

main

March 22, 2023

```
[2]: # Import required packages
import numpy as np
import cv2
import pandas as pd
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from keras.callbacks import ModelCheckpoint
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D, Input, BatchNormalization, Activation, InputLayer
from keras.utils import np_utils
from sklearn.model_selection import KFold

import os
import time
```

0.1 1. Load the datasets

For the project, we provide a training set with 50000 images in the directory `../data/images/` with: - noisy labels for all images provided in `../data/noisy_label.csv`; - clean labels for the first 10000 images provided in `../data/clean_labels.csv`.

```
[3]: # [DO NOT MODIFY THIS CELL]

# load the images
n_img = 50000
n_noisy = 40000
n_clean_noisy = n_img - n_noisy
imgs = np.empty((n_img,32,32,3))
for i in range(n_img):
    img_fn = f'../data/images/{i+1:05d}.png'
    imgs[i,:,:,:]=cv2.cvtColor(cv2.imread(img_fn),cv2.COLOR_BGR2RGB)
```

```
# load the labels
clean_labels = np.genfromtxt('../data/clean_labels.csv', delimiter=',',
                             dtype="int8")
noisy_labels = np.genfromtxt('../data/noisy_labels.csv', delimiter=',',
                             dtype="int8")
```

For illustration, we present a small subset (of size 8) of the images with their clean and noisy labels in `clean_noisy_trainset`. You are encouraged to explore more characteristics of the label noises on the whole dataset.

```
[4]: # [DO NOT MODIFY THIS CELL]

fig = plt.figure()

ax1 = fig.add_subplot(2,4,1)
ax1.imshow(imgs[0]/255)
ax2 = fig.add_subplot(2,4,2)
ax2.imshow(imgs[1]/255)
ax3 = fig.add_subplot(2,4,3)
ax3.imshow(imgs[2]/255)
ax4 = fig.add_subplot(2,4,4)
ax4.imshow(imgs[3]/255)
ax1 = fig.add_subplot(2,4,5)
ax1.imshow(imgs[4]/255)
ax2 = fig.add_subplot(2,4,6)
ax2.imshow(imgs[5]/255)
ax3 = fig.add_subplot(2,4,7)
ax3.imshow(imgs[6]/255)
ax4 = fig.add_subplot(2,4,8)
ax4.imshow(imgs[7]/255)

# The class-label correspondence
classes = ('plane', 'car', 'bird', 'cat',
           'deer', 'dog', 'frog', 'horse', 'ship', 'truck')

# print clean labels
print('Clean labels:')
print(' '.join('%5s' % classes[clean_labels[j]] for j in range(8)))
# print noisy labels
print('Noisy labels:')
print(' '.join('%5s' % classes[noisy_labels[j]] for j in range(8)))
```

Clean labels:

```
frog truck truck deer car car bird horse
```

Noisy labels:

```
cat dog truck frog dog ship bird deer
```



0.2 2. The predictive model

We consider a baseline model directly on the noisy dataset without any label corrections. RGB histogram features are extracted to fit a logistic regression model.

0.2.1 2.1. Baseline Model

```
[5]: # [DO NOT MODIFY THIS CELL]
# RGB histogram dataset construction
no_bins = 6
bins = np.linspace(0,255,no_bins) # the range of the rgb histogram
target_vec = np.empty(n_img)
feature_mtx = np.empty((n_img,3*(len(bins)-1)))
i = 0
for i in range(n_img):
    # The target vector consists of noisy labels
    target_vec[i] = noisy_labels[i]

    # Use the numbers of pixels in each bin for all three channels as the
    ↪ features
    feature1 = np.histogram(imgs[i][:,:,0],bins=bins)[0]
    feature2 = np.histogram(imgs[i][:,:,1],bins=bins)[0]
    feature3 = np.histogram(imgs[i][:,:,2],bins=bins)[0]

    # Concatenate three features
    feature_mtx[i,:] = np.concatenate((feature1, feature2, feature3), axis=None)
    i += 1
```

```
[6]: # [DO NOT MODIFY THIS CELL]
# Train a logistic regression model
clf = LogisticRegression(random_state=0).fit(feature_mtx, target_vec)
```

For the convenience of evaluation, we write the following function `predictive_model` that does the label prediction. **For your predictive model, feel free to modify the function, but make sure the function takes an RGB image of `numpy.array` format with dimension $32 \times 32 \times 3$ as input, and returns one single label as output.**

```
[7]: # [DO NOT MODIFY THIS CELL]
def baseline_model(image):
    '''
    This is the baseline predictive model that takes in the image and returns a
    ↪ label prediction
    '''
    feature1 = np.histogram(image[:, :, 0], bins=bins)[0]
    feature2 = np.histogram(image[:, :, 1], bins=bins)[0]
    feature3 = np.histogram(image[:, :, 2], bins=bins)[0]
    feature = np.concatenate((feature1, feature2, feature3), axis=None).
    ↪ reshape(1, -1)
    return clf.predict(feature)
```

0.2.2 2.2. Model I

LeNet 5 is the one of most famous convolutional neural network designed to recognize handwritten digits from images.

It consists of 5 layers, including three convolutional layers, two subsampling layers, and two fully connected layers.

We add dropout to avoid overfitting

```
[8]: # [BUILD A MORE SOPHISTICATED PREDICTIVE MODEL]

# write your code here...

np.random.seed(5243_1)

# split the data into training and validation sets basing on the noisy labels
↪ and clean labels
imgs = imgs.astype('float32') / 255.0

# Clean data
x_clean = imgs[0:10000]
y_clean = clean_labels

# Noisy data
x_noisy = imgs[10000:]
```

```

y_noisy = noisy_labels[10000:]

x_train_noisy, x_val_noisy, y_train_noisy, y_val_noisy =   

    ↪train_test_split(imgs[10000:], noisy_labels[10000:], test_size=0.2,   

    ↪random_state=42)

x_train_clean, x_val_clean, y_train_clean, y_val_clean =   

    ↪train_test_split(imgs[0:10000], clean_labels, test_size=0.2, random_state=42)

```

```

[9]: # Model 1: LeNet 5
np.random.seed(5243_2)
model1 = Sequential()
model1.add(Conv2D(filters=6, kernel_size=(5,5), padding='same',   

    ↪activation='relu', input_shape=(32, 32, 3)))
model1.add(MaxPool2D(pool_size=(2,2)))
model1.add(Dropout(0.3))
model1.add(Conv2D(filters=16, kernel_size=(5,5), padding = 'valid' ,   

    ↪activation='relu'))
model1.add(MaxPool2D(pool_size=(2,2)))
model1.add(Dropout(0.3))
model1.add(Conv2D(filters=120, kernel_size=(5,5), padding = 'valid' ,   

    ↪activation='relu'))
model1.add(Flatten())
model1.add(Dropout(0.3))
model1.add(Dense(84, activation='relu'))
model1.add(Dropout(0.3))
model1.add(Dense(10, activation='softmax'))

model1.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy',   

    ↪metrics = ['accuracy'])

filepath='model1_weights.best.hdf5'
checkpoint = ModelCheckpoint(filepath, monitor='val_accuracy', verbose=1,   

    ↪save_best_only=True, mode='max')
callbacks_list = [checkpoint]

Model1_history_callback = model1.fit(x_train_noisy, y_train_noisy, batch_size =   

    ↪128, epochs = 40, validation_data = (x_val_noisy, y_val_noisy),   

    ↪callbacks=callbacks_list)

```

Epoch 1/40

250/250 [=====] - ETA: 0s - loss: 2.2972 - accuracy: 0.1161

Epoch 1: val_accuracy improved from -inf to 0.12962, saving model to model1_weights.best.hdf5

250/250 [=====] - 32s 119ms/step - loss: 2.2972 - accuracy: 0.1161 - val_loss: 2.2883 - val_accuracy: 0.1296

Epoch 2/40
250/250 [=====] - ETA: 0s - loss: 2.2740 - accuracy: 0.1480
Epoch 2: val_accuracy improved from 0.12962 to 0.14762, saving model to model1_weights.best.hdf5
250/250 [=====] - 23s 93ms/step - loss: 2.2740 - accuracy: 0.1480 - val_loss: 2.2742 - val_accuracy: 0.1476
Epoch 3/40
250/250 [=====] - ETA: 0s - loss: 2.2643 - accuracy: 0.1596
Epoch 3: val_accuracy improved from 0.14762 to 0.16688, saving model to model1_weights.best.hdf5
250/250 [=====] - 25s 99ms/step - loss: 2.2643 - accuracy: 0.1596 - val_loss: 2.2635 - val_accuracy: 0.1669
Epoch 4/40
250/250 [=====] - ETA: 0s - loss: 2.2561 - accuracy: 0.1700
Epoch 4: val_accuracy improved from 0.16688 to 0.16900, saving model to model1_weights.best.hdf5
250/250 [=====] - 21s 85ms/step - loss: 2.2561 - accuracy: 0.1700 - val_loss: 2.2620 - val_accuracy: 0.1690
Epoch 5/40
250/250 [=====] - ETA: 0s - loss: 2.2471 - accuracy: 0.1800
Epoch 5: val_accuracy improved from 0.16900 to 0.18125, saving model to model1_weights.best.hdf5
250/250 [=====] - 21s 83ms/step - loss: 2.2471 - accuracy: 0.1800 - val_loss: 2.2538 - val_accuracy: 0.1813
Epoch 6/40
250/250 [=====] - ETA: 0s - loss: 2.2426 - accuracy: 0.1845
Epoch 6: val_accuracy did not improve from 0.18125
250/250 [=====] - 20s 82ms/step - loss: 2.2426 - accuracy: 0.1845 - val_loss: 2.2611 - val_accuracy: 0.1731
Epoch 7/40
250/250 [=====] - ETA: 0s - loss: 2.2397 - accuracy: 0.1876
Epoch 7: val_accuracy improved from 0.18125 to 0.18462, saving model to model1_weights.best.hdf5
250/250 [=====] - 19s 76ms/step - loss: 2.2397 - accuracy: 0.1876 - val_loss: 2.2468 - val_accuracy: 0.1846
Epoch 8/40
250/250 [=====] - ETA: 0s - loss: 2.2345 - accuracy: 0.1911
Epoch 8: val_accuracy improved from 0.18462 to 0.19162, saving model to model1_weights.best.hdf5
250/250 [=====] - 19s 77ms/step - loss: 2.2345 - accuracy: 0.1911 - val_loss: 2.2432 - val_accuracy: 0.1916

Epoch 9/40
250/250 [=====] - ETA: 0s - loss: 2.2296 - accuracy: 0.1958
Epoch 9: val_accuracy improved from 0.19162 to 0.19613, saving model to model1_weights.best.hdf5
250/250 [=====] - 19s 78ms/step - loss: 2.2296 - accuracy: 0.1958 - val_loss: 2.2433 - val_accuracy: 0.1961
Epoch 10/40
250/250 [=====] - ETA: 0s - loss: 2.2264 - accuracy: 0.1987
Epoch 10: val_accuracy did not improve from 0.19613
250/250 [=====] - 20s 80ms/step - loss: 2.2264 - accuracy: 0.1987 - val_loss: 2.2402 - val_accuracy: 0.1950
Epoch 11/40
250/250 [=====] - ETA: 0s - loss: 2.2241 - accuracy: 0.2014
Epoch 11: val_accuracy did not improve from 0.19613
250/250 [=====] - 19s 75ms/step - loss: 2.2241 - accuracy: 0.2014 - val_loss: 2.2488 - val_accuracy: 0.1867
Epoch 12/40
250/250 [=====] - ETA: 0s - loss: 2.2196 - accuracy: 0.2044
Epoch 12: val_accuracy improved from 0.19613 to 0.20237, saving model to model1_weights.best.hdf5
250/250 [=====] - 19s 76ms/step - loss: 2.2196 - accuracy: 0.2044 - val_loss: 2.2313 - val_accuracy: 0.2024
Epoch 13/40
250/250 [=====] - ETA: 0s - loss: 2.2170 - accuracy: 0.2063
Epoch 13: val_accuracy improved from 0.20237 to 0.20263, saving model to model1_weights.best.hdf5
250/250 [=====] - 19s 78ms/step - loss: 2.2170 - accuracy: 0.2063 - val_loss: 2.2319 - val_accuracy: 0.2026
Epoch 14/40
250/250 [=====] - ETA: 0s - loss: 2.2112 - accuracy: 0.2130
Epoch 14: val_accuracy improved from 0.20263 to 0.20838, saving model to model1_weights.best.hdf5
250/250 [=====] - 21s 86ms/step - loss: 2.2112 - accuracy: 0.2130 - val_loss: 2.2265 - val_accuracy: 0.2084
Epoch 15/40
250/250 [=====] - ETA: 0s - loss: 2.2090 - accuracy: 0.2118
Epoch 15: val_accuracy did not improve from 0.20838
250/250 [=====] - 21s 86ms/step - loss: 2.2090 - accuracy: 0.2118 - val_loss: 2.2306 - val_accuracy: 0.2021
Epoch 16/40
250/250 [=====] - ETA: 0s - loss: 2.2073 - accuracy:

0.2137
Epoch 16: val_accuracy did not improve from 0.20838
250/250 [=====] - 21s 84ms/step - loss: 2.2073 -
accuracy: 0.2137 - val_loss: 2.2260 - val_accuracy: 0.2040
Epoch 17/40
250/250 [=====] - ETA: 0s - loss: 2.2031 - accuracy:
0.2173
Epoch 17: val_accuracy improved from 0.20838 to 0.21262, saving model to
model1_weights.best.hdf5
250/250 [=====] - 21s 84ms/step - loss: 2.2031 -
accuracy: 0.2173 - val_loss: 2.2220 - val_accuracy: 0.2126
Epoch 18/40
250/250 [=====] - ETA: 0s - loss: 2.1986 - accuracy:
0.2203
Epoch 18: val_accuracy did not improve from 0.21262
250/250 [=====] - 21s 84ms/step - loss: 2.1986 -
accuracy: 0.2203 - val_loss: 2.2227 - val_accuracy: 0.2125
Epoch 19/40
250/250 [=====] - ETA: 0s - loss: 2.1979 - accuracy:
0.2203
Epoch 19: val_accuracy did not improve from 0.21262
250/250 [=====] - 21s 82ms/step - loss: 2.1979 -
accuracy: 0.2203 - val_loss: 2.2253 - val_accuracy: 0.2121
Epoch 20/40
250/250 [=====] - ETA: 0s - loss: 2.1947 - accuracy:
0.2231
Epoch 20: val_accuracy did not improve from 0.21262
250/250 [=====] - 21s 84ms/step - loss: 2.1947 -
accuracy: 0.2231 - val_loss: 2.2214 - val_accuracy: 0.2115
Epoch 21/40
250/250 [=====] - ETA: 0s - loss: 2.1886 - accuracy:
0.2259
Epoch 21: val_accuracy did not improve from 0.21262
250/250 [=====] - 20s 81ms/step - loss: 2.1886 -
accuracy: 0.2259 - val_loss: 2.2238 - val_accuracy: 0.2113
Epoch 22/40
250/250 [=====] - ETA: 0s - loss: 2.1884 - accuracy:
0.2242
Epoch 22: val_accuracy did not improve from 0.21262
250/250 [=====] - 20s 82ms/step - loss: 2.1884 -
accuracy: 0.2242 - val_loss: 2.2247 - val_accuracy: 0.2094
Epoch 23/40
250/250 [=====] - ETA: 0s - loss: 2.1860 - accuracy:
0.2251
Epoch 23: val_accuracy did not improve from 0.21262
250/250 [=====] - 21s 82ms/step - loss: 2.1860 -
accuracy: 0.2251 - val_loss: 2.2260 - val_accuracy: 0.2075
Epoch 24/40

