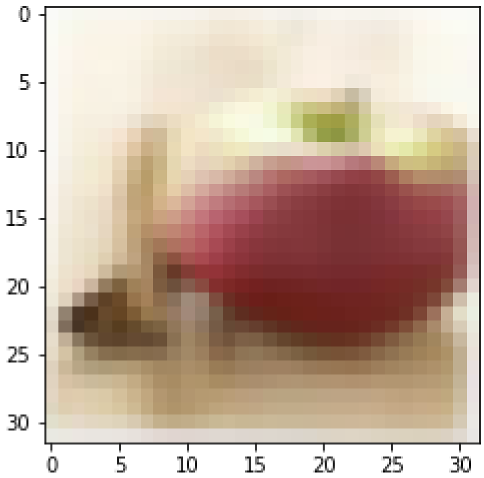
```
(50000, 32, 32, 3)
(50000, 100)
(10000, 32, 32, 3)
(10000, 100)
```

In [78]:

```
plt.imshow(x_train_original[2])
```

Out[78]:

<matplotlib.image.AxesImage at 0x1df1f640940>



split data

In [79]:

```
# split data into val and train
x_train, x_val, y_train, y_val = train_test_split(x_train_original, y_train_original, test_size
= 0.1)
```

Image augmentation

In [80]:

```
# define image augmentation
img_gen = ImageDataGenerator(
    rotation_range=30,
    width_shift_range=0.2,#width shift
    height_shift_range=0.2,
    horizontal_flip=True,
)

img_gen.fit(x_train)
```

Function to detect training

In [81]:

```
# plot the history to detect training
def plot_history(hist):
    plt.figure(figsize=(15,6))
    plt.subplot(1,2,1)
    plt.xlabel('Epoch')
    plt.ylabel('Loss')
    plt.plot(hist['loss'],
             label='loss')
    plt.plot(hist['val_loss'],
             label='val_loss')
    plt.legend()
    plt.subplot(1,2,2)
    plt.xlabel('Epoch')
    plt.ylabel('Accuracy')
    plt.plot( hist['accuracy'],
             label = 'accuracy',color='red')
    plt.plot( hist['val_accuracy'],
             label = 'val_accuracy',color='green')
    plt.legend()
```

Structure of cnn

In [82]:

```
# a structure which is very similar to VGG16 but add 2 conv, 2 activation, 2 batch normalizatio
n, 2 pooling, 2 dropout layers,
# and change the final dropout from 0.5 to 0.3
weight_decay = 0.0005
nb_epoch=100
batch_size=20
#layer1
model = Sequential()
model.add(layers.Conv2D(64, (3, 3), padding='same',strides=(1, 1),
input_shape=(32,32,3),kernel_regularizer=regularizers.l2(weight_decay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.3))

#layer2
model.add(layers.Conv2D(64, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_dec
ay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.MaxPooling2D(pool_size=(2, 2),strides=(2,2),padding='same'))

#layer3
model.add(layers.Conv2D(128, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_de
cay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.4))

#layer4
model.add(layers.Conv2D(128, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_de
cay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.MaxPooling2D(pool_size=(2, 2)))

#layer5
model.add(layers.Conv2D(256, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_de
cay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.4))

#layer6
model.add(layers.Conv2D(256, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_de
cay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.4))

#layer7
model.add(layers.Conv2D(256, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_de
cay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.MaxPooling2D(pool_size=(2, 2)))

#layer8
model.add(layers.Conv2D(512, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_de
cay)))
```

```
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.4))

#layer9
model.add(layers.Conv2D(512, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_de
cay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.4))

#layer10
model.add(layers.Conv2D(512, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_de
cay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.MaxPooling2D(pool_size=(2, 2)))

#layer11
model.add(layers.Conv2D(512, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_de
cay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.4))

#layer12
model.add(layers.Conv2D(512, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_de
cay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.4))

# Extra to VGG16
# layer13
model.add(layers.Conv2D(512, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_de
cay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.4))

# Extra to VGG16
# layer14

model.add(layers.Conv2D(512, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_de
cay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.3))

#layer15
model.add(layers.Conv2D(512, (3, 3), padding='same',kernel_regularizer=regularizers.l2(weight_de
cay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())
model.add(layers.MaxPooling2D(pool_size=(2, 2)))
model.add(layers.Dropout(0.5))

#layer16
model.add(layers.Flatten())
model.add(layers.Dense(512,kernel_regularizer=regularizers.l2(weight_decay)))
model.add(layers.Activation('relu'))
```

```
model.add(layers.BatchNormalization())

#layer17
model.add(layers.Dense(512,kernel_regularizer=regularizers.l2(weight_decay)))
model.add(layers.Activation('relu'))
model.add(layers.BatchNormalization())

#layer18
model.add(layers.Dropout(0.3)) # from 0.5 to 0.3
model.add(layers.Dense(100,activation='softmax'))

model.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_15 (Conv2D)	(None, 32, 32, 64)	1792
activation_17 (Activation)	(None, 32, 32, 64)	0
batch_normalization_17 (Batc	(None, 32, 32, 64)	256
dropout_12 (Dropout)	(None, 32, 32, 64)	0
conv2d_16 (Conv2D)	(None, 32, 32, 64)	36928
activation_18 (Activation)	(None, 32, 32, 64)	0
batch_normalization_18 (Batc	(None, 32, 32, 64)	256
max_pooling2d_5 (MaxPooling2	(None, 16, 16, 64)	0
conv2d_17 (Conv2D)	(None, 16, 16, 128)	73856
activation_19 (Activation)	(None, 16, 16, 128)	0
batch_normalization_19 (Batc	(None, 16, 16, 128)	512
dropout_13 (Dropout)	(None, 16, 16, 128)	0
conv2d_18 (Conv2D)	(None, 16, 16, 128)	147584
activation_20 (Activation)	(None, 16, 16, 128)	0
batch_normalization_20 (Batc	(None, 16, 16, 128)	512
max_pooling2d_6 (MaxPooling2	(None, 8, 8, 128)	0
conv2d_19 (Conv2D)	(None, 8, 8, 256)	295168
activation_21 (Activation)	(None, 8, 8, 256)	0
batch_normalization_21 (Batc	(None, 8, 8, 256)	1024
dropout_14 (Dropout)	(None, 8, 8, 256)	0
conv2d_20 (Conv2D)	(None, 8, 8, 256)	590080
activation_22 (Activation)	(None, 8, 8, 256)	0
batch_normalization_22 (Batc	(None, 8, 8, 256)	1024
dropout_15 (Dropout)	(None, 8, 8, 256)	0
conv2d_21 (Conv2D)	(None, 8, 8, 256)	590080
activation_23 (Activation)	(None, 8, 8, 256)	0
batch_normalization_23 (Batc	(None, 8, 8, 256)	1024
max_pooling2d_7 (MaxPooling2	(None, 4, 4, 256)	0
conv2d_22 (Conv2D)	(None, 4, 4, 512)	1180160

activation_24 (Activation)	(None, 4, 4, 512)	0
batch_normalization_24 (Batc	(None, 4, 4, 512)	2048
dropout_16 (Dropout)	(None, 4, 4, 512)	0
conv2d_23 (Conv2D)	(None, 4, 4, 512)	2359808
activation_25 (Activation)	(None, 4, 4, 512)	0
batch_normalization_25 (Batc	(None, 4, 4, 512)	2048
dropout_17 (Dropout)	(None, 4, 4, 512)	0
conv2d_24 (Conv2D)	(None, 4, 4, 512)	2359808
activation_26 (Activation)	(None, 4, 4, 512)	0
batch_normalization_26 (Batc	(None, 4, 4, 512)	2048
max_pooling2d_8 (MaxPooling2	(None, 2, 2, 512)	0
conv2d_25 (Conv2D)	(None, 2, 2, 512)	2359808
activation_27 (Activation)	(None, 2, 2, 512)	0
batch_normalization_27 (Batc	(None, 2, 2, 512)	2048
dropout_18 (Dropout)	(None, 2, 2, 512)	0
conv2d_26 (Conv2D)	(None, 2, 2, 512)	2359808
activation_28 (Activation)	(None, 2, 2, 512)	0
batch_normalization_28 (Batc	(None, 2, 2, 512)	2048
dropout_19 (Dropout)	(None, 2, 2, 512)	0
conv2d_27 (Conv2D)	(None, 2, 2, 512)	2359808
activation_29 (Activation)	(None, 2, 2, 512)	0
batch_normalization_29 (Batc	(None, 2, 2, 512)	2048
dropout_20 (Dropout)	(None, 2, 2, 512)	0
conv2d_28 (Conv2D)	(None, 2, 2, 512)	2359808
activation_30 (Activation)	(None, 2, 2, 512)	0
batch_normalization_30 (Batc	(None, 2, 2, 512)	2048
dropout_21 (Dropout)	(None, 2, 2, 512)	0
conv2d_29 (Conv2D)	(None, 2, 2, 512)	2359808
activation_31 (Activation)	(None, 2, 2, 512)	0
batch_normalization_31 (Batc	(None, 2, 2, 512)	2048

```
max_pooling2d_9 (MaxPooling2 (None, 1, 1, 512)          0
_________________________________________________________________
dropout_22 (Dropout)         (None, 1, 1, 512)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 512)                0
_________________________________________________________________
dense_3 (Dense)              (None, 512)                262656
_________________________________________________________________
activation_32 (Activation)   (None, 512)                0
_________________________________________________________________
batch_normalization_32 (Batc (None, 512)                2048
_________________________________________________________________
dense_4 (Dense)              (None, 512)                262656
_________________________________________________________________
activation_33 (Activation)   (None, 512)                0
_________________________________________________________________
batch_normalization_33 (Batc (None, 512)                2048
_________________________________________________________________
dropout_23 (Dropout)         (None, 512)                0
_________________________________________________________________
dense_5 (Dense)              (None, 100)                51300
=================================================================
Total params: 20,036,004
Trainable params: 20,023,460
Non-trainable params: 12,544
_________________________________________________________________
```

Training

5 learning rate training steps

In [84]:

```
model_sgd_2 = model
sgd = SGD(lr=0.03, decay = 0.0001, momentum=0.9, nesterov=True)
model_sgd_2.compile(loss='categorical_crossentropy', optimizer=sgd,metrics=['accuracy'])
history2 = model_sgd_2.fit(img_gen.flow(x_train, y_train, batch_size = 16), epochs=50,
                validation_data = (x_val, y_val), verbose=1)

model_sgd_2.save('group114_sgd_2.h5')
```

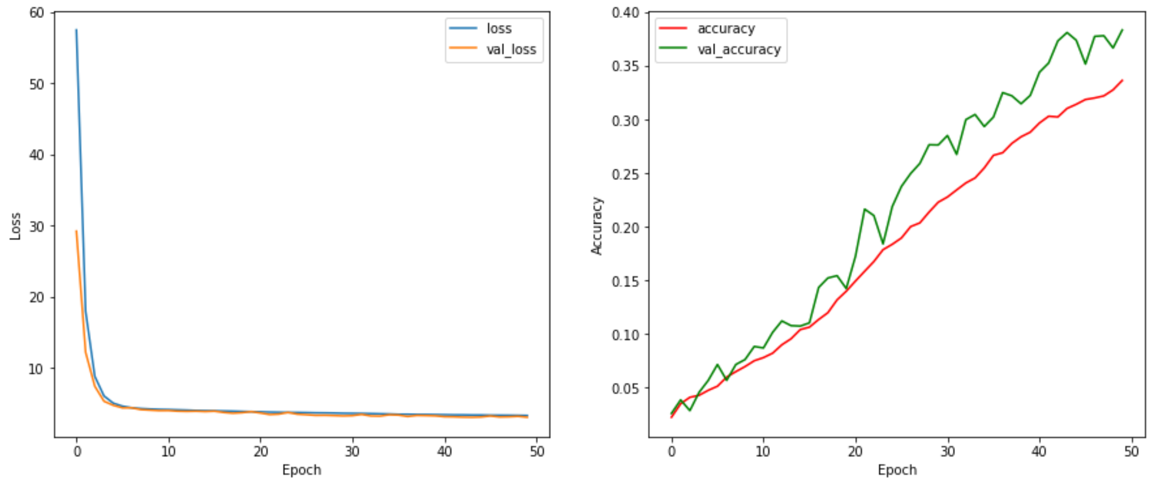
```
Epoch 1/50
2813/2813 [==============================] - 101s 36ms/step - loss: 57.4624 - accu
racy: 0.0224 - val_loss: 29.2074 - val_accuracy: 0.0260
Epoch 2/50
2813/2813 [==============================] - 101s 36ms/step - loss: 17.9655 - accu
racy: 0.0350 - val_loss: 12.2034 - val_accuracy: 0.0386
Epoch 3/50
2813/2813 [==============================] - 101s 36ms/step - loss: 8.8116 - accur
acy: 0.0410 - val_loss: 7.4301 - val_accuracy: 0.0286
Epoch 4/50
2813/2813 [==============================] - 102s 36ms/step - loss: 6.0648 - accur
acy: 0.0429 - val_loss: 5.3266 - val_accuracy: 0.0456
Epoch 5/50
2813/2813 [==============================] - 102s 36ms/step - loss: 5.0595 - accur
acy: 0.0475 - val_loss: 4.7595 - val_accuracy: 0.0568
Epoch 6/50
2813/2813 [==============================] - 103s 37ms/step - loss: 4.6303 - accur
acy: 0.0514 - val_loss: 4.3950 - val_accuracy: 0.0716
Epoch 7/50
2813/2813 [==============================] - 104s 37ms/step - loss: 4.4113 - accur
acy: 0.0599 - val_loss: 4.4021 - val_accuracy: 0.0568
Epoch 8/50
2813/2813 [==============================] - 104s 37ms/step - loss: 4.3018 - accur
acy: 0.0650 - val_loss: 4.1518 - val_accuracy: 0.0716
Epoch 9/50
2813/2813 [==============================] - 105s 37ms/step - loss: 4.2308 - accur
acy: 0.0697 - val_loss: 4.0791 - val_accuracy: 0.0762
Epoch 10/50
2813/2813 [==============================] - 105s 37ms/step - loss: 4.1794 - accur
acy: 0.0751 - val_loss: 4.0106 - val_accuracy: 0.0884
Epoch 11/50
2813/2813 [==============================] - 105s 37ms/step - loss: 4.1554 - accur
acy: 0.0780 - val_loss: 4.0202 - val_accuracy: 0.0870
Epoch 12/50
2813/2813 [==============================] - 105s 37ms/step - loss: 4.1142 - accur
acy: 0.0822 - val_loss: 3.9298 - val_accuracy: 0.1016
Epoch 13/50
2813/2813 [==============================] - 105s 37ms/step - loss: 4.0766 - accur
acy: 0.0900 - val_loss: 3.9046 - val_accuracy: 0.1122
Epoch 14/50
2813/2813 [==============================] - 105s 37ms/step - loss: 4.0292 - accur
acy: 0.0956 - val_loss: 3.9281 - val_accuracy: 0.1078
Epoch 15/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.9996 - accur
acy: 0.1041 - val_loss: 3.8808 - val_accuracy: 0.1074
Epoch 16/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.9783 - accur
acy: 0.1064 - val_loss: 3.9302 - val_accuracy: 0.1104
Epoch 17/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.9550 - accur
acy: 0.1135 - val_loss: 3.7549 - val_accuracy: 0.1434
Epoch 18/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.9292 - accur
acy: 0.1198 - val_loss: 3.6478 - val_accuracy: 0.1522
Epoch 19/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.8944 - accur
acy: 0.1318 - val_loss: 3.7210 - val_accuracy: 0.1544
Epoch 20/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.8640 - accur
acy: 0.1399 - val_loss: 3.8664 - val_accuracy: 0.1424
Epoch 21/50
```