250/250 [=====] - ETA: 0s - loss: 2.1837 - accuracy: 0.2287
Epoch 24: val_accuracy improved from 0.21262 to 0.21325, saving model to model1_weights.best.hdf5
250/250 [=====] - 21s 83ms/step - loss: 2.1837 - accuracy: 0.2287 - val_loss: 2.2230 - val_accuracy: 0.2132
Epoch 25/40
250/250 [=====] - ETA: 0s - loss: 2.1801 - accuracy: 0.2290
Epoch 25: val_accuracy did not improve from 0.21325
250/250 [=====] - 21s 84ms/step - loss: 2.1801 - accuracy: 0.2290 - val_loss: 2.2243 - val_accuracy: 0.2111
Epoch 26/40
250/250 [=====] - ETA: 0s - loss: 2.1743 - accuracy: 0.2336
Epoch 26: val_accuracy did not improve from 0.21325
250/250 [=====] - 21s 83ms/step - loss: 2.1743 - accuracy: 0.2336 - val_loss: 2.2287 - val_accuracy: 0.2094
Epoch 27/40
250/250 [=====] - ETA: 0s - loss: 2.1749 - accuracy: 0.2333
Epoch 27: val_accuracy did not improve from 0.21325
250/250 [=====] - 21s 83ms/step - loss: 2.1749 - accuracy: 0.2333 - val_loss: 2.2236 - val_accuracy: 0.2062
Epoch 28/40
250/250 [=====] - ETA: 0s - loss: 2.1714 - accuracy: 0.2360
Epoch 28: val_accuracy improved from 0.21325 to 0.21387, saving model to model1_weights.best.hdf5
250/250 [=====] - 21s 83ms/step - loss: 2.1714 - accuracy: 0.2360 - val_loss: 2.2249 - val_accuracy: 0.2139
Epoch 29/40
250/250 [=====] - ETA: 0s - loss: 2.1717 - accuracy: 0.2358
Epoch 29: val_accuracy improved from 0.21387 to 0.21438, saving model to model1_weights.best.hdf5
250/250 [=====] - 21s 83ms/step - loss: 2.1717 - accuracy: 0.2358 - val_loss: 2.2245 - val_accuracy: 0.2144
Epoch 30/40
250/250 [=====] - ETA: 0s - loss: 2.1707 - accuracy: 0.2359
Epoch 30: val_accuracy did not improve from 0.21438
250/250 [=====] - 21s 82ms/step - loss: 2.1707 - accuracy: 0.2359 - val_loss: 2.2285 - val_accuracy: 0.2120
Epoch 31/40
250/250 [=====] - ETA: 0s - loss: 2.1675 - accuracy: 0.2366
Epoch 31: val_accuracy improved from 0.21438 to 0.21525, saving model to

```

model1_weights.best.hdf5
250/250 [=====] - 21s 83ms/step - loss: 2.1675 -
accuracy: 0.2366 - val_loss: 2.2239 - val_accuracy: 0.2153
Epoch 32/40
250/250 [=====] - ETA: 0s - loss: 2.1624 - accuracy:
0.2390
Epoch 32: val_accuracy improved from 0.21525 to 0.21612, saving model to
model1_weights.best.hdf5
250/250 [=====] - 21s 83ms/step - loss: 2.1624 -
accuracy: 0.2390 - val_loss: 2.2275 - val_accuracy: 0.2161
Epoch 33/40
250/250 [=====] - ETA: 0s - loss: 2.1641 - accuracy:
0.2400
Epoch 33: val_accuracy did not improve from 0.21612
250/250 [=====] - 21s 83ms/step - loss: 2.1641 -
accuracy: 0.2400 - val_loss: 2.2255 - val_accuracy: 0.2134
Epoch 34/40
250/250 [=====] - ETA: 0s - loss: 2.1655 - accuracy:
0.2345
Epoch 34: val_accuracy did not improve from 0.21612
250/250 [=====] - 21s 82ms/step - loss: 2.1655 -
accuracy: 0.2345 - val_loss: 2.2276 - val_accuracy: 0.2154
Epoch 35/40
250/250 [=====] - ETA: 0s - loss: 2.1573 - accuracy:
0.2396
Epoch 35: val_accuracy did not improve from 0.21612
250/250 [=====] - 21s 83ms/step - loss: 2.1573 -
accuracy: 0.2396 - val_loss: 2.2269 - val_accuracy: 0.2131
Epoch 36/40
250/250 [=====] - ETA: 0s - loss: 2.1546 - accuracy:
0.2413
Epoch 36: val_accuracy did not improve from 0.21612
250/250 [=====] - 21s 82ms/step - loss: 2.1546 -
accuracy: 0.2413 - val_loss: 2.2272 - val_accuracy: 0.2145
Epoch 37/40
250/250 [=====] - ETA: 0s - loss: 2.1523 - accuracy:
0.2420
Epoch 37: val_accuracy did not improve from 0.21612
250/250 [=====] - 20s 79ms/step - loss: 2.1523 -
accuracy: 0.2420 - val_loss: 2.2269 - val_accuracy: 0.2140
Epoch 38/40
250/250 [=====] - ETA: 0s - loss: 2.1540 - accuracy:
0.2419
Epoch 38: val_accuracy did not improve from 0.21612
250/250 [=====] - 19s 78ms/step - loss: 2.1540 -
accuracy: 0.2419 - val_loss: 2.2299 - val_accuracy: 0.2130
Epoch 39/40
250/250 [=====] - ETA: 0s - loss: 2.1537 - accuracy:

```

```

0.2424
Epoch 39: val_accuracy improved from 0.21612 to 0.21663, saving model to
model1_weights.best.hdf5
250/250 [=====] - 20s 78ms/step - loss: 2.1537 -
accuracy: 0.2424 - val_loss: 2.2264 - val_accuracy: 0.2166
Epoch 40/40
250/250 [=====] - ETA: 0s - loss: 2.1507 - accuracy:
0.2421
Epoch 40: val_accuracy did not improve from 0.21663
250/250 [=====] - 19s 78ms/step - loss: 2.1507 -
accuracy: 0.2421 - val_loss: 2.2314 - val_accuracy: 0.2118

```

```

[10]: # Load the best weights from the training

model1 = Sequential()
model1.add(Conv2D(filters=6, kernel_size=(5,5), padding='same',
    ↪activation='relu', input_shape=(32, 32, 3)))
model1.add(MaxPool2D(pool_size=(2,2)))
model1.add(Dropout(0.3))
model1.add(Conv2D(filters=16, kernel_size=(5,5), padding = 'valid' ,
    ↪activation='relu'))
model1.add(MaxPool2D(pool_size=(2,2)))
model1.add(Dropout(0.3))
model1.add(Conv2D(filters=120, kernel_size=(5,5), padding = 'valid' ,
    ↪activation='relu'))
model1.add(Flatten())
model1.add(Dropout(0.3))
model1.add(Dense(84, activation='relu'))
model1.add(Dropout(0.3))
model1.add(Dense(10, activation='softmax'))

model1.load_weights('model1_weights.best.hdf5')
model1.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy',
    ↪metrics = ['accuracy'])

# Evaluate the model on the test data
test_loss, test_acc = model1.evaluate(x_clean, y_clean, verbose=0)
print('Test accuracy:', test_acc, 'Test loss:', test_loss)

```

Test accuracy: 0.4742000102996826 Test loss: 1.8490393161773682

The accuracy of model 1 is 49%, which is better a lot than the baseline model.

```

[11]: def model_I(image):
    """
    This function should takes in the image of dimension 32*32*3 as input and
    ↪returns a label prediction
    """

```

```

# write your code here...

img=image.reshape(1,32,32,3)/255.0
prediction=model1.predict(img)
return np.argmax(prediction)

```

0.2.3 2.3. Model II

0.2.4 2.3.1 Label Correction

```

[12]: # [ADD WEAKLY SUPERVISED LEARNING FEATURE TO MODEL I]

# write your code here...
np.random.seed(5243_3)
label_correction_model =Sequential(
    [
        InputLayer(input_shape=(32, 32, 3)),
        Conv2D(filters=32, kernel_size=3, activation='relu'),
        MaxPool2D(pool_size=(2, 2), padding='same'),
        Dropout(0.3),
        Conv2D(filters=64, kernel_size=3, activation='relu'),
        BatchNormalization(momentum=0.9),
        MaxPool2D(pool_size=(2, 2), padding='same'),
        Dropout(0.3),
        Conv2D(filters=128, kernel_size=3, activation='relu'),
        MaxPool2D(pool_size=(2, 2), padding='same'),
        Dropout(0.3),
        Conv2D(filters=256, kernel_size=3, activation='relu'),
        BatchNormalization(momentum=0.9),
        Flatten(),
        Dropout(0.3),
        Dense(64, activation = 'relu'),
        Dropout(0.3),
        Dense(10, activation = 'softmax')
    ])
filepath='lcn_weights.best.hdf5'
checkpoint = ModelCheckpoint(filepath, monitor='val_accuracy', verbose=1,
    ↪save_best_only=True,mode='max')
callbacks_list_2 = [checkpoint]
label_correction_model.compile(optimizer=tf.keras.optimizers.
    ↪RMSprop(learning_rate=1e-3), loss='sparse_categorical_crossentropy',
    ↪metrics=['accuracy'])

label_correction_model_history_callback = label_correction_model.
    ↪fit(x_train_clean, y_train_clean, epochs=40, validation_data = (x_val_clean,
    ↪y_val_clean), callbacks=callbacks_list_2)

```

Epoch 1/40
249/250 [=====>.] - ETA: 0s - loss: 2.2085 - accuracy: 0.2383
Epoch 1: val_accuracy improved from -inf to 0.30750, saving model to lcn_weights.best.hdf5
250/250 [=====] - 13s 46ms/step - loss: 2.2072 - accuracy: 0.2385 - val_loss: 1.8975 - val_accuracy: 0.3075
Epoch 2/40
249/250 [=====>.] - ETA: 0s - loss: 1.7848 - accuracy: 0.3524
Epoch 2: val_accuracy improved from 0.30750 to 0.42550, saving model to lcn_weights.best.hdf5
250/250 [=====] - 11s 44ms/step - loss: 1.7828 - accuracy: 0.3531 - val_loss: 1.5434 - val_accuracy: 0.4255
Epoch 3/40
250/250 [=====] - ETA: 0s - loss: 1.6314 - accuracy: 0.4027
Epoch 3: val_accuracy improved from 0.42550 to 0.42850, saving model to lcn_weights.best.hdf5
250/250 [=====] - 11s 44ms/step - loss: 1.6314 - accuracy: 0.4027 - val_loss: 1.5961 - val_accuracy: 0.4285
Epoch 4/40
249/250 [=====>.] - ETA: 0s - loss: 1.5490 - accuracy: 0.4360
Epoch 4: val_accuracy did not improve from 0.42850
250/250 [=====] - 11s 44ms/step - loss: 1.5494 - accuracy: 0.4361 - val_loss: 1.8141 - val_accuracy: 0.4080
Epoch 5/40
250/250 [=====] - ETA: 0s - loss: 1.4477 - accuracy: 0.4820
Epoch 5: val_accuracy improved from 0.42850 to 0.49800, saving model to lcn_weights.best.hdf5
250/250 [=====] - 10s 41ms/step - loss: 1.4477 - accuracy: 0.4820 - val_loss: 1.4328 - val_accuracy: 0.4980
Epoch 6/40
248/250 [=====>.] - ETA: 0s - loss: 1.4179 - accuracy: 0.4992
Epoch 6: val_accuracy improved from 0.49800 to 0.53250, saving model to lcn_weights.best.hdf5
250/250 [=====] - 7s 26ms/step - loss: 1.4157 - accuracy: 0.4999 - val_loss: 1.2926 - val_accuracy: 0.5325
Epoch 7/40
249/250 [=====>.] - ETA: 0s - loss: 1.3613 - accuracy: 0.5139
Epoch 7: val_accuracy improved from 0.53250 to 0.54200, saving model to lcn_weights.best.hdf5
250/250 [=====] - 7s 27ms/step - loss: 1.3599 - accuracy: 0.5145 - val_loss: 1.2737 - val_accuracy: 0.5420

Epoch 8/40
250/250 [=====] - ETA: 0s - loss: 1.3024 - accuracy: 0.5319
Epoch 8: val_accuracy improved from 0.54200 to 0.56750, saving model to lcn_weights.best.hdf5
250/250 [=====] - 7s 28ms/step - loss: 1.3024 - accuracy: 0.5319 - val_loss: 1.1906 - val_accuracy: 0.5675
Epoch 9/40
250/250 [=====] - ETA: 0s - loss: 1.2730 - accuracy: 0.5486
Epoch 9: val_accuracy improved from 0.56750 to 0.57150, saving model to lcn_weights.best.hdf5
250/250 [=====] - 7s 28ms/step - loss: 1.2730 - accuracy: 0.5486 - val_loss: 1.2132 - val_accuracy: 0.5715
Epoch 10/40
249/250 [=====>.] - ETA: 0s - loss: 1.2366 - accuracy: 0.5650
Epoch 10: val_accuracy did not improve from 0.57150
250/250 [=====] - 7s 28ms/step - loss: 1.2363 - accuracy: 0.5648 - val_loss: 1.8800 - val_accuracy: 0.4440
Epoch 11/40
250/250 [=====] - ETA: 0s - loss: 1.2014 - accuracy: 0.5760
Epoch 11: val_accuracy did not improve from 0.57150
250/250 [=====] - 7s 28ms/step - loss: 1.2014 - accuracy: 0.5760 - val_loss: 1.2718 - val_accuracy: 0.5405
Epoch 12/40
249/250 [=====>.] - ETA: 0s - loss: 1.1732 - accuracy: 0.5910
Epoch 12: val_accuracy improved from 0.57150 to 0.62100, saving model to lcn_weights.best.hdf5
250/250 [=====] - 7s 28ms/step - loss: 1.1731 - accuracy: 0.5906 - val_loss: 1.0800 - val_accuracy: 0.6210
Epoch 13/40
250/250 [=====] - ETA: 0s - loss: 1.1583 - accuracy: 0.5911
Epoch 13: val_accuracy did not improve from 0.62100
250/250 [=====] - 7s 28ms/step - loss: 1.1583 - accuracy: 0.5911 - val_loss: 1.0800 - val_accuracy: 0.6170
Epoch 14/40
249/250 [=====>.] - ETA: 0s - loss: 1.1276 - accuracy: 0.6003
Epoch 14: val_accuracy did not improve from 0.62100
250/250 [=====] - 7s 28ms/step - loss: 1.1270 - accuracy: 0.6005 - val_loss: 1.1250 - val_accuracy: 0.6020
Epoch 15/40
249/250 [=====>.] - ETA: 0s - loss: 1.1055 - accuracy: 0.6077