```
2813/2813 [==============================] - 105s 37ms/step - loss: 3.8402 - accur
acy: 0.1494 - val_loss: 3.6757 - val_accuracy: 0.1724
Epoch 22/50
2813/2813 [==============================] - 116s 41ms/step - loss: 3.8023 - accur
acy: 0.1585 - val_loss: 3.4604 - val_accuracy: 0.2164
Epoch 23/50
2813/2813 [==============================] - 110s 39ms/step - loss: 3.7805 - accur
acy: 0.1676 - val_loss: 3.5198 - val_accuracy: 0.2104
Epoch 24/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.7525 - accur
acy: 0.1786 - val_loss: 3.7519 - val_accuracy: 0.1840
Epoch 25/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.7469 - accur
acy: 0.1836 - val_loss: 3.5127 - val_accuracy: 0.2188
Epoch 26/50
2813/2813 [==============================] - 104s 37ms/step - loss: 3.7259 - accur
acy: 0.1895 - val_loss: 3.4346 - val_accuracy: 0.2376
Epoch 27/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.6980 - accur
acy: 0.2001 - val_loss: 3.3451 - val_accuracy: 0.2496
Epoch 28/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.6885 - accur
acy: 0.2036 - val_loss: 3.3545 - val_accuracy: 0.2590
Epoch 29/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.6693 - accur
acy: 0.2136 - val_loss: 3.3166 - val_accuracy: 0.2764
Epoch 30/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.6374 - accur
acy: 0.2229 - val_loss: 3.2698 - val_accuracy: 0.2762
Epoch 31/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.6259 - accur
acy: 0.2277 - val_loss: 3.3057 - val_accuracy: 0.2850
Epoch 32/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.6110 - accur
acy: 0.2343 - val_loss: 3.4748 - val_accuracy: 0.2674
Epoch 33/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.5879 - accur
acy: 0.2408 - val_loss: 3.2624 - val_accuracy: 0.2998
Epoch 34/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.5602 - accur
acy: 0.2454 - val_loss: 3.2379 - val_accuracy: 0.3046
Epoch 35/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.5376 - accur
acy: 0.2548 - val_loss: 3.4272 - val_accuracy: 0.2934
Epoch 36/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.4927 - accur
acy: 0.2666 - val_loss: 3.3851 - val_accuracy: 0.3022
Epoch 37/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.4924 - accur
acy: 0.2690 - val_loss: 3.1857 - val_accuracy: 0.3250
Epoch 38/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.4675 - accur
acy: 0.2777 - val_loss: 3.3354 - val_accuracy: 0.3220
Epoch 39/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.4615 - accur
acy: 0.2838 - val_loss: 3.3130 - val_accuracy: 0.3146
Epoch 40/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.4477 - accur
acy: 0.2881 - val_loss: 3.2765 - val_accuracy: 0.3224
Epoch 41/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.4326 - accur
```

```
acy: 0.2967 - val_loss: 3.1582 - val_accuracy: 0.3440
Epoch 42/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.4110 - accur
acy: 0.3030 - val_loss: 3.1409 - val_accuracy: 0.3526
Epoch 43/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.4086 - accur
acy: 0.3023 - val_loss: 3.0980 - val_accuracy: 0.3732
Epoch 44/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.3881 - accur
acy: 0.3102 - val_loss: 3.0789 - val_accuracy: 0.3810
Epoch 45/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.3878 - accur
acy: 0.3141 - val_loss: 3.1193 - val_accuracy: 0.3738
Epoch 46/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.3677 - accur
acy: 0.3186 - val_loss: 3.2585 - val_accuracy: 0.3516
Epoch 47/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.3591 - accur
acy: 0.3200 - val_loss: 3.1216 - val_accuracy: 0.3774
Epoch 48/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.3542 - accur
acy: 0.3220 - val_loss: 3.1465 - val_accuracy: 0.3780
Epoch 49/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.3429 - accur
acy: 0.3276 - val_loss: 3.2085 - val_accuracy: 0.3666
Epoch 50/50
2813/2813 [==============================] - 105s 37ms/step - loss: 3.3206 - accur
acy: 0.3364 - val_loss: 3.0826 - val_accuracy: 0.3834
```

In [85]:

```
plot_history(history2.history)
```



In [86]:

```
model_sgd_3 = model_sgd_2
sgd = SGD(lr=0.005, decay = 0.00001, momentum=0.9, nesterov=True)
model_sgd_3.compile(loss='categorical_crossentropy', optimizer=sgd,metrics=['accuracy'])
# 减少epochs 减少batchsize
history3 = model_sgd_3.fit(img_gen.flow(x_train, y_train, batch_size = 32), epochs=50,
                  validation_data = (x_val, y_val), verbose=1)
model_sgd_3.save('group114_sgd_3.h5')
```

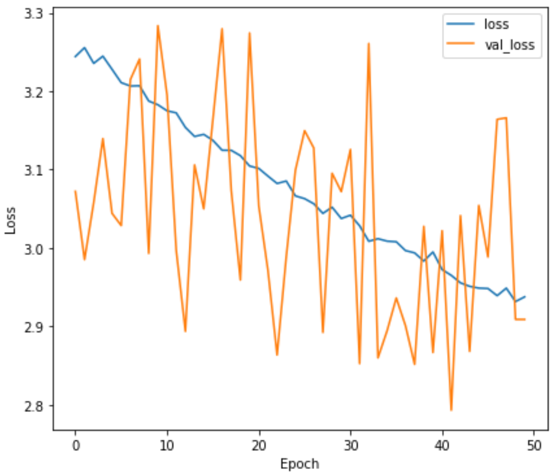
```
Epoch 1/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.2442 - accura
cy: 0.3513 - val_loss: 3.0722 - val_accuracy: 0.3984
Epoch 2/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.2555 - accura
cy: 0.3545 - val_loss: 2.9853 - val_accuracy: 0.4024
Epoch 3/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.2355 - accura
cy: 0.3586 - val_loss: 3.0580 - val_accuracy: 0.4094
Epoch 4/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.2447 - accura
cy: 0.3640 - val_loss: 3.1396 - val_accuracy: 0.4026
Epoch 5/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.2279 - accura
cy: 0.3688 - val_loss: 3.0443 - val_accuracy: 0.4192
Epoch 6/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.2109 - accura
cy: 0.3736 - val_loss: 3.0287 - val_accuracy: 0.4202
Epoch 7/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.2066 - accura
cy: 0.3779 - val_loss: 3.2149 - val_accuracy: 0.4040
Epoch 8/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.2070 - accura
cy: 0.3830 - val_loss: 3.2411 - val_accuracy: 0.3844
Epoch 9/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.1875 - accura
cy: 0.3881 - val_loss: 2.9931 - val_accuracy: 0.4310
Epoch 10/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.1827 - accura
cy: 0.3900 - val_loss: 3.2836 - val_accuracy: 0.3910
Epoch 11/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.1750 - accura
cy: 0.3944 - val_loss: 3.1951 - val_accuracy: 0.4080
Epoch 12/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.1724 - accura
cy: 0.3969 - val_loss: 2.9970 - val_accuracy: 0.4338
Epoch 13/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.1539 - accura
cy: 0.3989 - val_loss: 2.8937 - val_accuracy: 0.4598
Epoch 14/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.1423 - accura
cy: 0.4054 - val_loss: 3.1060 - val_accuracy: 0.4438
Epoch 15/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.1450 - accura
cy: 0.4073 - val_loss: 3.0500 - val_accuracy: 0.4392
Epoch 16/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.1374 - accura
cy: 0.4100 - val_loss: 3.1640 - val_accuracy: 0.4258
Epoch 17/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.1248 - accura
cy: 0.4159 - val_loss: 3.2796 - val_accuracy: 0.4152
Epoch 18/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.1246 - accura
cy: 0.4160 - val_loss: 3.0736 - val_accuracy: 0.4502
Epoch 19/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.1178 - accura
cy: 0.4213 - val_loss: 2.9592 - val_accuracy: 0.4562
Epoch 20/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.1046 - accura
cy: 0.4255 - val_loss: 3.2743 - val_accuracy: 0.4262
Epoch 21/50
```

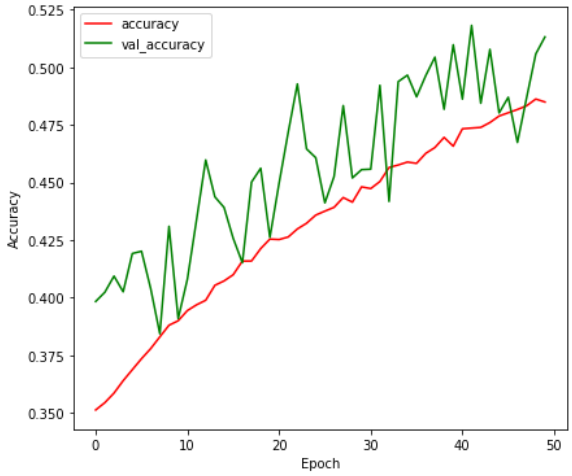
```
1407/1407 [==============================] - 65s 46ms/step - loss: 3.1016 - accura
cy: 0.4253 - val_loss: 3.0539 - val_accuracy: 0.4496
Epoch 22/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0918 - accura
cy: 0.4264 - val_loss: 2.9718 - val_accuracy: 0.4718
Epoch 23/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0823 - accura
cy: 0.4299 - val_loss: 2.8637 - val_accuracy: 0.4928
Epoch 24/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0855 - accura
cy: 0.4323 - val_loss: 2.9888 - val_accuracy: 0.4646
Epoch 25/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0665 - accura
cy: 0.4359 - val_loss: 3.0994 - val_accuracy: 0.4608
Epoch 26/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0630 - accura
cy: 0.4376 - val_loss: 3.1497 - val_accuracy: 0.4412
Epoch 27/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0565 - accura
cy: 0.4392 - val_loss: 3.1276 - val_accuracy: 0.4526
Epoch 28/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0442 - accura
cy: 0.4435 - val_loss: 2.8924 - val_accuracy: 0.4834
Epoch 29/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0521 - accura
cy: 0.4415 - val_loss: 3.0954 - val_accuracy: 0.4520
Epoch 30/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0376 - accura
cy: 0.4481 - val_loss: 3.0720 - val_accuracy: 0.4556
Epoch 31/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0419 - accura
cy: 0.4474 - val_loss: 3.1257 - val_accuracy: 0.4558
Epoch 32/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0284 - accura
cy: 0.4504 - val_loss: 2.8526 - val_accuracy: 0.4922
Epoch 33/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0086 - accura
cy: 0.4565 - val_loss: 3.2609 - val_accuracy: 0.4418
Epoch 34/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0119 - accura
cy: 0.4576 - val_loss: 2.8599 - val_accuracy: 0.4938
Epoch 35/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0088 - accura
cy: 0.4589 - val_loss: 2.8946 - val_accuracy: 0.4966
Epoch 36/50
1407/1407 [==============================] - 65s 46ms/step - loss: 3.0080 - accura
cy: 0.4583 - val_loss: 2.9363 - val_accuracy: 0.4872
Epoch 37/50
1407/1407 [==============================] - 65s 46ms/step - loss: 2.9968 - accura
cy: 0.4627 - val_loss: 2.9011 - val_accuracy: 0.4964
Epoch 38/50
1407/1407 [==============================] - 65s 46ms/step - loss: 2.9938 - accura
cy: 0.4652 - val_loss: 2.8517 - val_accuracy: 0.5044
Epoch 39/50
1407/1407 [==============================] - 65s 46ms/step - loss: 2.9834 - accura
cy: 0.4696 - val_loss: 3.0277 - val_accuracy: 0.4818
Epoch 40/50
1407/1407 [==============================] - 65s 46ms/step - loss: 2.9950 - accura
cy: 0.4658 - val_loss: 2.8668 - val_accuracy: 0.5098
Epoch 41/50
1407/1407 [==============================] - 65s 46ms/step - loss: 2.9726 - accura
```

```
cy: 0.4733 - val_loss: 3.0221 - val_accuracy: 0.4862
Epoch 42/50
1407/1407 [==============================] - 65s 46ms/step - loss: 2.9652 - accura
cy: 0.4736 - val_loss: 2.7932 - val_accuracy: 0.5182
Epoch 43/50
1407/1407 [==============================] - 65s 46ms/step - loss: 2.9556 - accura
cy: 0.4739 - val_loss: 3.0414 - val_accuracy: 0.4844
Epoch 44/50
1407/1407 [==============================] - 65s 46ms/step - loss: 2.9512 - accura
cy: 0.4760 - val_loss: 2.8682 - val_accuracy: 0.5078
Epoch 45/50
1407/1407 [==============================] - 65s 46ms/step - loss: 2.9489 - accura
cy: 0.4789 - val_loss: 3.0544 - val_accuracy: 0.4802
Epoch 46/50
1407/1407 [==============================] - 65s 46ms/step - loss: 2.9484 - accura
cy: 0.4803 - val_loss: 2.9886 - val_accuracy: 0.4870
Epoch 47/50
1407/1407 [==============================] - 65s 46ms/step - loss: 2.9392 - accura
cy: 0.4817 - val_loss: 3.1641 - val_accuracy: 0.4674
Epoch 48/50
1407/1407 [==============================] - 65s 46ms/step - loss: 2.9489 - accura
cy: 0.4833 - val_loss: 3.1660 - val_accuracy: 0.4868
Epoch 49/50
1407/1407 [==============================] - 65s 46ms/step - loss: 2.9317 - accura
cy: 0.4862 - val_loss: 2.9090 - val_accuracy: 0.5058
Epoch 50/50
1407/1407 [==============================] - 65s 46ms/step - loss: 2.9377 - accura
cy: 0.4850 - val_loss: 2.9089 - val_accuracy: 0.5132
```

In [87]:

```
plot_history(history3.history)
```





In [88]:

```
model_sgd_4 = model_sgd_3
sgd = SGD(lr=0.0005, decay = 0.0000001, momentum=0.9, nesterov=True)
model_sgd_4.compile(loss='categorical_crossentropy', optimizer=sgd,metrics=['accuracy'])
history4 = model_sgd_4.fit(img_gen.flow(x_train, y_train, batch_size = 64), epochs=50,
                 validation_data = (x_val, y_val), verbose=1)
model_sgd_4.save('group114_sgd_4.h5')
```

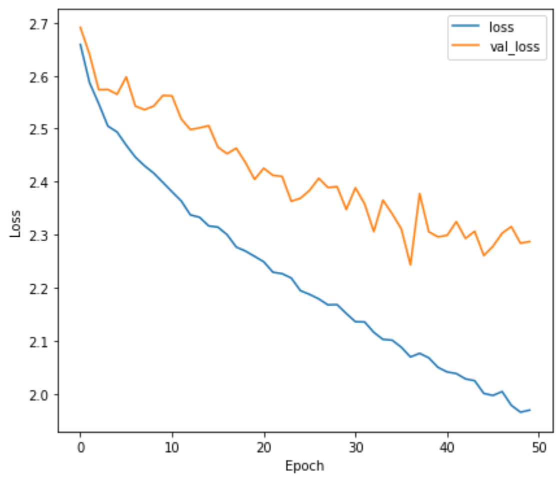
```
Epoch 1/50
704/704 [==============================] - 41s 58ms/step - loss: 2.6588 - accurac
y: 0.5478 - val_loss: 2.6909 - val_accuracy: 0.5514
Epoch 2/50
704/704 [==============================] - 41s 59ms/step - loss: 2.5868 - accurac
y: 0.5618 - val_loss: 2.6400 - val_accuracy: 0.5630
Epoch 3/50
704/704 [==============================] - 41s 58ms/step - loss: 2.5473 - accurac
y: 0.5672 - val_loss: 2.5734 - val_accuracy: 0.5740
Epoch 4/50
704/704 [==============================] - 41s 58ms/step - loss: 2.5049 - accurac
y: 0.5766 - val_loss: 2.5738 - val_accuracy: 0.5720
Epoch 5/50
704/704 [==============================] - 41s 58ms/step - loss: 2.4937 - accurac
y: 0.5791 - val_loss: 2.5649 - val_accuracy: 0.5760
Epoch 6/50
704/704 [==============================] - 41s 58ms/step - loss: 2.4689 - accurac
y: 0.5821 - val_loss: 2.5976 - val_accuracy: 0.5704
Epoch 7/50
704/704 [==============================] - 41s 58ms/step - loss: 2.4462 - accurac
y: 0.5862 - val_loss: 2.5424 - val_accuracy: 0.5828
Epoch 8/50
704/704 [==============================] - 41s 59ms/step - loss: 2.4299 - accurac
y: 0.5884 - val_loss: 2.5358 - val_accuracy: 0.5788
Epoch 9/50
704/704 [==============================] - 41s 59ms/step - loss: 2.4161 - accurac
y: 0.5911 - val_loss: 2.5427 - val_accuracy: 0.5770
Epoch 10/50
704/704 [==============================] - 41s 59ms/step - loss: 2.3986 - accurac
y: 0.5937 - val_loss: 2.5626 - val_accuracy: 0.5748
Epoch 11/50
704/704 [==============================] - 41s 59ms/step - loss: 2.3809 - accurac
y: 0.5974 - val_loss: 2.5619 - val_accuracy: 0.5722
Epoch 12/50
704/704 [==============================] - 41s 58ms/step - loss: 2.3637 - accurac
y: 0.5998 - val_loss: 2.5183 - val_accuracy: 0.5800
Epoch 13/50
704/704 [==============================] - 41s 58ms/step - loss: 2.3373 - accurac
y: 0.6046 - val_loss: 2.4982 - val_accuracy: 0.5810
Epoch 14/50
704/704 [==============================] - 41s 58ms/step - loss: 2.3328 - accurac
y: 0.6026 - val_loss: 2.5015 - val_accuracy: 0.5798
Epoch 15/50
704/704 [==============================] - 41s 58ms/step - loss: 2.3166 - accurac
y: 0.6066 - val_loss: 2.5057 - val_accuracy: 0.5808
Epoch 16/50
704/704 [==============================] - 41s 59ms/step - loss: 2.3143 - accurac
y: 0.6061 - val_loss: 2.4652 - val_accuracy: 0.5864
Epoch 17/50
704/704 [==============================] - 41s 59ms/step - loss: 2.3001 - accurac
y: 0.6074 - val_loss: 2.4528 - val_accuracy: 0.5866
Epoch 18/50
704/704 [==============================] - 41s 58ms/step - loss: 2.2768 - accurac
y: 0.6112 - val_loss: 2.4632 - val_accuracy: 0.5858
Epoch 19/50
704/704 [==============================] - 41s 59ms/step - loss: 2.2691 - accurac
y: 0.6128 - val_loss: 2.4361 - val_accuracy: 0.5898
Epoch 20/50
704/704 [==============================] - 41s 59ms/step - loss: 2.2591 - accurac
y: 0.6168 - val_loss: 2.4044 - val_accuracy: 0.5924
Epoch 21/50
```

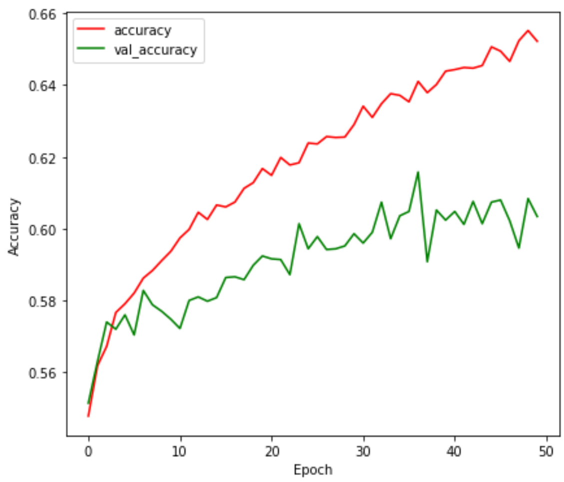
```
704/704 [==============================] - 41s 58ms/step - loss: 2.2489 - accurac
y: 0.6149 - val_loss: 2.4255 - val_accuracy: 0.5916
Epoch 22/50
704/704 [==============================] - 41s 59ms/step - loss: 2.2292 - accurac
y: 0.6198 - val_loss: 2.4119 - val_accuracy: 0.5914
Epoch 23/50
704/704 [==============================] - 41s 59ms/step - loss: 2.2265 - accurac
y: 0.6178 - val_loss: 2.4099 - val_accuracy: 0.5872
Epoch 24/50
704/704 [==============================] - 41s 58ms/step - loss: 2.2183 - accurac
y: 0.6184 - val_loss: 2.3631 - val_accuracy: 0.6014
Epoch 25/50
704/704 [==============================] - 41s 59ms/step - loss: 2.1945 - accurac
y: 0.6239 - val_loss: 2.3688 - val_accuracy: 0.5944
Epoch 26/50
704/704 [==============================] - 41s 59ms/step - loss: 2.1874 - accurac
y: 0.6236 - val_loss: 2.3838 - val_accuracy: 0.5978
Epoch 27/50
704/704 [==============================] - 41s 58ms/step - loss: 2.1789 - accurac
y: 0.6257 - val_loss: 2.4065 - val_accuracy: 0.5942
Epoch 28/50
704/704 [==============================] - 41s 58ms/step - loss: 2.1676 - accurac
y: 0.6254 - val_loss: 2.3890 - val_accuracy: 0.5944
Epoch 29/50
704/704 [==============================] - 41s 58ms/step - loss: 2.1681 - accurac
y: 0.6255 - val_loss: 2.3906 - val_accuracy: 0.5952
Epoch 30/50
704/704 [==============================] - 41s 58ms/step - loss: 2.1513 - accurac
y: 0.6290 - val_loss: 2.3477 - val_accuracy: 0.5986
Epoch 31/50
704/704 [==============================] - 41s 58ms/step - loss: 2.1358 - accurac
y: 0.6341 - val_loss: 2.3885 - val_accuracy: 0.5960
Epoch 32/50
704/704 [==============================] - 41s 58ms/step - loss: 2.1355 - accurac
y: 0.6310 - val_loss: 2.3581 - val_accuracy: 0.5990
Epoch 33/50
704/704 [==============================] - 41s 59ms/step - loss: 2.1159 - accurac
y: 0.6348 - val_loss: 2.3058 - val_accuracy: 0.6074
Epoch 34/50
704/704 [==============================] - 41s 58ms/step - loss: 2.1025 - accurac
y: 0.6376 - val_loss: 2.3653 - val_accuracy: 0.5972
Epoch 35/50
704/704 [==============================] - 41s 59ms/step - loss: 2.1011 - accurac
y: 0.6372 - val_loss: 2.3395 - val_accuracy: 0.6036
Epoch 36/50
704/704 [==============================] - 41s 59ms/step - loss: 2.0878 - accurac
y: 0.6354 - val_loss: 2.3110 - val_accuracy: 0.6048
Epoch 37/50
704/704 [==============================] - 41s 59ms/step - loss: 2.0694 - accurac
y: 0.6410 - val_loss: 2.2432 - val_accuracy: 0.6158
Epoch 38/50
704/704 [==============================] - 41s 59ms/step - loss: 2.0762 - accurac
y: 0.6379 - val_loss: 2.3776 - val_accuracy: 0.5908
Epoch 39/50
704/704 [==============================] - 41s 58ms/step - loss: 2.0677 - accurac
y: 0.6402 - val_loss: 2.3055 - val_accuracy: 0.6052
Epoch 40/50
704/704 [==============================] - 41s 58ms/step - loss: 2.0498 - accurac
y: 0.6439 - val_loss: 2.2959 - val_accuracy: 0.6024
Epoch 41/50
704/704 [==============================] - 41s 59ms/step - loss: 2.0411 - accurac
```