Epoch 15: val_accuracy did not improve from 0.62100
250/250 [=====] - 7s 28ms/step - loss: 1.1055 -
accuracy: 0.6071 - val_loss: 1.1127 - val_accuracy: 0.6020
Epoch 16/40
250/250 [=====] - ETA: 0s - loss: 1.0963 - accuracy:
0.6189
Epoch 16: val_accuracy did not improve from 0.62100
250/250 [=====] - 7s 28ms/step - loss: 1.0963 -
accuracy: 0.6189 - val_loss: 1.3001 - val_accuracy: 0.5690
Epoch 17/40
250/250 [=====] - ETA: 0s - loss: 1.0554 - accuracy:
0.6311
Epoch 17: val_accuracy did not improve from 0.62100
250/250 [=====] - 7s 30ms/step - loss: 1.0554 -
accuracy: 0.6311 - val_loss: 1.1008 - val_accuracy: 0.6125
Epoch 18/40
249/250 [=====>.] - ETA: 0s - loss: 1.0343 - accuracy:
0.6398
Epoch 18: val_accuracy did not improve from 0.62100
250/250 [=====] - 7s 29ms/step - loss: 1.0331 -
accuracy: 0.6401 - val_loss: 1.0900 - val_accuracy: 0.6160
Epoch 19/40
248/250 [=====>.] - ETA: 0s - loss: 1.0229 - accuracy:
0.6421
Epoch 19: val_accuracy did not improve from 0.62100
250/250 [=====] - 7s 29ms/step - loss: 1.0239 -
accuracy: 0.6419 - val_loss: 1.0564 - val_accuracy: 0.6200
Epoch 20/40
249/250 [=====>.] - ETA: 0s - loss: 1.0219 - accuracy:
0.6468
Epoch 20: val_accuracy improved from 0.62100 to 0.62700, saving model to
lcn_weights.best.hdf5
250/250 [=====] - 7s 28ms/step - loss: 1.0227 -
accuracy: 0.6466 - val_loss: 1.0357 - val_accuracy: 0.6270
Epoch 21/40
249/250 [=====>.] - ETA: 0s - loss: 0.9914 - accuracy:
0.6532
Epoch 21: val_accuracy improved from 0.62700 to 0.64950, saving model to
lcn_weights.best.hdf5
250/250 [=====] - 7s 28ms/step - loss: 0.9905 -
accuracy: 0.6536 - val_loss: 1.0175 - val_accuracy: 0.6495
Epoch 22/40
249/250 [=====>.] - ETA: 0s - loss: 0.9903 - accuracy:
0.6598
Epoch 22: val_accuracy did not improve from 0.64950
250/250 [=====] - 7s 28ms/step - loss: 0.9893 -
accuracy: 0.6601 - val_loss: 1.1226 - val_accuracy: 0.6255
Epoch 23/40

250/250 [=====] - ETA: 0s - loss: 0.9744 - accuracy: 0.6622
Epoch 23: val_accuracy did not improve from 0.64950
250/250 [=====] - 7s 28ms/step - loss: 0.9744 - accuracy: 0.6622 - val_loss: 1.0649 - val_accuracy: 0.6385
Epoch 24/40
249/250 [=====>.] - ETA: 0s - loss: 0.9686 - accuracy: 0.6673
Epoch 24: val_accuracy did not improve from 0.64950
250/250 [=====] - 7s 28ms/step - loss: 0.9674 - accuracy: 0.6676 - val_loss: 1.3801 - val_accuracy: 0.5585
Epoch 25/40
248/250 [=====>.] - ETA: 0s - loss: 0.9461 - accuracy: 0.6694
Epoch 25: val_accuracy did not improve from 0.64950
250/250 [=====] - 7s 29ms/step - loss: 0.9453 - accuracy: 0.6701 - val_loss: 1.0810 - val_accuracy: 0.6375
Epoch 26/40
250/250 [=====] - ETA: 0s - loss: 0.9565 - accuracy: 0.6741
Epoch 26: val_accuracy did not improve from 0.64950
250/250 [=====] - 7s 28ms/step - loss: 0.9565 - accuracy: 0.6741 - val_loss: 1.1416 - val_accuracy: 0.6165
Epoch 27/40
249/250 [=====>.] - ETA: 0s - loss: 0.9208 - accuracy: 0.6827
Epoch 27: val_accuracy did not improve from 0.64950
250/250 [=====] - 7s 29ms/step - loss: 0.9206 - accuracy: 0.6826 - val_loss: 1.7852 - val_accuracy: 0.5250
Epoch 28/40
249/250 [=====>.] - ETA: 0s - loss: 0.9093 - accuracy: 0.6889
Epoch 28: val_accuracy improved from 0.64950 to 0.68350, saving model to lcn_weights.best.hdf5
250/250 [=====] - 8s 30ms/step - loss: 0.9092 - accuracy: 0.6888 - val_loss: 0.9237 - val_accuracy: 0.6835
Epoch 29/40
249/250 [=====>.] - ETA: 0s - loss: 0.8959 - accuracy: 0.6890
Epoch 29: val_accuracy did not improve from 0.68350
250/250 [=====] - 8s 31ms/step - loss: 0.8962 - accuracy: 0.6886 - val_loss: 0.9725 - val_accuracy: 0.6760
Epoch 30/40
249/250 [=====>.] - ETA: 0s - loss: 0.9070 - accuracy: 0.6855
Epoch 30: val_accuracy did not improve from 0.68350
250/250 [=====] - 8s 32ms/step - loss: 0.9064 - accuracy: 0.6852 - val_loss: 0.9256 - val_accuracy: 0.6815

Epoch 31/40
250/250 [=====] - ETA: 0s - loss: 0.9046 - accuracy: 0.6957
Epoch 31: val_accuracy did not improve from 0.68350
250/250 [=====] - 8s 32ms/step - loss: 0.9046 - accuracy: 0.6957 - val_loss: 0.9889 - val_accuracy: 0.6695
Epoch 32/40
250/250 [=====] - ETA: 0s - loss: 0.8869 - accuracy: 0.6982
Epoch 32: val_accuracy did not improve from 0.68350
250/250 [=====] - 8s 32ms/step - loss: 0.8869 - accuracy: 0.6982 - val_loss: 0.9825 - val_accuracy: 0.6660
Epoch 33/40
250/250 [=====] - ETA: 0s - loss: 0.8850 - accuracy: 0.7001
Epoch 33: val_accuracy did not improve from 0.68350
250/250 [=====] - 8s 31ms/step - loss: 0.8850 - accuracy: 0.7001 - val_loss: 0.9702 - val_accuracy: 0.6745
Epoch 34/40
250/250 [=====] - ETA: 0s - loss: 0.8647 - accuracy: 0.7074
Epoch 34: val_accuracy did not improve from 0.68350
250/250 [=====] - 8s 31ms/step - loss: 0.8647 - accuracy: 0.7074 - val_loss: 0.9868 - val_accuracy: 0.6775
Epoch 35/40
250/250 [=====] - ETA: 0s - loss: 0.8508 - accuracy: 0.7124
Epoch 35: val_accuracy did not improve from 0.68350
250/250 [=====] - 8s 31ms/step - loss: 0.8508 - accuracy: 0.7124 - val_loss: 0.9931 - val_accuracy: 0.6645
Epoch 36/40
249/250 [=====>.] - ETA: 0s - loss: 0.8534 - accuracy: 0.7123
Epoch 36: val_accuracy did not improve from 0.68350
250/250 [=====] - 8s 31ms/step - loss: 0.8521 - accuracy: 0.7125 - val_loss: 1.0741 - val_accuracy: 0.6395
Epoch 37/40
249/250 [=====>.] - ETA: 0s - loss: 0.8530 - accuracy: 0.7112
Epoch 37: val_accuracy improved from 0.68350 to 0.68700, saving model to lcn_weights.best.hdf5
250/250 [=====] - 8s 31ms/step - loss: 0.8524 - accuracy: 0.7114 - val_loss: 0.9179 - val_accuracy: 0.6870
Epoch 38/40
249/250 [=====>.] - ETA: 0s - loss: 0.8287 - accuracy: 0.7172
Epoch 38: val_accuracy did not improve from 0.68700
250/250 [=====] - 7s 30ms/step - loss: 0.8289 -

```

accuracy: 0.7175 - val_loss: 1.2276 - val_accuracy: 0.6260
Epoch 39/40
250/250 [=====] - ETA: 0s - loss: 0.8110 - accuracy:
0.7258
Epoch 39: val_accuracy did not improve from 0.68700
250/250 [=====] - 7s 30ms/step - loss: 0.8110 -
accuracy: 0.7258 - val_loss: 1.0867 - val_accuracy: 0.6585
Epoch 40/40
249/250 [=====>.] - ETA: 0s - loss: 0.8236 - accuracy:
0.7198
Epoch 40: val_accuracy improved from 0.68700 to 0.70500, saving model to
lcn_weights.best.hdf5
250/250 [=====] - 8s 30ms/step - loss: 0.8233 -
accuracy: 0.7196 - val_loss: 0.9852 - val_accuracy: 0.7050

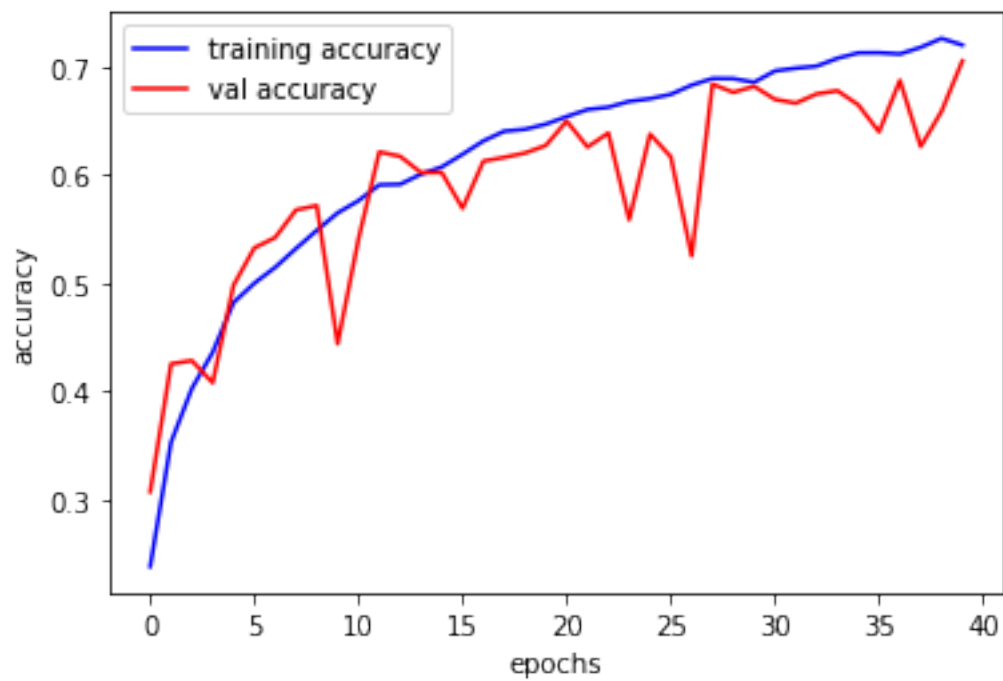
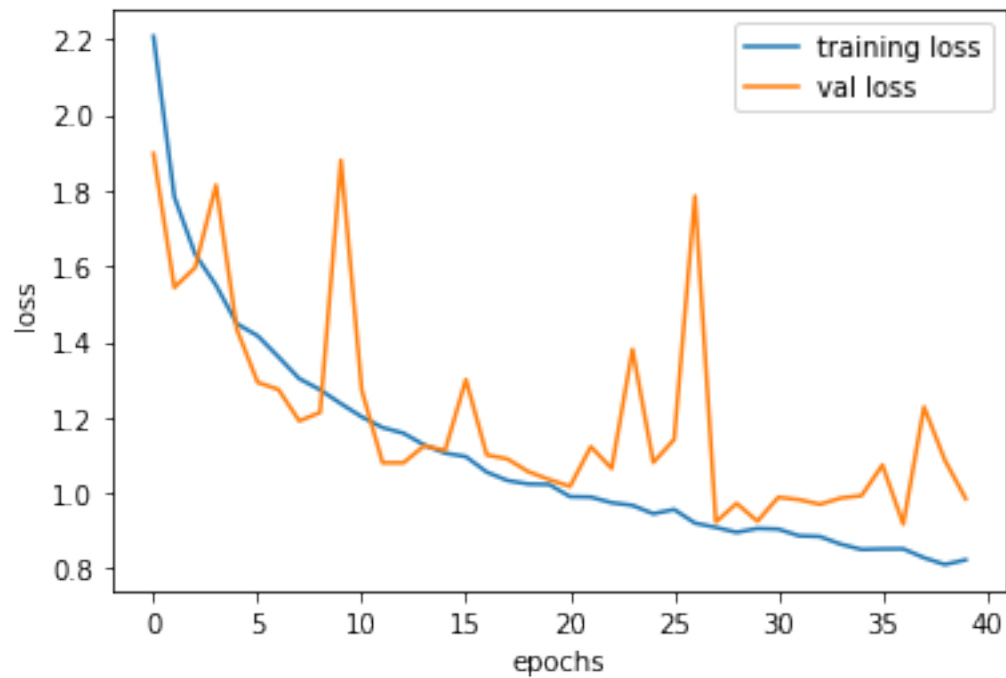
```

```

[13]: hist = pd.DataFrame(label_correction_model_history_callback.history)
# fig, ax = plt.subplots()
plt.plot(hist.index, hist['loss'], label='training loss')
plt.plot(hist.index, hist['val_loss'], label='val loss')
plt.legend()
plt.xlabel('epochs')
plt.ylabel('loss')
plt.show()

# ax2 = ax.twinx()
plt.plot(hist.index, hist['accuracy'], '-b', label='training accuracy')
plt.plot(hist.index, hist['val_accuracy'], '-r', label='val accuracy')
plt.legend()
plt.ylabel('accuracy')
plt.xlabel('epochs')
plt.show()

```



```
[14]: label_correction_model =Sequential(
      [
```

```

        InputLayer(input_shape=(32, 32, 3)),
        Conv2D(filters=32, kernel_size=3, activation='relu'),
        MaxPool2D(pool_size=(2, 2), padding='same'),
        Dropout(0.3),
        Conv2D(filters=64, kernel_size=3, activation='relu'),
        BatchNormalization(momentum=0.9),
        MaxPool2D(pool_size=(2, 2), padding='same'),
        Dropout(0.3),
        Conv2D(filters=128, kernel_size=3, activation='relu'),
        MaxPool2D(pool_size=(2, 2), padding='same'),
        Dropout(0.3),
        Conv2D(filters=256, kernel_size=3, activation='relu'),
        BatchNormalization(momentum=0.9),
        Flatten(),
        Dropout(0.3),
        Dense(64, activation = 'relu'),
        Dropout(0.3),
        Dense(10, activation = 'softmax')
    ])
label_correction_model.load_weights('lcn_weights.best.hdf5')
label_correction_model.compile(optimizer=tf.keras.optimizers.
    ↳RMSprop(learning_rate=1e-3), loss='sparse_categorical_crossentropy',
    ↳metrics=['accuracy'])

```

```
[15]: label_correction_model.evaluate(x_val_clean, y_val_clean)
```

```
63/63 [=====] - 1s 9ms/step - loss: 0.9852 - accuracy:
0.7050
```

```
[15]: [0.9852386116981506, 0.7049999833106995]
```

```

[16]: cleansed_y_labels_train = label_correction_model.predict(x_train_noisy)
      cleansed_y_labels_train = np.array([np.argmax(i) for i in
      ↳cleansed_y_labels_train])

      cleansed_y_labels_val = label_correction_model.predict(x_val_noisy)
      cleansed_y_labels_val = np.array([np.argmax(i) for i in cleansed_y_labels_val])

      cleansed_y_labels = label_correction_model.predict(x_noisy)
      cleansed_y_labels = np.array([np.argmax(i) for i in cleansed_y_labels])