```
y: 0.6443 - val_loss: 2.2988 - val_accuracy: 0.6048
Epoch 42/50
704/704 [==============================] - 41s 58ms/step - loss: 2.0381 - accurac
y: 0.6449 - val_loss: 2.3247 - val_accuracy: 0.6012
Epoch 43/50
704/704 [==============================] - 41s 58ms/step - loss: 2.0280 - accurac
y: 0.6447 - val_loss: 2.2930 - val_accuracy: 0.6076
Epoch 44/50
704/704 [==============================] - 41s 59ms/step - loss: 2.0246 - accurac
y: 0.6455 - val_loss: 2.3065 - val_accuracy: 0.6014
Epoch 45/50
704/704 [==============================] - 41s 59ms/step - loss: 2.0006 - accurac
y: 0.6507 - val_loss: 2.2607 - val_accuracy: 0.6074
Epoch 46/50
704/704 [==============================] - 41s 59ms/step - loss: 1.9967 - accurac
y: 0.6495 - val_loss: 2.2783 - val_accuracy: 0.6080
Epoch 47/50
704/704 [==============================] - 41s 58ms/step - loss: 2.0042 - accurac
y: 0.6466 - val_loss: 2.3029 - val_accuracy: 0.6022
Epoch 48/50
704/704 [==============================] - 41s 58ms/step - loss: 1.9782 - accurac
y: 0.6524 - val_loss: 2.3153 - val_accuracy: 0.5946
Epoch 49/50
704/704 [==============================] - 41s 58ms/step - loss: 1.9652 - accurac
y: 0.6552 - val_loss: 2.2839 - val_accuracy: 0.6084
Epoch 50/50
704/704 [==============================] - 41s 58ms/step - loss: 1.9691 - accurac
y: 0.6522 - val_loss: 2.2871 - val_accuracy: 0.6034
```

In [89]:

```
plot_history(history4.history)
```





In [90]:

```
model_sgd_5 = model_sgd_4
sgd = SGD(lr=0.00005, decay = 0.00000001, momentum=0.9, nesterov=True)
model_sgd_5.compile(loss='categorical_crossentropy', optimizer=sgd,metrics=['accuracy'])
history5 = model_sgd_5.fit(img_gen.flow(x_train, y_train, batch_size = 128), epochs=50,
                 validation_data = (x_val, y_val), verbose=1)
model_sgd_5.save('group114_sgd_5.h5')
```

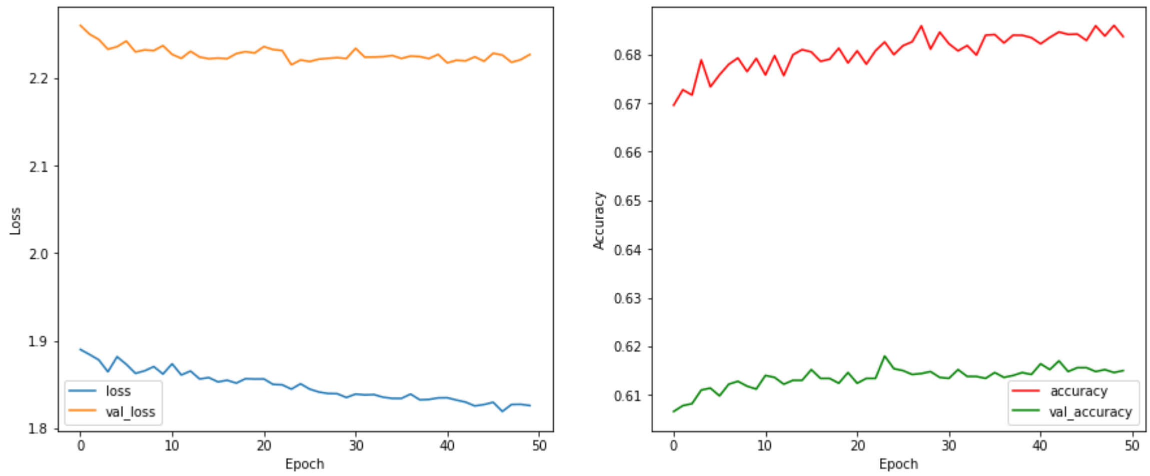
```
Epoch 1/50
352/352 [==============================] - 33s 93ms/step - loss: 1.8899 - accurac
y: 0.6696 - val_loss: 2.2599 - val_accuracy: 0.6066
Epoch 2/50
352/352 [==============================] - 31s 87ms/step - loss: 1.8842 - accurac
y: 0.6727 - val_loss: 2.2499 - val_accuracy: 0.6078
Epoch 3/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8781 - accurac
y: 0.6716 - val_loss: 2.2439 - val_accuracy: 0.6082
Epoch 4/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8644 - accurac
y: 0.6789 - val_loss: 2.2328 - val_accuracy: 0.6110
Epoch 5/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8816 - accurac
y: 0.6733 - val_loss: 2.2357 - val_accuracy: 0.6114
Epoch 6/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8730 - accurac
y: 0.6758 - val_loss: 2.2421 - val_accuracy: 0.6098
Epoch 7/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8626 - accurac
y: 0.6780 - val_loss: 2.2297 - val_accuracy: 0.6122
Epoch 8/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8655 - accurac
y: 0.6793 - val_loss: 2.2320 - val_accuracy: 0.6128
Epoch 9/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8704 - accurac
y: 0.6765 - val_loss: 2.2312 - val_accuracy: 0.6118
Epoch 10/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8619 - accurac
y: 0.6792 - val_loss: 2.2368 - val_accuracy: 0.6112
Epoch 11/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8735 - accurac
y: 0.6758 - val_loss: 2.2269 - val_accuracy: 0.6140
Epoch 12/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8608 - accurac
y: 0.6798 - val_loss: 2.2223 - val_accuracy: 0.6136
Epoch 13/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8653 - accurac
y: 0.6756 - val_loss: 2.2304 - val_accuracy: 0.6122
Epoch 14/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8562 - accurac
y: 0.6799 - val_loss: 2.2239 - val_accuracy: 0.6130
Epoch 15/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8578 - accurac
y: 0.6810 - val_loss: 2.2219 - val_accuracy: 0.6130
Epoch 16/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8529 - accurac
y: 0.6805 - val_loss: 2.2227 - val_accuracy: 0.6152
Epoch 17/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8548 - accurac
y: 0.6786 - val_loss: 2.2219 - val_accuracy: 0.6134
Epoch 18/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8515 - accurac
y: 0.6790 - val_loss: 2.2278 - val_accuracy: 0.6134
Epoch 19/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8566 - accurac
y: 0.6813 - val_loss: 2.2300 - val_accuracy: 0.6124
Epoch 20/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8562 - accurac
y: 0.6782 - val_loss: 2.2285 - val_accuracy: 0.6146
Epoch 21/50
```

```
352/352 [==============================] - 31s 88ms/step - loss: 1.8563 - accurac
y: 0.6807 - val_loss: 2.2358 - val_accuracy: 0.6124
Epoch 22/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8501 - accurac
y: 0.6780 - val_loss: 2.2324 - val_accuracy: 0.6134
Epoch 23/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8496 - accurac
y: 0.6808 - val_loss: 2.2313 - val_accuracy: 0.6134
Epoch 24/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8447 - accurac
y: 0.6826 - val_loss: 2.2150 - val_accuracy: 0.6180
Epoch 25/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8508 - accurac
y: 0.6800 - val_loss: 2.2205 - val_accuracy: 0.6154
Epoch 26/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8447 - accurac
y: 0.6818 - val_loss: 2.2187 - val_accuracy: 0.6150
Epoch 27/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8414 - accurac
y: 0.6826 - val_loss: 2.2214 - val_accuracy: 0.6142
Epoch 28/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8397 - accurac
y: 0.6859 - val_loss: 2.2223 - val_accuracy: 0.6144
Epoch 29/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8394 - accurac
y: 0.6811 - val_loss: 2.2232 - val_accuracy: 0.6148
Epoch 30/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8351 - accurac
y: 0.6846 - val_loss: 2.2221 - val_accuracy: 0.6136
Epoch 31/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8390 - accurac
y: 0.6822 - val_loss: 2.2338 - val_accuracy: 0.6134
Epoch 32/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8380 - accurac
y: 0.6807 - val_loss: 2.2238 - val_accuracy: 0.6152
Epoch 33/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8384 - accurac
y: 0.6818 - val_loss: 2.2239 - val_accuracy: 0.6138
Epoch 34/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8354 - accurac
y: 0.6799 - val_loss: 2.2242 - val_accuracy: 0.6138
Epoch 35/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8341 - accurac
y: 0.6840 - val_loss: 2.2256 - val_accuracy: 0.6134
Epoch 36/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8341 - accurac
y: 0.6841 - val_loss: 2.2222 - val_accuracy: 0.6146
Epoch 37/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8390 - accurac
y: 0.6824 - val_loss: 2.2249 - val_accuracy: 0.6136
Epoch 38/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8325 - accurac
y: 0.6840 - val_loss: 2.2244 - val_accuracy: 0.6140
Epoch 39/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8329 - accurac
y: 0.6839 - val_loss: 2.2221 - val_accuracy: 0.6146
Epoch 40/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8347 - accurac
y: 0.6834 - val_loss: 2.2268 - val_accuracy: 0.6142
Epoch 41/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8348 - accurac
```

```
y: 0.6822 - val_loss: 2.2172 - val_accuracy: 0.6164
Epoch 42/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8322 - accurac
y: 0.6835 - val_loss: 2.2202 - val_accuracy: 0.6152
Epoch 43/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8298 - accurac
y: 0.6846 - val_loss: 2.2195 - val_accuracy: 0.6170
Epoch 44/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8255 - accurac
y: 0.6841 - val_loss: 2.2241 - val_accuracy: 0.6148
Epoch 45/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8271 - accurac
y: 0.6842 - val_loss: 2.2190 - val_accuracy: 0.6156
Epoch 46/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8296 - accurac
y: 0.6829 - val_loss: 2.2281 - val_accuracy: 0.6156
Epoch 47/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8192 - accurac
y: 0.6859 - val_loss: 2.2261 - val_accuracy: 0.6148
Epoch 48/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8271 - accurac
y: 0.6838 - val_loss: 2.2178 - val_accuracy: 0.6152
Epoch 49/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8273 - accurac
y: 0.6860 - val_loss: 2.2208 - val_accuracy: 0.6146
Epoch 50/50
352/352 [==============================] - 31s 88ms/step - loss: 1.8259 - accurac
y: 0.6837 - val_loss: 2.2267 - val_accuracy: 0.6150
```

In [91]:

```
plot_history(history5.history)
```



In [92]:

```
model_sgd_6 = model_sgd_5
sgd = SGD(lr=0.000005, decay = 0.00000001, momentum=0.9, nesterov=True)
model_sgd_6.compile(loss='categorical_crossentropy', optimizer=sgd,metrics=['accuracy'])
history6 = model_sgd_6.fit(img_gen.flow(x_train, y_train, batch_size = 256), epochs=50,
                 validation_data = (x_val, y_val), verbose=1)
```

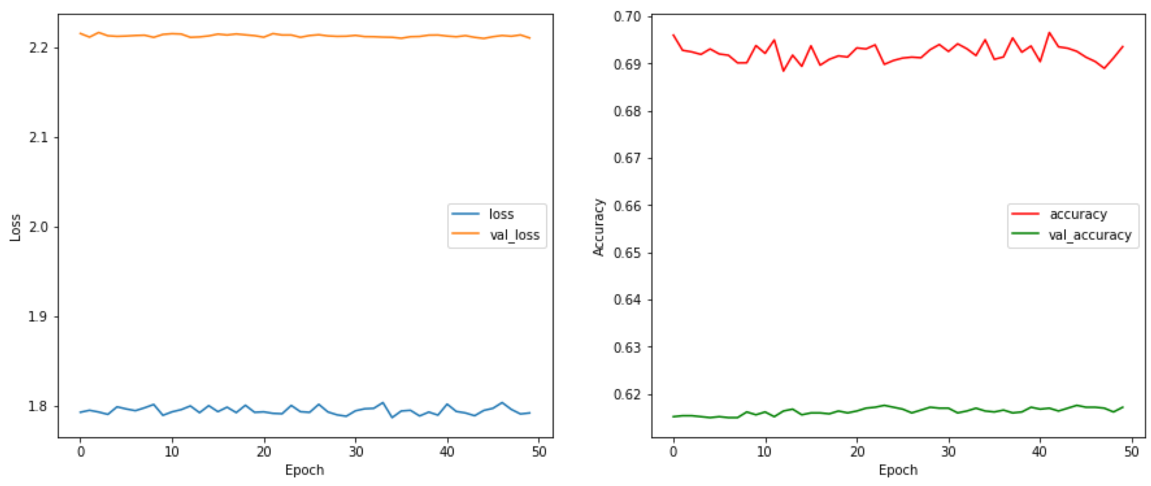
```
Epoch 1/50
176/176 [==============================] - 29s 166ms/step - loss: 1.7930 - accurac
y: 0.6960 - val_loss: 2.2159 - val_accuracy: 0.6152
Epoch 2/50
176/176 [==============================] - 26s 148ms/step - loss: 1.7952 - accurac
y: 0.6928 - val_loss: 2.2118 - val_accuracy: 0.6154
Epoch 3/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7932 - accurac
y: 0.6925 - val_loss: 2.2170 - val_accuracy: 0.6154
Epoch 4/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7906 - accurac
y: 0.6920 - val_loss: 2.2133 - val_accuracy: 0.6152
Epoch 5/50
176/176 [==============================] - 26s 148ms/step - loss: 1.7991 - accurac
y: 0.6931 - val_loss: 2.2126 - val_accuracy: 0.6150
Epoch 6/50
176/176 [==============================] - 26s 148ms/step - loss: 1.7967 - accurac
y: 0.6920 - val_loss: 2.2130 - val_accuracy: 0.6152
Epoch 7/50
176/176 [==============================] - 26s 150ms/step - loss: 1.7947 - accurac
y: 0.6918 - val_loss: 2.2136 - val_accuracy: 0.6150
Epoch 8/50
176/176 [==============================] - 26s 150ms/step - loss: 1.7980 - accurac
y: 0.6902 - val_loss: 2.2139 - val_accuracy: 0.6150
Epoch 9/50
176/176 [==============================] - 26s 148ms/step - loss: 1.8017 - accurac
y: 0.6902 - val_loss: 2.2116 - val_accuracy: 0.6162
Epoch 10/50
176/176 [==============================] - 26s 150ms/step - loss: 1.7896 - accurac
y: 0.6938 - val_loss: 2.2148 - val_accuracy: 0.6156
Epoch 11/50
176/176 [==============================] - 26s 148ms/step - loss: 1.7935 - accurac
y: 0.6922 - val_loss: 2.2156 - val_accuracy: 0.6162
Epoch 12/50
176/176 [==============================] - 26s 150ms/step - loss: 1.7959 - accurac
y: 0.6950 - val_loss: 2.2152 - val_accuracy: 0.6152
Epoch 13/50
176/176 [==============================] - 26s 149ms/step - loss: 1.8001 - accurac
y: 0.6884 - val_loss: 2.2116 - val_accuracy: 0.6164
Epoch 14/50
176/176 [==============================] - 26s 150ms/step - loss: 1.7925 - accurac
y: 0.6918 - val_loss: 2.2120 - val_accuracy: 0.6168
Epoch 15/50
176/176 [==============================] - 26s 148ms/step - loss: 1.8004 - accurac
y: 0.6894 - val_loss: 2.2131 - val_accuracy: 0.6156
Epoch 16/50
176/176 [==============================] - 26s 150ms/step - loss: 1.7937 - accurac
y: 0.6938 - val_loss: 2.2152 - val_accuracy: 0.6160
Epoch 17/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7987 - accurac
y: 0.6897 - val_loss: 2.2142 - val_accuracy: 0.6160
Epoch 18/50
176/176 [==============================] - 26s 150ms/step - loss: 1.7926 - accurac
y: 0.6909 - val_loss: 2.2153 - val_accuracy: 0.6158
Epoch 19/50
176/176 [==============================] - 26s 149ms/step - loss: 1.8008 - accurac
y: 0.6916 - val_loss: 2.2143 - val_accuracy: 0.6164
Epoch 20/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7929 - accurac
y: 0.6914 - val_loss: 2.2134 - val_accuracy: 0.6160
Epoch 21/50
```