```

```

1000/1000 [=====] - 8s 8ms/step
250/250 [=====] - 2s 8ms/step
1250/1250 [=====] - 10s 8ms/step

```

0.2.5 Training Model 1 with cleaned label

The accuracy of the model 2 on the clean data set is

```
[17]: def create_model2():
    np.random.seed(5243_3)
    model2 = Sequential()
    model2.add(Conv2D(filters=6, kernel_size=(5,5), padding='same',
    ↪activation='relu', input_shape=(32, 32, 3)))
    model2.add(MaxPool2D(pool_size=(2,2)))
    model2.add(Dropout(0.3))
    model2.add(Conv2D(filters=16, kernel_size=(5,5), padding = 'valid' ,
    ↪activation='relu'))
    model2.add(MaxPool2D(pool_size=(2,2)))
    model2.add(Dropout(0.3))
    model2.add(Conv2D(filters=120, kernel_size=(5,5), padding = 'valid' ,
    ↪activation='relu'))
    model2.add(Flatten())
    model2.add(Dropout(0.3))
    model2.add(Dense(84, activation='relu'))
    model2.add(Dropout(0.3))
    model2.add(Dense(10, activation='softmax'))

    model2.compile(optimizer = 'adam', loss =
    ↪'sparse_categorical_crossentropy', metrics = ['accuracy'])

    return model2
```

```
[18]: # Define the number of folds for cross-validation
num_folds = 5

# Define the K-Fold cross-validator
kfold = KFold(n_splits=num_folds, shuffle=True)

# Initialize arrays to store the validation accuracies and losses for each fold
val_accs = []
val_losses = []
h=[]

filepath='model2_weights.best.hdf5'
checkpoint = ModelCheckpoint(filepath, monitor='val_accuracy', verbose=1,
    ↪save_best_only=True,mode='max')
callbacks_list_3 = [checkpoint]

# Loop over the folds
for fold_idx, (train_indices, val_indices) in enumerate(kfold.split(x_noisy,
    ↪y_noisy)):
    print('Training on fold', fold_idx+1)

    # Create a new instance of the model for each fold
    model = create_model2()
```

```

# Compile the model
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])

# Fit the model to the training data for this fold
history = model.fit(x_noisy[train_indices],
cleansed_y_labels[train_indices], batch_size=128, epochs=40, verbose=1,
validation_data=(x_noisy[val_indices], cleansed_y_labels[val_indices]),
callbacks=callbacks_list_3)
h.append(history)

# Evaluate the model on the validation data for this fold
val_loss, val_acc = model.evaluate(x_clean, y_clean, verbose=0)
print('Validation accuracy:', val_acc, 'Validation loss:', val_loss)

# Store the validation accuracy and loss for this fold
val_accs.append(val_acc)
val_losses.append(val_loss)

# Print the average validation accuracy and loss over all folds
print('Average validation accuracy:', np.mean(val_accs))
print('Average validation loss:', np.mean(val_losses))

```

Training on fold 1

Epoch 1/40

250/250 [=====] - ETA: 0s - loss: 1.9758 - accuracy: 0.2646

Epoch 1: val_accuracy improved from 0.21663 to 0.37775, saving model to model1_weights.best.hdf5

250/250 [=====] - 14s 51ms/step - loss: 1.9758 - accuracy: 0.2646 - val_loss: 1.7848 - val_accuracy: 0.3778

Epoch 2/40

250/250 [=====] - ETA: 0s - loss: 1.6078 - accuracy: 0.4094

Epoch 2: val_accuracy improved from 0.37775 to 0.50287, saving model to model1_weights.best.hdf5

250/250 [=====] - 13s 52ms/step - loss: 1.6078 - accuracy: 0.4094 - val_loss: 1.3911 - val_accuracy: 0.5029

Epoch 3/40

249/250 [=====>.] - ETA: 0s - loss: 1.4200 - accuracy: 0.4837

Epoch 3: val_accuracy improved from 0.50287 to 0.52950, saving model to model1_weights.best.hdf5

250/250 [=====] - 13s 53ms/step - loss: 1.4199 - accuracy: 0.4837 - val_loss: 1.2686 - val_accuracy: 0.5295

Epoch 4/40

249/250 [=====>.] - ETA: 0s - loss: 1.3394 - accuracy: 0.5132
Epoch 4: val_accuracy improved from 0.52950 to 0.54562, saving model to model1_weights.best.hdf5
250/250 [=====] - 13s 51ms/step - loss: 1.3386 - accuracy: 0.5135 - val_loss: 1.2595 - val_accuracy: 0.5456
Epoch 5/40
250/250 [=====] - ETA: 0s - loss: 1.2690 - accuracy: 0.5346
Epoch 5: val_accuracy improved from 0.54562 to 0.58200, saving model to model1_weights.best.hdf5
250/250 [=====] - 13s 52ms/step - loss: 1.2690 - accuracy: 0.5346 - val_loss: 1.1420 - val_accuracy: 0.5820
Epoch 6/40
250/250 [=====] - ETA: 0s - loss: 1.2199 - accuracy: 0.5525
Epoch 6: val_accuracy improved from 0.58200 to 0.60925, saving model to model1_weights.best.hdf5
250/250 [=====] - 14s 55ms/step - loss: 1.2199 - accuracy: 0.5525 - val_loss: 1.0806 - val_accuracy: 0.6093
Epoch 7/40
249/250 [=====>.] - ETA: 0s - loss: 1.1843 - accuracy: 0.5714
Epoch 7: val_accuracy did not improve from 0.60925
250/250 [=====] - 13s 53ms/step - loss: 1.1838 - accuracy: 0.5715 - val_loss: 1.0683 - val_accuracy: 0.6085
Epoch 8/40
250/250 [=====] - ETA: 0s - loss: 1.1672 - accuracy: 0.5727
Epoch 8: val_accuracy did not improve from 0.60925
250/250 [=====] - 13s 53ms/step - loss: 1.1672 - accuracy: 0.5727 - val_loss: 1.1161 - val_accuracy: 0.5957
Epoch 9/40
250/250 [=====] - ETA: 0s - loss: 1.1251 - accuracy: 0.5922
Epoch 9: val_accuracy improved from 0.60925 to 0.63375, saving model to model1_weights.best.hdf5
250/250 [=====] - 13s 53ms/step - loss: 1.1251 - accuracy: 0.5922 - val_loss: 1.0029 - val_accuracy: 0.6338
Epoch 10/40
250/250 [=====] - ETA: 0s - loss: 1.1086 - accuracy: 0.5966
Epoch 10: val_accuracy did not improve from 0.63375
250/250 [=====] - 13s 53ms/step - loss: 1.1086 - accuracy: 0.5966 - val_loss: 1.0382 - val_accuracy: 0.6226
Epoch 11/40
250/250 [=====] - ETA: 0s - loss: 1.0971 - accuracy: 0.5985

Epoch 11: val_accuracy did not improve from 0.63375
250/250 [=====] - 13s 53ms/step - loss: 1.0971 -
accuracy: 0.5985 - val_loss: 0.9954 - val_accuracy: 0.6336

Epoch 12/40
250/250 [=====] - ETA: 0s - loss: 1.0754 - accuracy:
0.6092

Epoch 12: val_accuracy improved from 0.63375 to 0.63813, saving model to
model1_weights.best.hdf5
250/250 [=====] - 14s 56ms/step - loss: 1.0754 -
accuracy: 0.6092 - val_loss: 1.0237 - val_accuracy: 0.6381

Epoch 13/40
249/250 [=====>.] - ETA: 0s - loss: 1.0615 - accuracy:
0.6139

Epoch 13: val_accuracy improved from 0.63813 to 0.64388, saving model to
model1_weights.best.hdf5
250/250 [=====] - 14s 54ms/step - loss: 1.0622 -
accuracy: 0.6136 - val_loss: 0.9873 - val_accuracy: 0.6439

Epoch 14/40
250/250 [=====] - ETA: 0s - loss: 1.0445 - accuracy:
0.6205

Epoch 14: val_accuracy did not improve from 0.64388
250/250 [=====] - 13s 53ms/step - loss: 1.0445 -
accuracy: 0.6205 - val_loss: 0.9867 - val_accuracy: 0.6369

Epoch 15/40
249/250 [=====>.] - ETA: 0s - loss: 1.0393 - accuracy:
0.6201

Epoch 15: val_accuracy improved from 0.64388 to 0.66725, saving model to
model1_weights.best.hdf5
250/250 [=====] - 13s 53ms/step - loss: 1.0397 -
accuracy: 0.6199 - val_loss: 0.9000 - val_accuracy: 0.6672

Epoch 16/40
249/250 [=====>.] - ETA: 0s - loss: 1.0235 - accuracy:
0.6272

Epoch 16: val_accuracy did not improve from 0.66725
250/250 [=====] - 13s 53ms/step - loss: 1.0230 -
accuracy: 0.6274 - val_loss: 0.9586 - val_accuracy: 0.6444

Epoch 17/40
250/250 [=====] - ETA: 0s - loss: 1.0050 - accuracy:
0.6347

Epoch 17: val_accuracy did not improve from 0.66725
250/250 [=====] - 13s 53ms/step - loss: 1.0050 -
accuracy: 0.6347 - val_loss: 0.9455 - val_accuracy: 0.6471

Epoch 18/40
250/250 [=====] - ETA: 0s - loss: 0.9996 - accuracy:
0.6361

Epoch 18: val_accuracy did not improve from 0.66725
250/250 [=====] - 14s 55ms/step - loss: 0.9996 -
accuracy: 0.6361 - val_loss: 0.9201 - val_accuracy: 0.6664

Epoch 19/40
250/250 [=====] - ETA: 0s - loss: 0.9936 - accuracy: 0.6392
Epoch 19: val_accuracy improved from 0.66725 to 0.66775, saving model to model1_weights.best.hdf5
250/250 [=====] - 14s 56ms/step - loss: 0.9936 - accuracy: 0.6392 - val_loss: 0.9220 - val_accuracy: 0.6678
Epoch 20/40
250/250 [=====] - ETA: 0s - loss: 0.9825 - accuracy: 0.6433
Epoch 20: val_accuracy improved from 0.66775 to 0.67400, saving model to model1_weights.best.hdf5
250/250 [=====] - 14s 56ms/step - loss: 0.9825 - accuracy: 0.6433 - val_loss: 0.8830 - val_accuracy: 0.6740
Epoch 21/40
250/250 [=====] - ETA: 0s - loss: 0.9665 - accuracy: 0.6478
Epoch 21: val_accuracy improved from 0.67400 to 0.67862, saving model to model1_weights.best.hdf5
250/250 [=====] - 14s 55ms/step - loss: 0.9665 - accuracy: 0.6478 - val_loss: 0.8812 - val_accuracy: 0.6786
Epoch 22/40
250/250 [=====] - ETA: 0s - loss: 0.9611 - accuracy: 0.6506
Epoch 22: val_accuracy did not improve from 0.67862
250/250 [=====] - 13s 53ms/step - loss: 0.9611 - accuracy: 0.6506 - val_loss: 0.8622 - val_accuracy: 0.6781
Epoch 23/40
249/250 [=====>.] - ETA: 0s - loss: 0.9569 - accuracy: 0.6517
Epoch 23: val_accuracy did not improve from 0.67862
250/250 [=====] - 13s 53ms/step - loss: 0.9568 - accuracy: 0.6520 - val_loss: 0.8880 - val_accuracy: 0.6784
Epoch 24/40
250/250 [=====] - ETA: 0s - loss: 0.9592 - accuracy: 0.6489
Epoch 24: val_accuracy did not improve from 0.67862
250/250 [=====] - 14s 55ms/step - loss: 0.9592 - accuracy: 0.6489 - val_loss: 0.8982 - val_accuracy: 0.6629
Epoch 25/40
250/250 [=====] - ETA: 0s - loss: 0.9392 - accuracy: 0.6566
Epoch 25: val_accuracy did not improve from 0.67862
250/250 [=====] - 14s 56ms/step - loss: 0.9392 - accuracy: 0.6566 - val_loss: 0.8761 - val_accuracy: 0.6783
Epoch 26/40
249/250 [=====>.] - ETA: 0s - loss: 0.9337 - accuracy: 0.6575

Epoch 26: val_accuracy did not improve from 0.67862
250/250 [=====] - 14s 55ms/step - loss: 0.9341 - accuracy: 0.6573 - val_loss: 0.9439 - val_accuracy: 0.6541
Epoch 27/40
249/250 [=====>.] - ETA: 0s - loss: 0.9384 - accuracy: 0.6585
Epoch 27: val_accuracy did not improve from 0.67862
250/250 [=====] - 13s 53ms/step - loss: 0.9378 - accuracy: 0.6586 - val_loss: 0.8793 - val_accuracy: 0.6756
Epoch 28/40
250/250 [=====] - ETA: 0s - loss: 0.9273 - accuracy: 0.6610
Epoch 28: val_accuracy improved from 0.67862 to 0.68388, saving model to model1_weights.best.hdf5
250/250 [=====] - 14s 54ms/step - loss: 0.9273 - accuracy: 0.6610 - val_loss: 0.8531 - val_accuracy: 0.6839
Epoch 29/40
250/250 [=====] - ETA: 0s - loss: 0.9251 - accuracy: 0.6617
Epoch 29: val_accuracy improved from 0.68388 to 0.68500, saving model to model1_weights.best.hdf5
250/250 [=====] - 14s 56ms/step - loss: 0.9251 - accuracy: 0.6617 - val_loss: 0.8680 - val_accuracy: 0.6850
Epoch 30/40
250/250 [=====] - ETA: 0s - loss: 0.9085 - accuracy: 0.6689
Epoch 30: val_accuracy did not improve from 0.68500
250/250 [=====] - 14s 58ms/step - loss: 0.9085 - accuracy: 0.6689 - val_loss: 0.9029 - val_accuracy: 0.6690
Epoch 31/40
250/250 [=====] - ETA: 0s - loss: 0.9051 - accuracy: 0.6690
Epoch 31: val_accuracy did not improve from 0.68500
250/250 [=====] - 14s 56ms/step - loss: 0.9051 - accuracy: 0.6690 - val_loss: 0.9012 - val_accuracy: 0.6607
Epoch 32/40
250/250 [=====] - ETA: 0s - loss: 0.9043 - accuracy: 0.6676
Epoch 32: val_accuracy improved from 0.68500 to 0.69375, saving model to model1_weights.best.hdf5
250/250 [=====] - 14s 56ms/step - loss: 0.9043 - accuracy: 0.6676 - val_loss: 0.8345 - val_accuracy: 0.6938
Epoch 33/40
250/250 [=====] - ETA: 0s - loss: 0.8956 - accuracy: 0.6727
Epoch 33: val_accuracy improved from 0.69375 to 0.69650, saving model to model1_weights.best.hdf5
250/250 [=====] - 14s 56ms/step - loss: 0.8956 -

```

accuracy: 0.6727 - val_loss: 0.8073 - val_accuracy: 0.6965
Epoch 34/40
250/250 [=====] - ETA: 0s - loss: 0.8991 - accuracy:
0.6755
Epoch 34: val_accuracy did not improve from 0.69650
250/250 [=====] - 14s 56ms/step - loss: 0.8991 -
accuracy: 0.6755 - val_loss: 0.8984 - val_accuracy: 0.6660
Epoch 35/40
250/250 [=====] - ETA: 0s - loss: 0.8897 - accuracy:
0.6773
Epoch 35: val_accuracy improved from 0.69650 to 0.69975, saving model to
model1_weights.best.hdf5
250/250 [=====] - 14s 56ms/step - loss: 0.8897 -
accuracy: 0.6773 - val_loss: 0.8177 - val_accuracy: 0.6998
Epoch 36/40
250/250 [=====] - ETA: 0s - loss: 0.8968 - accuracy:
0.6724
Epoch 36: val_accuracy did not improve from 0.69975
250/250 [=====] - 13s 53ms/step - loss: 0.8968 -
accuracy: 0.6724 - val_loss: 0.8064 - val_accuracy: 0.6992
Epoch 37/40
250/250 [=====] - ETA: 0s - loss: 0.8838 - accuracy:
0.6745
Epoch 37: val_accuracy did not improve from 0.69975
250/250 [=====] - 15s 59ms/step - loss: 0.8838 -
accuracy: 0.6745 - val_loss: 0.8301 - val_accuracy: 0.6846
Epoch 38/40
249/250 [=====>.] - ETA: 0s - loss: 0.8877 - accuracy:
0.6739
Epoch 38: val_accuracy did not improve from 0.69975
250/250 [=====] - 14s 55ms/step - loss: 0.8880 -
accuracy: 0.6739 - val_loss: 0.9018 - val_accuracy: 0.6736
Epoch 39/40
250/250 [=====] - ETA: 0s - loss: 0.8809 - accuracy:
0.6781
Epoch 39: val_accuracy did not improve from 0.69975
250/250 [=====] - 13s 54ms/step - loss: 0.8809 -
accuracy: 0.6781 - val_loss: 0.8316 - val_accuracy: 0.6876
Epoch 40/40
250/250 [=====] - ETA: 0s - loss: 0.8821 - accuracy:
0.6789
Epoch 40: val_accuracy did not improve from 0.69975
250/250 [=====] - 14s 55ms/step - loss: 0.8821 -
accuracy: 0.6789 - val_loss: 0.8448 - val_accuracy: 0.6876
Validation accuracy: 0.6159999966621399 Validation loss: 1.1784030199050903
Training on fold 2
Epoch 1/40
250/250 [=====] - ETA: 0s - loss: 1.8968 - accuracy:

```

0.2961
Epoch 1: val_accuracy did not improve from 0.69975
250/250 [=====] - 14s 50ms/step - loss: 1.8968 -
accuracy: 0.2961 - val_loss: 1.4798 - val_accuracy: 0.4730
Epoch 2/40
250/250 [=====] - ETA: 0s - loss: 1.4903 - accuracy:
0.4562
Epoch 2: val_accuracy did not improve from 0.69975
250/250 [=====] - 13s 53ms/step - loss: 1.4903 -
accuracy: 0.4562 - val_loss: 1.3117 - val_accuracy: 0.5163
Epoch 3/40
249/250 [=====>.] - ETA: 0s - loss: 1.3648 - accuracy:
0.5006
Epoch 3: val_accuracy did not improve from 0.69975
250/250 [=====] - 13s 53ms/step - loss: 1.3640 -
accuracy: 0.5008 - val_loss: 1.2244 - val_accuracy: 0.5505
Epoch 4/40
249/250 [=====>.] - ETA: 0s - loss: 1.2959 - accuracy:
0.5274
Epoch 4: val_accuracy did not improve from 0.69975
250/250 [=====] - 14s 56ms/step - loss: 1.2956 -
accuracy: 0.5275 - val_loss: 1.1502 - val_accuracy: 0.5854
Epoch 5/40
250/250 [=====] - ETA: 0s - loss: 1.2441 - accuracy:
0.5491
Epoch 5: val_accuracy did not improve from 0.69975
250/250 [=====] - 13s 54ms/step - loss: 1.2441 -
accuracy: 0.5491 - val_loss: 1.0956 - val_accuracy: 0.6089
Epoch 6/40
250/250 [=====] - ETA: 0s - loss: 1.1959 - accuracy:
0.5700
Epoch 6: val_accuracy did not improve from 0.69975
250/250 [=====] - 13s 54ms/step - loss: 1.1959 -
accuracy: 0.5700 - val_loss: 1.0684 - val_accuracy: 0.6219
Epoch 7/40
249/250 [=====>.] - ETA: 0s - loss: 1.1595 - accuracy:
0.5776
Epoch 7: val_accuracy did not improve from 0.69975
250/250 [=====] - 13s 52ms/step - loss: 1.1590 -
accuracy: 0.5778 - val_loss: 1.0250 - val_accuracy: 0.6336
Epoch 8/40
250/250 [=====] - ETA: 0s - loss: 1.1255 - accuracy:
0.5896
Epoch 8: val_accuracy did not improve from 0.69975
250/250 [=====] - 13s 53ms/step - loss: 1.1255 -
accuracy: 0.5896 - val_loss: 1.0853 - val_accuracy: 0.6160
Epoch 9/40
250/250 [=====] - ETA: 0s - loss: 1.1053 - accuracy:

0.5966
Epoch 9: val_accuracy did not improve from 0.69975
250/250 [=====] - 14s 54ms/step - loss: 1.1053 -
accuracy: 0.5966 - val_loss: 0.9656 - val_accuracy: 0.6538
Epoch 10/40
249/250 [=====>.] - ETA: 0s - loss: 1.0768 - accuracy:
0.6104
Epoch 10: val_accuracy did not improve from 0.69975
250/250 [=====] - 13s 53ms/step - loss: 1.0770 -
accuracy: 0.6104 - val_loss: 0.9376 - val_accuracy: 0.6660
Epoch 11/40
249/250 [=====>.] - ETA: 0s - loss: 1.0636 - accuracy:
0.6133
Epoch 11: val_accuracy did not improve from 0.69975
250/250 [=====] - 14s 54ms/step - loss: 1.0638 -
accuracy: 0.6133 - val_loss: 0.9214 - val_accuracy: 0.6661
Epoch 12/40
250/250 [=====] - ETA: 0s - loss: 1.0360 - accuracy:
0.6216
Epoch 12: val_accuracy did not improve from 0.69975
250/250 [=====] - 15s 58ms/step - loss: 1.0360 -
accuracy: 0.6216 - val_loss: 0.9008 - val_accuracy: 0.6724
Epoch 13/40
250/250 [=====] - ETA: 0s - loss: 1.0189 - accuracy:
0.6296
Epoch 13: val_accuracy did not improve from 0.69975
250/250 [=====] - 17s 69ms/step - loss: 1.0189 -
accuracy: 0.6296 - val_loss: 0.9050 - val_accuracy: 0.6704
Epoch 14/40
250/250 [=====] - ETA: 0s - loss: 1.0142 - accuracy:
0.6314
Epoch 14: val_accuracy did not improve from 0.69975
250/250 [=====] - 17s 67ms/step - loss: 1.0142 -
accuracy: 0.6314 - val_loss: 0.8668 - val_accuracy: 0.6852
Epoch 15/40
250/250 [=====] - ETA: 0s - loss: 1.0054 - accuracy:
0.6386
Epoch 15: val_accuracy did not improve from 0.69975
250/250 [=====] - 15s 62ms/step - loss: 1.0054 -
accuracy: 0.6386 - val_loss: 0.9213 - val_accuracy: 0.6746
Epoch 16/40
250/250 [=====] - ETA: 0s - loss: 0.9928 - accuracy:
0.6383
Epoch 16: val_accuracy did not improve from 0.69975
250/250 [=====] - 14s 58ms/step - loss: 0.9928 -
accuracy: 0.6383 - val_loss: 0.9272 - val_accuracy: 0.6654
Epoch 17/40
250/250 [=====] - ETA: 0s - loss: 0.9696 - accuracy:

0.6453
Epoch 17: val_accuracy did not improve from 0.69975
250/250 [=====] - 15s 59ms/step - loss: 0.9696 - accuracy: 0.6453 - val_loss: 0.8384 - val_accuracy: 0.6923
Epoch 18/40
250/250 [=====] - ETA: 0s - loss: 0.9637 - accuracy: 0.6489
Epoch 18: val_accuracy did not improve from 0.69975
250/250 [=====] - 15s 60ms/step - loss: 0.9637 - accuracy: 0.6489 - val_loss: 0.8365 - val_accuracy: 0.6985
Epoch 19/40
250/250 [=====] - ETA: 0s - loss: 0.9616 - accuracy: 0.6492
Epoch 19: val_accuracy did not improve from 0.69975
250/250 [=====] - 15s 62ms/step - loss: 0.9616 - accuracy: 0.6492 - val_loss: 0.8977 - val_accuracy: 0.6712
Epoch 20/40
250/250 [=====] - ETA: 0s - loss: 0.9506 - accuracy: 0.6538
Epoch 20: val_accuracy did not improve from 0.69975
250/250 [=====] - 15s 62ms/step - loss: 0.9506 - accuracy: 0.6538 - val_loss: 0.8257 - val_accuracy: 0.6996
Epoch 21/40
250/250 [=====] - ETA: 0s - loss: 0.9473 - accuracy: 0.6562
Epoch 21: val_accuracy did not improve from 0.69975
250/250 [=====] - 14s 57ms/step - loss: 0.9473 - accuracy: 0.6562 - val_loss: 0.8778 - val_accuracy: 0.6867
Epoch 22/40
250/250 [=====] - ETA: 0s - loss: 0.9344 - accuracy: 0.6570
Epoch 22: val_accuracy did not improve from 0.69975
250/250 [=====] - 14s 57ms/step - loss: 0.9344 - accuracy: 0.6570 - val_loss: 0.8331 - val_accuracy: 0.6956
Epoch 23/40
250/250 [=====] - ETA: 0s - loss: 0.9249 - accuracy: 0.6633
Epoch 23: val_accuracy did not improve from 0.69975
250/250 [=====] - 15s 60ms/step - loss: 0.9249 - accuracy: 0.6633 - val_loss: 0.8227 - val_accuracy: 0.6984
Epoch 24/40
250/250 [=====] - ETA: 0s - loss: 0.9267 - accuracy: 0.6624
Epoch 24: val_accuracy did not improve from 0.69975
250/250 [=====] - 15s 60ms/step - loss: 0.9267 - accuracy: 0.6624 - val_loss: 0.8532 - val_accuracy: 0.6894
Epoch 25/40
250/250 [=====] - ETA: 0s - loss: 0.9226 - accuracy:

0.6645
Epoch 25: val_accuracy improved from 0.69975 to 0.71188, saving model to
model1_weights.best.hdf5
250/250 [=====] - 16s 64ms/step - loss: 0.9226 -
accuracy: 0.6645 - val_loss: 0.7939 - val_accuracy: 0.7119
Epoch 26/40
250/250 [=====] - ETA: 0s - loss: 0.9157 - accuracy:
0.6664
Epoch 26: val_accuracy improved from 0.71188 to 0.71375, saving model to
model1_weights.best.hdf5
250/250 [=====] - 15s 61ms/step - loss: 0.9157 -
accuracy: 0.6664 - val_loss: 0.7908 - val_accuracy: 0.7138
Epoch 27/40
250/250 [=====] - ETA: 0s - loss: 0.9119 - accuracy:
0.6677
Epoch 27: val_accuracy did not improve from 0.71375
250/250 [=====] - 14s 58ms/step - loss: 0.9119 -
accuracy: 0.6677 - val_loss: 0.8020 - val_accuracy: 0.7122
Epoch 28/40
250/250 [=====] - ETA: 0s - loss: 0.9111 - accuracy:
0.6705
Epoch 28: val_accuracy did not improve from 0.71375
250/250 [=====] - 13s 54ms/step - loss: 0.9111 -
accuracy: 0.6705 - val_loss: 0.8014 - val_accuracy: 0.7055
Epoch 29/40
250/250 [=====] - ETA: 0s - loss: 0.9021 - accuracy:
0.6708
Epoch 29: val_accuracy did not improve from 0.71375
250/250 [=====] - 13s 54ms/step - loss: 0.9021 -
accuracy: 0.6708 - val_loss: 0.8293 - val_accuracy: 0.6938
Epoch 30/40
250/250 [=====] - ETA: 0s - loss: 0.8967 - accuracy:
0.6702
Epoch 30: val_accuracy improved from 0.71375 to 0.71412, saving model to
model1_weights.best.hdf5
250/250 [=====] - 15s 60ms/step - loss: 0.8967 -
accuracy: 0.6702 - val_loss: 0.7828 - val_accuracy: 0.7141
Epoch 31/40
250/250 [=====] - ETA: 0s - loss: 0.8935 - accuracy:
0.6738
Epoch 31: val_accuracy improved from 0.71412 to 0.72038, saving model to
model1_weights.best.hdf5
250/250 [=====] - 15s 59ms/step - loss: 0.8935 -
accuracy: 0.6738 - val_loss: 0.7714 - val_accuracy: 0.7204
Epoch 32/40
250/250 [=====] - ETA: 0s - loss: 0.8830 - accuracy:
0.6779
Epoch 32: val_accuracy did not improve from 0.72038

250/250 [=====] - 14s 56ms/step - loss: 0.8830 - accuracy: 0.6779 - val_loss: 0.7857 - val_accuracy: 0.7054
Epoch 33/40
250/250 [=====] - ETA: 0s - loss: 0.8926 - accuracy: 0.6728
Epoch 33: val_accuracy did not improve from 0.72038
250/250 [=====] - 15s 58ms/step - loss: 0.8926 - accuracy: 0.6728 - val_loss: 0.8323 - val_accuracy: 0.7030
Epoch 34/40
250/250 [=====] - ETA: 0s - loss: 0.8818 - accuracy: 0.6769
Epoch 34: val_accuracy did not improve from 0.72038
250/250 [=====] - 14s 58ms/step - loss: 0.8818 - accuracy: 0.6769 - val_loss: 0.8014 - val_accuracy: 0.7014
Epoch 35/40
250/250 [=====] - ETA: 0s - loss: 0.8776 - accuracy: 0.6773
Epoch 35: val_accuracy did not improve from 0.72038
250/250 [=====] - 15s 60ms/step - loss: 0.8776 - accuracy: 0.6773 - val_loss: 0.7768 - val_accuracy: 0.7163
Epoch 36/40
250/250 [=====] - ETA: 0s - loss: 0.8764 - accuracy: 0.6782
Epoch 36: val_accuracy improved from 0.72038 to 0.72562, saving model to model1_weights.best.hdf5
250/250 [=====] - 15s 59ms/step - loss: 0.8764 - accuracy: 0.6782 - val_loss: 0.7500 - val_accuracy: 0.7256
Epoch 37/40
250/250 [=====] - ETA: 0s - loss: 0.8745 - accuracy: 0.6820
Epoch 37: val_accuracy did not improve from 0.72562
250/250 [=====] - 15s 60ms/step - loss: 0.8745 - accuracy: 0.6820 - val_loss: 0.7901 - val_accuracy: 0.7101
Epoch 38/40
249/250 [=====>.] - ETA: 0s - loss: 0.8577 - accuracy: 0.6849
Epoch 38: val_accuracy did not improve from 0.72562
250/250 [=====] - 14s 56ms/step - loss: 0.8575 - accuracy: 0.6850 - val_loss: 0.7536 - val_accuracy: 0.7171
Epoch 39/40
250/250 [=====] - ETA: 0s - loss: 0.8620 - accuracy: 0.6836
Epoch 39: val_accuracy did not improve from 0.72562
250/250 [=====] - 14s 55ms/step - loss: 0.8620 - accuracy: 0.6836 - val_loss: 0.7617 - val_accuracy: 0.7228
Epoch 40/40
250/250 [=====] - ETA: 0s - loss: 0.8557 - accuracy: 0.6902