```
176/176 [==============================] - 26s 148ms/step - loss: 1.7934 - accurac
y: 0.6933 - val_loss: 2.2117 - val_accuracy: 0.6164
Epoch 22/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7918 - accurac
y: 0.6931 - val_loss: 2.2157 - val_accuracy: 0.6170
Epoch 23/50
176/176 [==============================] - 26s 150ms/step - loss: 1.7913 - accurac
y: 0.6940 - val_loss: 2.2142 - val_accuracy: 0.6172
Epoch 24/50
176/176 [==============================] - 26s 148ms/step - loss: 1.8006 - accurac
y: 0.6898 - val_loss: 2.2142 - val_accuracy: 0.6176
Epoch 25/50
176/176 [==============================] - 26s 150ms/step - loss: 1.7936 - accurac
y: 0.6907 - val_loss: 2.2116 - val_accuracy: 0.6172
Epoch 26/50
176/176 [==============================] - 26s 150ms/step - loss: 1.7929 - accurac
y: 0.6912 - val_loss: 2.2136 - val_accuracy: 0.6168
Epoch 27/50
176/176 [==============================] - 26s 149ms/step - loss: 1.8018 - accurac
y: 0.6914 - val_loss: 2.2145 - val_accuracy: 0.6160
Epoch 28/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7933 - accurac
y: 0.6912 - val_loss: 2.2131 - val_accuracy: 0.6166
Epoch 29/50
176/176 [==============================] - 26s 148ms/step - loss: 1.7901 - accurac
y: 0.6930 - val_loss: 2.2126 - val_accuracy: 0.6172
Epoch 30/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7885 - accurac
y: 0.6940 - val_loss: 2.2128 - val_accuracy: 0.6170
Epoch 31/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7946 - accurac
y: 0.6926 - val_loss: 2.2137 - val_accuracy: 0.6170
Epoch 32/50
176/176 [==============================] - 26s 151ms/step - loss: 1.7969 - accurac
y: 0.6942 - val_loss: 2.2124 - val_accuracy: 0.6160
Epoch 33/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7974 - accurac
y: 0.6932 - val_loss: 2.2122 - val_accuracy: 0.6164
Epoch 34/50
176/176 [==============================] - 26s 148ms/step - loss: 1.8038 - accurac
y: 0.6917 - val_loss: 2.2118 - val_accuracy: 0.6170
Epoch 35/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7870 - accurac
y: 0.6951 - val_loss: 2.2116 - val_accuracy: 0.6164
Epoch 36/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7944 - accurac
y: 0.6909 - val_loss: 2.2105 - val_accuracy: 0.6162
Epoch 37/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7952 - accurac
y: 0.6914 - val_loss: 2.2123 - val_accuracy: 0.6166
Epoch 38/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7888 - accurac
y: 0.6954 - val_loss: 2.2125 - val_accuracy: 0.6160
Epoch 39/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7932 - accurac
y: 0.6924 - val_loss: 2.2140 - val_accuracy: 0.6162
Epoch 40/50
176/176 [==============================] - 26s 148ms/step - loss: 1.7898 - accurac
y: 0.6937 - val_loss: 2.2141 - val_accuracy: 0.6172
Epoch 41/50
176/176 [==============================] - 26s 150ms/step - loss: 1.8021 - accurac
```

```
y: 0.6904 - val_loss: 2.2130 - val_accuracy: 0.6168
Epoch 42/50
176/176 [==============================] - 26s 148ms/step - loss: 1.7939 - accurac
y: 0.6966 - val_loss: 2.2122 - val_accuracy: 0.6170
Epoch 43/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7922 - accurac
y: 0.6936 - val_loss: 2.2134 - val_accuracy: 0.6164
Epoch 44/50
176/176 [==============================] - 26s 150ms/step - loss: 1.7891 - accurac
y: 0.6933 - val_loss: 2.2116 - val_accuracy: 0.6170
Epoch 45/50
176/176 [==============================] - 26s 150ms/step - loss: 1.7950 - accurac
y: 0.6926 - val_loss: 2.2103 - val_accuracy: 0.6176
Epoch 46/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7974 - accurac
y: 0.6914 - val_loss: 2.2123 - val_accuracy: 0.6172
Epoch 47/50
176/176 [==============================] - 26s 149ms/step - loss: 1.8038 - accurac
y: 0.6904 - val_loss: 2.2136 - val_accuracy: 0.6172
Epoch 48/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7962 - accurac
y: 0.6890 - val_loss: 2.2128 - val_accuracy: 0.6170
Epoch 49/50
176/176 [==============================] - 26s 148ms/step - loss: 1.7911 - accurac
y: 0.6912 - val_loss: 2.2143 - val_accuracy: 0.6162
Epoch 50/50
176/176 [==============================] - 26s 149ms/step - loss: 1.7922 - accurac
y: 0.6936 - val_loss: 2.2109 - val_accuracy: 0.6172
```

In [93]:

```
plot_history(history6.history)
```



Save model

In [107]:

```
model_sgd_6.save('group114_pretrained_model.h5')
```

Evaluation

In [108]:

```
model_load = keras.models.load_model('group114_pretrained_model.h5')
```

In [109]:

```
loss, accuracy = model_load.evaluate(x_test, y_test, batch_size=32, verbose=1, sample_weight=None)
print("accuracy_sgd:",acc)
print("loss_sgd",loss)
```

```
313/313 [==============================] - 3s 10ms/step - loss: 2.1907 - accuracy: 0.6236
accuracy_sgd: 0.6229000091552734
loss_sgd 2.1907074451446533
```

In [110]:

```
pred_cnn = model_load.predict(x_test)
```

In [111]:

```
# Transform prob to one-hot
for i in range(len(pred_cnn)):
    max_value = max(pred_cnn[i])
    for j in range(len(pred_cnn[i])):
        if max_value == pred_cnn[i][j]:
            pred_cnn[i][j] = 1
        else:
            pred_cnn[i][j] = 0
```

In [112]:

```
from sklearn.metrics import classification_report
eval_cnn = classification_report(y_test, pred_cnn,output_dict = True)
```

In [113]:

```
print("CNN version evaluation: ")
print("\nAccuracy: " + str(acc))
print("\nPrecision: "+ str(eval_cnn["macro avg"]["precision"]))
print("\nRecall: "+ str(eval_cnn["macro avg"]["recall"]))
```

```
CNN version evaluation: 

Accuracy: 0.6229000091552734

Precision: 0.6410205738913081

Recall: 0.6236
```

In [114]:

```
# Transform one-hot to 0~100 numbers
from tensorflow.keras.utils import to_categorical

y_test_num = [np.argmax(i) for i in y_test]
pred_cnn_num = [np.argmax(i) for i in pred_cnn]
```

In [115]:

```
print("Before \n------------------------------------------------------------\n" + str(y_test[0:5]))
print("\nNow \n------------------------------------------------------------\n" + str(y_test_num[0:5]))
```

```
Before
------------------------------------------------------------
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0.]]

Now
------------------------------------------------------------
[49, 33, 72, 51, 71]
```

In [116]:

```
print("Before \n--------------------------------------------------------------------------------\n" + str(pred_cnn[
0:5]))
print("\nNow \n--------------------------------------------------------------------------------\n" + str(pred_cnn_n
um[0:5]))
```

```
Before
--------------------------------------------------------------------------------
[[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0.
  0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.
  0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0.]]

Now
--------------------------------------------------------------------------------
[90, 33, 55, 51, 71]
```

In [117]:

```
from sklearn.metrics import confusion_matrix

cm_cnn = confusion_matrix(y_test_num, pred_cnn_num)
```

In [118]:

```
fig = plt.figure(figsize=(12, 12),dpi = 60)
plt.matshow(cm_cnn,fignum=0)
plt.colorbar()
plt.xlabel('Predicted( CNN )')
plt.ylabel('True labels')
```

Out[118]:

Text(0, 0.5, 'True labels')