Epoch 40: val_accuracy did not improve from 0.72562
250/250 [=====] - 14s 55ms/step - loss: 0.8557 -
accuracy: 0.6902 - val_loss: 0.7592 - val_accuracy: 0.7225
Validation accuracy: 0.6187000274658203 Validation loss: 1.161220908164978
Training on fold 3
Epoch 1/40
250/250 [=====] - ETA: 0s - loss: 1.8225 - accuracy:
0.3253
Epoch 1: val_accuracy did not improve from 0.72562
250/250 [=====] - 16s 58ms/step - loss: 1.8225 -
accuracy: 0.3253 - val_loss: 1.4670 - val_accuracy: 0.4680
Epoch 2/40
250/250 [=====] - ETA: 0s - loss: 1.4648 - accuracy:
0.4593
Epoch 2: val_accuracy did not improve from 0.72562
250/250 [=====] - 14s 57ms/step - loss: 1.4648 -
accuracy: 0.4593 - val_loss: 1.3136 - val_accuracy: 0.5278
Epoch 3/40
250/250 [=====] - ETA: 0s - loss: 1.3607 - accuracy:
0.5002
Epoch 3: val_accuracy did not improve from 0.72562
250/250 [=====] - 15s 62ms/step - loss: 1.3607 -
accuracy: 0.5002 - val_loss: 1.2159 - val_accuracy: 0.5494
Epoch 4/40
250/250 [=====] - ETA: 0s - loss: 1.3005 - accuracy:
0.5230
Epoch 4: val_accuracy did not improve from 0.72562
250/250 [=====] - 14s 57ms/step - loss: 1.3005 -
accuracy: 0.5230 - val_loss: 1.1540 - val_accuracy: 0.5811
Epoch 5/40
250/250 [=====] - ETA: 0s - loss: 1.2497 - accuracy:
0.5448
Epoch 5: val_accuracy did not improve from 0.72562
250/250 [=====] - 12s 50ms/step - loss: 1.2497 -
accuracy: 0.5448 - val_loss: 1.1801 - val_accuracy: 0.5715
Epoch 6/40
249/250 [=====>.] - ETA: 0s - loss: 1.2094 - accuracy:
0.5586
Epoch 6: val_accuracy did not improve from 0.72562
250/250 [=====] - 12s 48ms/step - loss: 1.2089 -
accuracy: 0.5587 - val_loss: 1.0526 - val_accuracy: 0.6186
Epoch 7/40
250/250 [=====] - ETA: 0s - loss: 1.1724 - accuracy:
0.5706
Epoch 7: val_accuracy did not improve from 0.72562
250/250 [=====] - 12s 49ms/step - loss: 1.1724 -
accuracy: 0.5706 - val_loss: 1.0406 - val_accuracy: 0.6315
Epoch 8/40

249/250 [=====>.] - ETA: 0s - loss: 1.1425 - accuracy: 0.5849

Epoch 8: val_accuracy did not improve from 0.72562

250/250 [=====] - 12s 49ms/step - loss: 1.1427 - accuracy: 0.5848 - val_loss: 1.0233 - val_accuracy: 0.6386

Epoch 9/40

250/250 [=====] - ETA: 0s - loss: 1.1181 - accuracy: 0.5939

Epoch 9: val_accuracy did not improve from 0.72562

250/250 [=====] - 13s 50ms/step - loss: 1.1181 - accuracy: 0.5939 - val_loss: 0.9895 - val_accuracy: 0.6436

Epoch 10/40

250/250 [=====] - ETA: 0s - loss: 1.0968 - accuracy: 0.5996

Epoch 10: val_accuracy did not improve from 0.72562

250/250 [=====] - 12s 49ms/step - loss: 1.0968 - accuracy: 0.5996 - val_loss: 0.9829 - val_accuracy: 0.6459

Epoch 11/40

249/250 [=====>.] - ETA: 0s - loss: 1.0636 - accuracy: 0.6114

Epoch 11: val_accuracy did not improve from 0.72562

250/250 [=====] - 12s 49ms/step - loss: 1.0639 - accuracy: 0.6113 - val_loss: 0.9685 - val_accuracy: 0.6531

Epoch 12/40

250/250 [=====] - ETA: 0s - loss: 1.0486 - accuracy: 0.6180

Epoch 12: val_accuracy did not improve from 0.72562

250/250 [=====] - 12s 49ms/step - loss: 1.0486 - accuracy: 0.6180 - val_loss: 0.9084 - val_accuracy: 0.6725

Epoch 13/40

250/250 [=====] - ETA: 0s - loss: 1.0381 - accuracy: 0.6224

Epoch 13: val_accuracy did not improve from 0.72562

250/250 [=====] - 13s 53ms/step - loss: 1.0381 - accuracy: 0.6224 - val_loss: 0.9199 - val_accuracy: 0.6731

Epoch 14/40

250/250 [=====] - ETA: 0s - loss: 1.0175 - accuracy: 0.6286

Epoch 14: val_accuracy did not improve from 0.72562

250/250 [=====] - 14s 55ms/step - loss: 1.0175 - accuracy: 0.6286 - val_loss: 0.8748 - val_accuracy: 0.6860

Epoch 15/40

250/250 [=====] - ETA: 0s - loss: 1.0027 - accuracy: 0.6352

Epoch 15: val_accuracy did not improve from 0.72562

250/250 [=====] - 16s 64ms/step - loss: 1.0027 - accuracy: 0.6352 - val_loss: 0.8791 - val_accuracy: 0.6830

Epoch 16/40

250/250 [=====] - ETA: 0s - loss: 0.9912 - accuracy: 0.6410
Epoch 16: val_accuracy did not improve from 0.72562
250/250 [=====] - 16s 63ms/step - loss: 0.9912 - accuracy: 0.6410 - val_loss: 0.8488 - val_accuracy: 0.6909
Epoch 17/40
250/250 [=====] - ETA: 0s - loss: 0.9817 - accuracy: 0.6428
Epoch 17: val_accuracy did not improve from 0.72562
250/250 [=====] - 14s 55ms/step - loss: 0.9817 - accuracy: 0.6428 - val_loss: 0.8526 - val_accuracy: 0.6900
Epoch 18/40
250/250 [=====] - ETA: 0s - loss: 0.9685 - accuracy: 0.6494
Epoch 18: val_accuracy did not improve from 0.72562
250/250 [=====] - 13s 53ms/step - loss: 0.9685 - accuracy: 0.6494 - val_loss: 0.8328 - val_accuracy: 0.6976
Epoch 19/40
250/250 [=====] - ETA: 0s - loss: 0.9637 - accuracy: 0.6472
Epoch 19: val_accuracy did not improve from 0.72562
250/250 [=====] - 15s 60ms/step - loss: 0.9637 - accuracy: 0.6472 - val_loss: 0.8243 - val_accuracy: 0.7011
Epoch 20/40
250/250 [=====] - ETA: 0s - loss: 0.9596 - accuracy: 0.6515
Epoch 20: val_accuracy did not improve from 0.72562
250/250 [=====] - 14s 58ms/step - loss: 0.9596 - accuracy: 0.6515 - val_loss: 0.8086 - val_accuracy: 0.7114
Epoch 21/40
250/250 [=====] - ETA: 0s - loss: 0.9441 - accuracy: 0.6581
Epoch 21: val_accuracy did not improve from 0.72562
250/250 [=====] - 14s 56ms/step - loss: 0.9441 - accuracy: 0.6581 - val_loss: 0.9067 - val_accuracy: 0.6739
Epoch 22/40
250/250 [=====] - ETA: 0s - loss: 0.9406 - accuracy: 0.6562
Epoch 22: val_accuracy did not improve from 0.72562
250/250 [=====] - 14s 57ms/step - loss: 0.9406 - accuracy: 0.6562 - val_loss: 0.8073 - val_accuracy: 0.7075
Epoch 23/40
250/250 [=====] - ETA: 0s - loss: 0.9258 - accuracy: 0.6615
Epoch 23: val_accuracy did not improve from 0.72562
250/250 [=====] - 15s 59ms/step - loss: 0.9258 - accuracy: 0.6615 - val_loss: 0.8504 - val_accuracy: 0.6935
Epoch 24/40

250/250 [=====] - ETA: 0s - loss: 0.9255 - accuracy: 0.6647
Epoch 24: val_accuracy did not improve from 0.72562
250/250 [=====] - 15s 60ms/step - loss: 0.9255 - accuracy: 0.6647 - val_loss: 0.8119 - val_accuracy: 0.7013
Epoch 25/40
250/250 [=====] - ETA: 0s - loss: 0.9236 - accuracy: 0.6632
Epoch 25: val_accuracy did not improve from 0.72562
250/250 [=====] - 15s 61ms/step - loss: 0.9236 - accuracy: 0.6632 - val_loss: 0.7992 - val_accuracy: 0.7076
Epoch 26/40
250/250 [=====] - ETA: 0s - loss: 0.9104 - accuracy: 0.6691
Epoch 26: val_accuracy did not improve from 0.72562
250/250 [=====] - 15s 59ms/step - loss: 0.9104 - accuracy: 0.6691 - val_loss: 0.7960 - val_accuracy: 0.7117
Epoch 27/40
250/250 [=====] - ETA: 0s - loss: 0.9060 - accuracy: 0.6721
Epoch 27: val_accuracy did not improve from 0.72562
250/250 [=====] - 15s 60ms/step - loss: 0.9060 - accuracy: 0.6721 - val_loss: 0.8416 - val_accuracy: 0.6916
Epoch 28/40
250/250 [=====] - ETA: 0s - loss: 0.9061 - accuracy: 0.6682
Epoch 28: val_accuracy did not improve from 0.72562
250/250 [=====] - 15s 60ms/step - loss: 0.9061 - accuracy: 0.6682 - val_loss: 0.7828 - val_accuracy: 0.7160
Epoch 29/40
250/250 [=====] - ETA: 0s - loss: 0.8944 - accuracy: 0.6750
Epoch 29: val_accuracy did not improve from 0.72562
250/250 [=====] - 14s 56ms/step - loss: 0.8944 - accuracy: 0.6750 - val_loss: 0.8447 - val_accuracy: 0.6974
Epoch 30/40
249/250 [=====>.] - ETA: 0s - loss: 0.8906 - accuracy: 0.6765
Epoch 30: val_accuracy did not improve from 0.72562
250/250 [=====] - 13s 53ms/step - loss: 0.8912 - accuracy: 0.6764 - val_loss: 0.7656 - val_accuracy: 0.7200
Epoch 31/40
250/250 [=====] - ETA: 0s - loss: 0.8854 - accuracy: 0.6756
Epoch 31: val_accuracy did not improve from 0.72562
250/250 [=====] - 13s 54ms/step - loss: 0.8854 - accuracy: 0.6756 - val_loss: 0.8152 - val_accuracy: 0.7028
Epoch 32/40

250/250 [=====] - ETA: 0s - loss: 0.8777 - accuracy: 0.6801
Epoch 32: val_accuracy did not improve from 0.72562
250/250 [=====] - 15s 59ms/step - loss: 0.8777 - accuracy: 0.6801 - val_loss: 0.7653 - val_accuracy: 0.7190
Epoch 33/40
250/250 [=====] - ETA: 0s - loss: 0.8783 - accuracy: 0.6813
Epoch 33: val_accuracy did not improve from 0.72562
250/250 [=====] - 15s 59ms/step - loss: 0.8783 - accuracy: 0.6813 - val_loss: 0.8026 - val_accuracy: 0.7059
Epoch 34/40
250/250 [=====] - ETA: 0s - loss: 0.8730 - accuracy: 0.6819
Epoch 34: val_accuracy did not improve from 0.72562
250/250 [=====] - 15s 59ms/step - loss: 0.8730 - accuracy: 0.6819 - val_loss: 0.7649 - val_accuracy: 0.7191
Epoch 35/40
250/250 [=====] - ETA: 0s - loss: 0.8746 - accuracy: 0.6819
Epoch 35: val_accuracy did not improve from 0.72562
250/250 [=====] - 15s 59ms/step - loss: 0.8746 - accuracy: 0.6819 - val_loss: 0.7611 - val_accuracy: 0.7204
Epoch 36/40
250/250 [=====] - ETA: 0s - loss: 0.8669 - accuracy: 0.6848
Epoch 36: val_accuracy improved from 0.72562 to 0.72763, saving model to model1_weights.best.hdf5
250/250 [=====] - 15s 59ms/step - loss: 0.8669 - accuracy: 0.6848 - val_loss: 0.7585 - val_accuracy: 0.7276
Epoch 37/40
250/250 [=====] - ETA: 0s - loss: 0.8638 - accuracy: 0.6865
Epoch 37: val_accuracy did not improve from 0.72763
250/250 [=====] - 14s 57ms/step - loss: 0.8638 - accuracy: 0.6865 - val_loss: 0.8168 - val_accuracy: 0.6992
Epoch 38/40
250/250 [=====] - ETA: 0s - loss: 0.8738 - accuracy: 0.6825
Epoch 38: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 59ms/step - loss: 0.8738 - accuracy: 0.6825 - val_loss: 0.7652 - val_accuracy: 0.7191
Epoch 39/40
250/250 [=====] - ETA: 0s - loss: 0.8614 - accuracy: 0.6867
Epoch 39: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 60ms/step - loss: 0.8614 - accuracy: 0.6867 - val_loss: 0.8206 - val_accuracy: 0.7006

Epoch 40/40
250/250 [=====] - ETA: 0s - loss: 0.8566 - accuracy: 0.6882
Epoch 40: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 60ms/step - loss: 0.8566 - accuracy: 0.6882 - val_loss: 0.7419 - val_accuracy: 0.7239
Validation accuracy: 0.6256999969482422 Validation loss: 1.1300629377365112
Training on fold 4
Epoch 1/40
250/250 [=====] - ETA: 0s - loss: 1.8295 - accuracy: 0.3186
Epoch 1: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 59ms/step - loss: 1.8295 - accuracy: 0.3186 - val_loss: 1.4697 - val_accuracy: 0.4665
Epoch 2/40
250/250 [=====] - ETA: 0s - loss: 1.4533 - accuracy: 0.4675
Epoch 2: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 59ms/step - loss: 1.4533 - accuracy: 0.4675 - val_loss: 1.2976 - val_accuracy: 0.5380
Epoch 3/40
250/250 [=====] - ETA: 0s - loss: 1.3445 - accuracy: 0.5092
Epoch 3: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 60ms/step - loss: 1.3445 - accuracy: 0.5092 - val_loss: 1.1827 - val_accuracy: 0.5735
Epoch 4/40
250/250 [=====] - ETA: 0s - loss: 1.2681 - accuracy: 0.5407
Epoch 4: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 65ms/step - loss: 1.2681 - accuracy: 0.5407 - val_loss: 1.1761 - val_accuracy: 0.5776
Epoch 5/40
250/250 [=====] - ETA: 0s - loss: 1.2164 - accuracy: 0.5616
Epoch 5: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 59ms/step - loss: 1.2164 - accuracy: 0.5616 - val_loss: 1.0925 - val_accuracy: 0.5990
Epoch 6/40
250/250 [=====] - ETA: 0s - loss: 1.1687 - accuracy: 0.5763
Epoch 6: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 59ms/step - loss: 1.1687 - accuracy: 0.5763 - val_loss: 1.0526 - val_accuracy: 0.6227
Epoch 7/40
250/250 [=====] - ETA: 0s - loss: 1.1376 - accuracy: 0.5909
Epoch 7: val_accuracy did not improve from 0.72763

250/250 [=====] - 15s 60ms/step - loss: 1.1376 - accuracy: 0.5909 - val_loss: 1.1061 - val_accuracy: 0.5994
Epoch 8/40
250/250 [=====] - ETA: 0s - loss: 1.1053 - accuracy: 0.5976
Epoch 8: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 62ms/step - loss: 1.1053 - accuracy: 0.5976 - val_loss: 0.9948 - val_accuracy: 0.6320
Epoch 9/40
250/250 [=====] - ETA: 0s - loss: 1.0911 - accuracy: 0.6059
Epoch 9: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 64ms/step - loss: 1.0911 - accuracy: 0.6059 - val_loss: 0.9981 - val_accuracy: 0.6436
Epoch 10/40
250/250 [=====] - ETA: 0s - loss: 1.0711 - accuracy: 0.6125
Epoch 10: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 64ms/step - loss: 1.0711 - accuracy: 0.6125 - val_loss: 0.9679 - val_accuracy: 0.6467
Epoch 11/40
250/250 [=====] - ETA: 0s - loss: 1.0512 - accuracy: 0.6189
Epoch 11: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 65ms/step - loss: 1.0512 - accuracy: 0.6189 - val_loss: 0.8917 - val_accuracy: 0.6712
Epoch 12/40
250/250 [=====] - ETA: 0s - loss: 1.0298 - accuracy: 0.6278
Epoch 12: val_accuracy did not improve from 0.72763
250/250 [=====] - 14s 57ms/step - loss: 1.0298 - accuracy: 0.6278 - val_loss: 0.9080 - val_accuracy: 0.6600
Epoch 13/40
250/250 [=====] - ETA: 0s - loss: 1.0175 - accuracy: 0.6280
Epoch 13: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 60ms/step - loss: 1.0175 - accuracy: 0.6280 - val_loss: 0.8786 - val_accuracy: 0.6749
Epoch 14/40
250/250 [=====] - ETA: 0s - loss: 0.9972 - accuracy: 0.6397
Epoch 14: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 59ms/step - loss: 0.9972 - accuracy: 0.6397 - val_loss: 0.8594 - val_accuracy: 0.6860
Epoch 15/40
249/250 [=====>.] - ETA: 0s - loss: 0.9937 - accuracy: 0.6386
Epoch 15: val_accuracy did not improve from 0.72763

250/250 [=====] - 14s 55ms/step - loss: 0.9941 - accuracy: 0.6385 - val_loss: 0.9010 - val_accuracy: 0.6684
Epoch 16/40
250/250 [=====] - ETA: 0s - loss: 0.9805 - accuracy: 0.6445
Epoch 16: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 58ms/step - loss: 0.9805 - accuracy: 0.6445 - val_loss: 0.8338 - val_accuracy: 0.6927
Epoch 17/40
250/250 [=====] - ETA: 0s - loss: 0.9621 - accuracy: 0.6501
Epoch 17: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 61ms/step - loss: 0.9621 - accuracy: 0.6501 - val_loss: 0.8428 - val_accuracy: 0.6867
Epoch 18/40
250/250 [=====] - ETA: 0s - loss: 0.9559 - accuracy: 0.6528
Epoch 18: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 60ms/step - loss: 0.9559 - accuracy: 0.6528 - val_loss: 0.8649 - val_accuracy: 0.6764
Epoch 19/40
250/250 [=====] - ETA: 0s - loss: 0.9561 - accuracy: 0.6498
Epoch 19: val_accuracy did not improve from 0.72763
250/250 [=====] - 14s 56ms/step - loss: 0.9561 - accuracy: 0.6498 - val_loss: 0.8894 - val_accuracy: 0.6681
Epoch 20/40
250/250 [=====] - ETA: 0s - loss: 0.9508 - accuracy: 0.6572
Epoch 20: val_accuracy did not improve from 0.72763
250/250 [=====] - 13s 54ms/step - loss: 0.9508 - accuracy: 0.6572 - val_loss: 0.8305 - val_accuracy: 0.6895
Epoch 21/40
250/250 [=====] - ETA: 0s - loss: 0.9393 - accuracy: 0.6567
Epoch 21: val_accuracy did not improve from 0.72763
250/250 [=====] - 14s 57ms/step - loss: 0.9393 - accuracy: 0.6567 - val_loss: 0.8484 - val_accuracy: 0.6793
Epoch 22/40
250/250 [=====] - ETA: 0s - loss: 0.9286 - accuracy: 0.6629
Epoch 22: val_accuracy did not improve from 0.72763
250/250 [=====] - 14s 57ms/step - loss: 0.9286 - accuracy: 0.6629 - val_loss: 0.8655 - val_accuracy: 0.6880
Epoch 23/40
250/250 [=====] - ETA: 0s - loss: 0.9210 - accuracy: 0.6642
Epoch 23: val_accuracy did not improve from 0.72763

250/250 [=====] - 14s 58ms/step - loss: 0.9210 - accuracy: 0.6642 - val_loss: 0.8012 - val_accuracy: 0.6986
Epoch 24/40
250/250 [=====] - ETA: 0s - loss: 0.9174 - accuracy: 0.6663
Epoch 24: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 59ms/step - loss: 0.9174 - accuracy: 0.6663 - val_loss: 0.8746 - val_accuracy: 0.6760
Epoch 25/40
250/250 [=====] - ETA: 0s - loss: 0.9183 - accuracy: 0.6668
Epoch 25: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 59ms/step - loss: 0.9183 - accuracy: 0.6668 - val_loss: 0.8076 - val_accuracy: 0.6936
Epoch 26/40
250/250 [=====] - ETA: 0s - loss: 0.9111 - accuracy: 0.6693
Epoch 26: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 59ms/step - loss: 0.9111 - accuracy: 0.6693 - val_loss: 0.8174 - val_accuracy: 0.6894
Epoch 27/40
249/250 [=====>.] - ETA: 0s - loss: 0.8912 - accuracy: 0.6738
Epoch 27: val_accuracy did not improve from 0.72763
250/250 [=====] - 13s 53ms/step - loss: 0.8910 - accuracy: 0.6737 - val_loss: 0.7921 - val_accuracy: 0.7009
Epoch 28/40
250/250 [=====] - ETA: 0s - loss: 0.8926 - accuracy: 0.6749
Epoch 28: val_accuracy did not improve from 0.72763
250/250 [=====] - 13s 52ms/step - loss: 0.8926 - accuracy: 0.6749 - val_loss: 0.8381 - val_accuracy: 0.6839
Epoch 29/40
250/250 [=====] - ETA: 0s - loss: 0.8967 - accuracy: 0.6756
Epoch 29: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 60ms/step - loss: 0.8967 - accuracy: 0.6756 - val_loss: 0.9227 - val_accuracy: 0.6574
Epoch 30/40
250/250 [=====] - ETA: 0s - loss: 0.8891 - accuracy: 0.6782
Epoch 30: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 61ms/step - loss: 0.8891 - accuracy: 0.6782 - val_loss: 0.7967 - val_accuracy: 0.6986
Epoch 31/40
250/250 [=====] - ETA: 0s - loss: 0.8794 - accuracy: 0.6768
Epoch 31: val_accuracy did not improve from 0.72763

250/250 [=====] - 17s 67ms/step - loss: 0.8794 - accuracy: 0.6768 - val_loss: 0.8563 - val_accuracy: 0.6761
Epoch 32/40
250/250 [=====] - ETA: 0s - loss: 0.8842 - accuracy: 0.6770
Epoch 32: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 60ms/step - loss: 0.8842 - accuracy: 0.6770 - val_loss: 0.7722 - val_accuracy: 0.7124
Epoch 33/40
250/250 [=====] - ETA: 0s - loss: 0.8783 - accuracy: 0.6803
Epoch 33: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 62ms/step - loss: 0.8783 - accuracy: 0.6803 - val_loss: 0.7868 - val_accuracy: 0.7061
Epoch 34/40
250/250 [=====] - ETA: 0s - loss: 0.8829 - accuracy: 0.6797
Epoch 34: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 65ms/step - loss: 0.8829 - accuracy: 0.6797 - val_loss: 0.7861 - val_accuracy: 0.6990
Epoch 35/40
250/250 [=====] - ETA: 0s - loss: 0.8747 - accuracy: 0.6809
Epoch 35: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 62ms/step - loss: 0.8747 - accuracy: 0.6809 - val_loss: 0.8066 - val_accuracy: 0.6960
Epoch 36/40
250/250 [=====] - ETA: 0s - loss: 0.8628 - accuracy: 0.6845
Epoch 36: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 59ms/step - loss: 0.8628 - accuracy: 0.6845 - val_loss: 0.7905 - val_accuracy: 0.7029
Epoch 37/40
249/250 [=====>.] - ETA: 0s - loss: 0.8635 - accuracy: 0.6878
Epoch 37: val_accuracy did not improve from 0.72763
250/250 [=====] - 13s 53ms/step - loss: 0.8637 - accuracy: 0.6876 - val_loss: 0.7728 - val_accuracy: 0.7157
Epoch 38/40
249/250 [=====>.] - ETA: 0s - loss: 0.8599 - accuracy: 0.6867
Epoch 38: val_accuracy did not improve from 0.72763
250/250 [=====] - 12s 49ms/step - loss: 0.8599 - accuracy: 0.6867 - val_loss: 0.8037 - val_accuracy: 0.6933
Epoch 39/40
250/250 [=====] - ETA: 0s - loss: 0.8623 - accuracy: 0.6825
Epoch 39: val_accuracy did not improve from 0.72763

250/250 [=====] - 15s 61ms/step - loss: 0.8623 - accuracy: 0.6825 - val_loss: 0.8036 - val_accuracy: 0.6992
Epoch 40/40
250/250 [=====] - ETA: 0s - loss: 0.8529 - accuracy: 0.6877
Epoch 40: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 62ms/step - loss: 0.8529 - accuracy: 0.6877 - val_loss: 0.7836 - val_accuracy: 0.7035
Validation accuracy: 0.6328999996185303 Validation loss: 1.110082983970642
Training on fold 5
Epoch 1/40
250/250 [=====] - ETA: 0s - loss: 1.8478 - accuracy: 0.3166
Epoch 1: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 59ms/step - loss: 1.8478 - accuracy: 0.3166 - val_loss: 1.4879 - val_accuracy: 0.4608
Epoch 2/40
250/250 [=====] - ETA: 0s - loss: 1.4947 - accuracy: 0.4548
Epoch 2: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 63ms/step - loss: 1.4947 - accuracy: 0.4548 - val_loss: 1.3275 - val_accuracy: 0.5146
Epoch 3/40
250/250 [=====] - ETA: 0s - loss: 1.3906 - accuracy: 0.4918
Epoch 3: val_accuracy did not improve from 0.72763
250/250 [=====] - 17s 68ms/step - loss: 1.3906 - accuracy: 0.4918 - val_loss: 1.2593 - val_accuracy: 0.5446
Epoch 4/40
250/250 [=====] - ETA: 0s - loss: 1.3234 - accuracy: 0.5166
Epoch 4: val_accuracy did not improve from 0.72763
250/250 [=====] - 17s 68ms/step - loss: 1.3234 - accuracy: 0.5166 - val_loss: 1.2474 - val_accuracy: 0.5511
Epoch 5/40
250/250 [=====] - ETA: 0s - loss: 1.2715 - accuracy: 0.5365
Epoch 5: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 64ms/step - loss: 1.2715 - accuracy: 0.5365 - val_loss: 1.1341 - val_accuracy: 0.5907
Epoch 6/40
250/250 [=====] - ETA: 0s - loss: 1.2328 - accuracy: 0.5522
Epoch 6: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 64ms/step - loss: 1.2328 - accuracy: 0.5522 - val_loss: 1.0709 - val_accuracy: 0.6062
Epoch 7/40
250/250 [=====] - ETA: 0s - loss: 1.1991 - accuracy:

0.5659
Epoch 7: val_accuracy did not improve from 0.72763
250/250 [=====] - 14s 57ms/step - loss: 1.1991 -
accuracy: 0.5659 - val_loss: 1.1091 - val_accuracy: 0.6000
Epoch 8/40
250/250 [=====] - ETA: 0s - loss: 1.1609 - accuracy:
0.5800
Epoch 8: val_accuracy did not improve from 0.72763
250/250 [=====] - 13s 51ms/step - loss: 1.1609 -
accuracy: 0.5800 - val_loss: 1.0191 - val_accuracy: 0.6274
Epoch 9/40
249/250 [=====>.] - ETA: 0s - loss: 1.1454 - accuracy:
0.5847
Epoch 9: val_accuracy did not improve from 0.72763
250/250 [=====] - 13s 51ms/step - loss: 1.1454 -
accuracy: 0.5847 - val_loss: 0.9859 - val_accuracy: 0.6434
Epoch 10/40
249/250 [=====>.] - ETA: 0s - loss: 1.1166 - accuracy:
0.5960
Epoch 10: val_accuracy did not improve from 0.72763
250/250 [=====] - 13s 51ms/step - loss: 1.1167 -
accuracy: 0.5961 - val_loss: 0.9771 - val_accuracy: 0.6436
Epoch 11/40
250/250 [=====] - ETA: 0s - loss: 1.0926 - accuracy:
0.6032
Epoch 11: val_accuracy did not improve from 0.72763
250/250 [=====] - 13s 52ms/step - loss: 1.0926 -
accuracy: 0.6032 - val_loss: 0.9779 - val_accuracy: 0.6476
Epoch 12/40
250/250 [=====] - ETA: 0s - loss: 1.0720 - accuracy:
0.6124
Epoch 12: val_accuracy did not improve from 0.72763
250/250 [=====] - 13s 50ms/step - loss: 1.0720 -
accuracy: 0.6124 - val_loss: 0.9713 - val_accuracy: 0.6497
Epoch 13/40
250/250 [=====] - ETA: 0s - loss: 1.0591 - accuracy:
0.6145
Epoch 13: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 58ms/step - loss: 1.0591 -
accuracy: 0.6145 - val_loss: 0.9013 - val_accuracy: 0.6774
Epoch 14/40
250/250 [=====] - ETA: 0s - loss: 1.0471 - accuracy:
0.6209
Epoch 14: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 61ms/step - loss: 1.0471 -
accuracy: 0.6209 - val_loss: 0.9640 - val_accuracy: 0.6518
Epoch 15/40
250/250 [=====] - ETA: 0s - loss: 1.0256 - accuracy:

0.6288
Epoch 15: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 62ms/step - loss: 1.0256 -
accuracy: 0.6288 - val_loss: 0.9176 - val_accuracy: 0.6625
Epoch 16/40
250/250 [=====] - ETA: 0s - loss: 1.0156 - accuracy:
0.6329
Epoch 16: val_accuracy did not improve from 0.72763
250/250 [=====] - 17s 66ms/step - loss: 1.0156 -
accuracy: 0.6329 - val_loss: 0.8833 - val_accuracy: 0.6697
Epoch 17/40
250/250 [=====] - ETA: 0s - loss: 1.0039 - accuracy:
0.6358
Epoch 17: val_accuracy did not improve from 0.72763
250/250 [=====] - 17s 66ms/step - loss: 1.0039 -
accuracy: 0.6358 - val_loss: 0.8690 - val_accuracy: 0.6860
Epoch 18/40
250/250 [=====] - ETA: 0s - loss: 0.9937 - accuracy:
0.6439
Epoch 18: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 61ms/step - loss: 0.9937 -
accuracy: 0.6439 - val_loss: 0.8671 - val_accuracy: 0.6833
Epoch 19/40
250/250 [=====] - ETA: 0s - loss: 0.9881 - accuracy:
0.6443
Epoch 19: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 60ms/step - loss: 0.9881 -
accuracy: 0.6443 - val_loss: 0.8636 - val_accuracy: 0.6795
Epoch 20/40
250/250 [=====] - ETA: 0s - loss: 0.9744 - accuracy:
0.6461
Epoch 20: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 59ms/step - loss: 0.9744 -
accuracy: 0.6461 - val_loss: 0.8412 - val_accuracy: 0.6923
Epoch 21/40
250/250 [=====] - ETA: 0s - loss: 0.9628 - accuracy:
0.6511
Epoch 21: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 58ms/step - loss: 0.9628 -
accuracy: 0.6511 - val_loss: 0.8167 - val_accuracy: 0.6988
Epoch 22/40
250/250 [=====] - ETA: 0s - loss: 0.9611 - accuracy:
0.6515
Epoch 22: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 63ms/step - loss: 0.9611 -
accuracy: 0.6515 - val_loss: 0.8436 - val_accuracy: 0.6920
Epoch 23/40
250/250 [=====] - ETA: 0s - loss: 0.9498 - accuracy:

0.6537
Epoch 23: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 62ms/step - loss: 0.9498 -
accuracy: 0.6537 - val_loss: 0.8360 - val_accuracy: 0.6979
Epoch 24/40
250/250 [=====] - ETA: 0s - loss: 0.9485 - accuracy:
0.6560
Epoch 24: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 62ms/step - loss: 0.9485 -
accuracy: 0.6560 - val_loss: 0.8388 - val_accuracy: 0.6949
Epoch 25/40
250/250 [=====] - ETA: 0s - loss: 0.9399 - accuracy:
0.6609
Epoch 25: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 63ms/step - loss: 0.9399 -
accuracy: 0.6609 - val_loss: 0.8138 - val_accuracy: 0.6988
Epoch 26/40
250/250 [=====] - ETA: 0s - loss: 0.9334 - accuracy:
0.6654
Epoch 26: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 63ms/step - loss: 0.9334 -
accuracy: 0.6654 - val_loss: 0.8295 - val_accuracy: 0.6917
Epoch 27/40
250/250 [=====] - ETA: 0s - loss: 0.9285 - accuracy:
0.6605
Epoch 27: val_accuracy did not improve from 0.72763
250/250 [=====] - 14s 58ms/step - loss: 0.9285 -
accuracy: 0.6605 - val_loss: 0.8115 - val_accuracy: 0.7042
Epoch 28/40
250/250 [=====] - ETA: 0s - loss: 0.9297 - accuracy:
0.6625
Epoch 28: val_accuracy did not improve from 0.72763
250/250 [=====] - 14s 57ms/step - loss: 0.9297 -
accuracy: 0.6625 - val_loss: 0.7952 - val_accuracy: 0.7064
Epoch 29/40
250/250 [=====] - ETA: 0s - loss: 0.9203 - accuracy:
0.6664
Epoch 29: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 64ms/step - loss: 0.9203 -
accuracy: 0.6664 - val_loss: 0.8010 - val_accuracy: 0.7095
Epoch 30/40
250/250 [=====] - ETA: 0s - loss: 0.9162 - accuracy:
0.6663
Epoch 30: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 64ms/step - loss: 0.9162 -
accuracy: 0.6663 - val_loss: 0.7957 - val_accuracy: 0.7056
Epoch 31/40
250/250 [=====] - ETA: 0s - loss: 0.8952 - accuracy:

0.6778
Epoch 31: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 63ms/step - loss: 0.8952 -
accuracy: 0.6778 - val_loss: 0.7847 - val_accuracy: 0.7126
Epoch 32/40
250/250 [=====] - ETA: 0s - loss: 0.9001 - accuracy:
0.6750
Epoch 32: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 64ms/step - loss: 0.9001 -
accuracy: 0.6750 - val_loss: 0.8218 - val_accuracy: 0.6979
Epoch 33/40
250/250 [=====] - ETA: 0s - loss: 0.8998 - accuracy:
0.6730
Epoch 33: val_accuracy did not improve from 0.72763
250/250 [=====] - 15s 60ms/step - loss: 0.8998 -
accuracy: 0.6730 - val_loss: 0.8403 - val_accuracy: 0.6906
Epoch 34/40
249/250 [=====>.] - ETA: 0s - loss: 0.8922 - accuracy:
0.6750
Epoch 34: val_accuracy did not improve from 0.72763
250/250 [=====] - 14s 54ms/step - loss: 0.8919 -
accuracy: 0.6750 - val_loss: 0.8188 - val_accuracy: 0.7044
Epoch 35/40
250/250 [=====] - ETA: 0s - loss: 0.8893 - accuracy:
0.6782
Epoch 35: val_accuracy did not improve from 0.72763
250/250 [=====] - 13s 52ms/step - loss: 0.8893 -
accuracy: 0.6782 - val_loss: 0.8027 - val_accuracy: 0.7013
Epoch 36/40
250/250 [=====] - ETA: 0s - loss: 0.8888 - accuracy:
0.6766
Epoch 36: val_accuracy did not improve from 0.72763
250/250 [=====] - 13s 51ms/step - loss: 0.8888 -
accuracy: 0.6766 - val_loss: 0.7737 - val_accuracy: 0.7184
Epoch 37/40
250/250 [=====] - ETA: 0s - loss: 0.8904 - accuracy:
0.6737
Epoch 37: val_accuracy did not improve from 0.72763
250/250 [=====] - 14s 57ms/step - loss: 0.8904 -
accuracy: 0.6737 - val_loss: 0.7698 - val_accuracy: 0.7160
Epoch 38/40
250/250 [=====] - ETA: 0s - loss: 0.8766 - accuracy:
0.6826
Epoch 38: val_accuracy did not improve from 0.72763
250/250 [=====] - 17s 66ms/step - loss: 0.8766 -
accuracy: 0.6826 - val_loss: 0.7795 - val_accuracy: 0.7121
Epoch 39/40
250/250 [=====] - ETA: 0s - loss: 0.8794 - accuracy:

```

0.6817
Epoch 39: val_accuracy did not improve from 0.72763
250/250 [=====] - 17s 67ms/step - loss: 0.8794 -
accuracy: 0.6817 - val_loss: 0.7874 - val_accuracy: 0.7085
Epoch 40/40
250/250 [=====] - ETA: 0s - loss: 0.8756 - accuracy:
0.6795
Epoch 40: val_accuracy did not improve from 0.72763
250/250 [=====] - 16s 65ms/step - loss: 0.8756 -
accuracy: 0.6795 - val_loss: 0.7624 - val_accuracy: 0.7153
Validation accuracy: 0.6136999726295471 Validation loss: 1.1567187309265137
Average validation accuracy: 0.621399998664856
Average validation loss: 1.1472977161407472

```

By applying cross validation to model 2, we can find that the model 2 has similar performance on each train and validation data set.

```

[19]: # Plot the training and validation loss for each fold
fig = plt.figure(figsize=(20,6))
hist = pd.DataFrame(h[0].history)
ax1 = fig.add_subplot(1,5,1)
ax1.plot(hist.index, hist['loss'], label='training loss')
ax1.plot(hist.index, hist['val_loss'], label='val loss')
ax1.legend()

hist = pd.DataFrame(h[1].history)
ax1 = fig.add_subplot(1,5,2)
ax1.plot(hist.index, hist['loss'], label='training loss')
ax1.plot(hist.index, hist['val_loss'], label='val loss')
ax1.legend()

hist = pd.DataFrame(h[2].history)
ax1 = fig.add_subplot(1,5,3)
ax1.plot(hist.index, hist['loss'], label='training loss')
ax1.plot(hist.index, hist['val_loss'], label='val loss')
ax1.legend()

hist = pd.DataFrame(h[3].history)
ax1 = fig.add_subplot(1,5,4)
ax1.plot(hist.index, hist['loss'], label='training loss')
ax1.plot(hist.index, hist['val_loss'], label='val loss')
ax1.legend()

hist = pd.DataFrame(h[4].history)

```

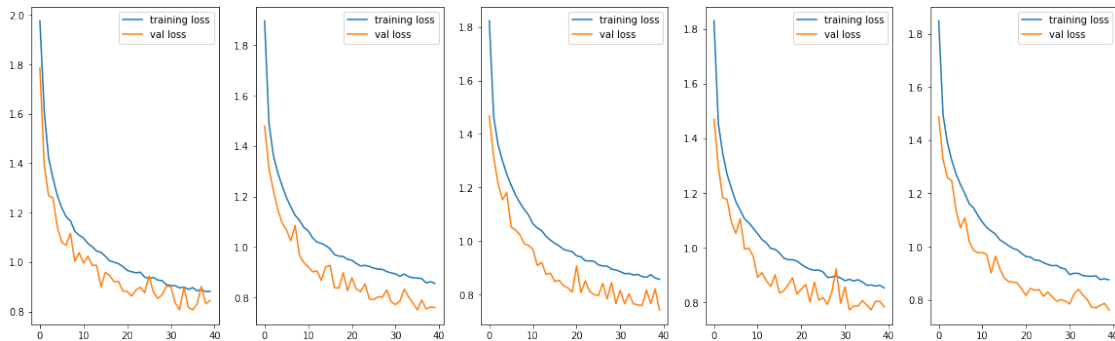


```

ax1 = fig.add_subplot(1,5,5)
ax1.plot(hist.index, hist['loss'], label='training loss')
ax1.plot(hist.index, hist['val_loss'], label='val loss')
ax1.legend()

```

[19]: <matplotlib.legend.Legend at 0x7fd9344c2d00>



```

[20]: # Plot the training and validation accuracy for each fold
fig = plt.figure(figsize=(20,6))
hist = pd.DataFrame(h[0].history)
ax1 = fig.add_subplot(1,5,1)
ax1.plot(hist.index, hist['accuracy'], label='training accuracy')
ax1.plot(hist.index, hist['val_accuracy'], label='val accuracy')
ax1.legend()

hist = pd.DataFrame(h[1].history)
ax1 = fig.add_subplot(1,5,2)
ax1.plot(hist.index, hist['accuracy'], label='training accuracy')
ax1.plot(hist.index, hist['val_accuracy'], label='val accuracy')
ax1.legend()

hist = pd.DataFrame(h[2].history)
ax1 = fig.add_subplot(1,5,3)
ax1.plot(hist.index, hist['accuracy'], label='training accuracy')
ax1.plot(hist.index, hist['val_accuracy'], label='val accuracy')
ax1.legend()

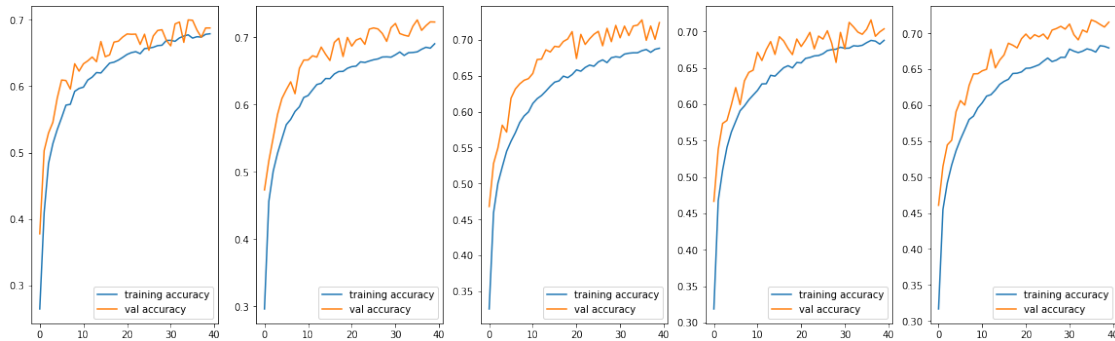
hist = pd.DataFrame(h[3].history)
ax1 = fig.add_subplot(1,5,4)
ax1.plot(hist.index, hist['accuracy'], label='training accuracy')
ax1.plot(hist.index, hist['val_accuracy'], label='val accuracy')

```

```
ax1.legend()

hist = pd.DataFrame(h[4].history)
ax1 = fig.add_subplot(1,5,5)
ax1.plot(hist.index, hist['accuracy'], label = 'training accuracy')
ax1.plot(hist.index, hist['val_accuracy'], label = 'val accuracy')
ax1.legend()
```

[20]: <matplotlib.legend.Legend at 0x7fd9338c1a30>



[24]: *# Load the best weights from the model 2 training*

```
model2 = Sequential()
model2.add(Conv2D(filters=6, kernel_size=(5,5), padding='same',
    ↪activation='relu', input_shape=(32, 32, 3)))
model2.add(MaxPool2D(pool_size=(2,2)))
model2.add(Dropout(0.3))
model2.add(Conv2D(filters=16, kernel_size=(5,5), padding = 'valid' ,
    ↪activation='relu'))
model2.add(MaxPool2D(pool_size=(2,2)))
model2.add(Dropout(0.3))
model2.add(Conv2D(filters=120, kernel_size=(5,5), padding = 'valid' ,
    ↪activation='relu'))
model2.add(Flatten())
model2.add(Dropout(0.3))
model2.add(Dense(84, activation='relu'))
model2.add(Dropout(0.3))
model2.add(Dense(10, activation='softmax'))

model2.load_weights(filepath)

model2.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy',
    ↪metrics = ['accuracy'])
```

```
# Evaluate the model on the test data
test_loss, test_acc = model2.evaluate(x_clean, y_clean, verbose=0)
print('Test accuracy:', test_acc, 'Test loss:', test_loss)
```

Test accuracy: 0.6097000241279602 Test loss: 1.250930666923523

```
[28]: def model_II(image):
        """
        This function should takes in the image of dimension 32*32*3 as input and
        ↪returns a label prediction
        """
        # write your code here...

        img=image.reshape(1,32,32,3)/255.0
        prediction=model2.predict(img)
        return np.argmax(prediction)
```

0.3 3. Evaluation

For assessment, we will evaluate your final model on a hidden test dataset with clean labels by the `evaluation` function defined as follows. Although you will not have the access to the test set, the function would be useful for the model developments. For example, you can split the small training set, using one portion for weakly supervised learning and the other for validation purpose.

```
[26]: # [DO NOT MODIFY THIS CELL]
def evaluation(model, test_labels, test_imgs):
    y_true = test_labels
    y_pred = []
    for image in test_imgs:
        y_pred.append(model(image))
    print(classification_report(y_true, y_pred))
```

```
[ ]: # [DO NOT MODIFY THIS CELL]
# This is the code for evaluating the prediction performance on a testset
# You will get an error if running this cell, as you do not have the testset
# Nonetheless, you can create your own validation set to run the evaluation
n_test = 10000
test_labels = np.genfromtxt('../data/test_labels.csv', delimiter=',',
    ↪dtype="int8")
test_imgs = np.empty((n_test,32,32,3))
for i in range(n_test):
    img_fn = f'../data/test_images/test{i+1:05d}.png'
    test_imgs[i,:,:,:]=cv2.cvtColor(cv2.imread(img_fn),cv2.COLOR_BGR2RGB)
evaluation(baseline_model, test_labels, test_imgs)
```

The overall accuracy is 0.24, which is better than random guess (which should have a accuracy around 0.10). For the project, you should try to improve the performance by the following strategies:

- Consider a better choice of model architectures, hyperparameters, or training scheme for the predictive model;
- Use both `clean_noisy_trainset` and `noisy_trainset` for model training via **weakly supervised learning** methods. One possible solution is to train a “label-correction” model using the former, correct the labels in the latter, and train the final predictive model using the corrected dataset.
- Apply techniques such as k -fold cross validation to avoid overfitting;
- Any other reasonable strategies.

[]: