Proposing the structure of a convolutional neural network, to identify the type of images of documents, the proposed structure of the convolutional neural network includes dividing the image of the document into four main regions and passing those regions to the convolutional neural network as a sequence of single images, and the goal of this methodology is the ability to facilitate the work of The convolutional neural network is able to extract the basic characteristics included in each of the four regions, and then use the GlobalAveragePooling1D layer in order to reach the general characteristics that distinguish the document, and thus the ability of the neural network to easily find the general characteristics that characterize each type of document.

We know that the type of document is determined according to many specifications, such as the design of the document, the header and footer, the body of the document and how the writing is formatted within the document, all of these factors help in the process of identifying the type of document.

Thus, we divided into the head of the document, the bottom of the document, the body of the document and it was divided into two regions (the right body and the left body). Using the TimeDistributed layer, we are able to pass the images to the neural network as a set of four sub-images, where the properties are extracted from each part and then the common general properties are extracted.

the References: Adam W. Harley, A. U. (2015). Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval. Toronto, Ontario: Ryerson University. [link study]

The study, which was my main reference, includes the same process of dividing the document into four sections, but with a difference in how to collect the common features. PCA & Conca was used in the study, while it was used in the GlobalAveragePooling1D code.

Another difference is the study used multiple convolutional neural structures for each part extracted from the images of the document, while in my notebook, one convolutional neural network was used, and the network input was considered to be 4 parts representing one complete image.

In [1]:

```
!pip install opendatasets
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/publi
c/simple/
Collecting opendatasets
  Downloading opendatasets-0.1.22-py3-none-any.whl (15 kB)
Requirement already satisfied: kaggle in /usr/local/lib/python3.7/dist-packages (from ope
ndatasets) (1.5.12)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages (from open
datasets) (7.1.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from opend
atasets) (4.64.0)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.7/dist-packages
(from kaggle->opendatasets) (2.8.2)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from k
aggle->opendatasets) (2.23.0)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.7/dist-packages (
from kaggle->opendatasets) (6.1.2)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from ka
ggle->opendatasets) (1.24.3)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.7/dist-packages (from
kaggle->opendatasets) (1.15.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from ka
ggle->opendatasets) (2022.6.15)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.7/dist-packa
ges (from python-slugify->kaggle->opendatasets) (1.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (fr
om requests->kaggle->opendatasets) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-package
s (from requests->kaggle->opendatasets) (3.0.4)
Installing collected packages: opendatasets
Successfully installed opendatasets-0.1.22
```

In [2]:

```
import opendatasets as op
op.download("https://www.kaggle.com/datasets/pdavpoojan/the-rvlcdip-dataset-test")
```

```
Please provide your Kaggle credentials to download this dataset. Learn more: http://bit.l
y/kaggle-creds
Your Kaggle username: kaledhoshme
Your Kaggle Key: ··········
Downloading the-rvlcdip-dataset-test.zip to ./the-rvlcdip-dataset-test
```

```
100%|██████████| 3.62G/3.62G [01:29<00:00, 43.4MB/s]
```

In [3]:

```
import pandas as pd
import numpy as np
import tensorflow as tf
import string
import nltk
import pathlib
import os
import cv2
import matplotlib.pyplot as plt
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.metrics import TruePositives, FalsePositives, TrueNegatives, FalseN
egatives, BinaryAccuracy, Precision, Recall, AUC
```

In [4]:

```
import shutil
shutil.rmtree("the-rvlcdip-dataset-test/test/scientific_publication")
```

In [5]:

```
datasetFolder = "the-rvlcdip-dataset-test/test/"
```

In [6]:

```
train = pathlib.Path(os.path.join(datasetFolder))
```

In [42]:

```
def get_images_labels(images, label):
  arr = []
  labels = []
  for i in images:
    img = cv2.imread(os.path.join(i))
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    img = cv2.resize(img, (120, 120))
    img1 = img[0:30, 0:120]/255
    img2 = img[30:90, 0:60]/255
    img3 = img[30:90, 60:120]/255
    img4 = img[90:120, 0:120]/255
    img = np.asarray([cv2.resize(img1, (48, 48)),
                      cv2.resize(img2, (48, 48)),
                      cv2.resize(img3, (48, 48)),
                      cv2.resize(img4, (48, 48))])
    img_mean = np.mean(img)
    img = img - img_mean
    img = img / np.std(img)
    arr.append(img)
    labels.append(label)
  return [arr, labels]
```

In [43]:

```
[advertisement, Y_advertisement] = get_images_labels(list(train.glob("advertisement/*.*")
), 0)
[budget, Y_budget] = get_images_labels(list(train.glob("budget/*.*")), 1)
[email, Y_email] = get_images_labels(list(train.glob("email/*.*")), 2)
[file_folder, Y_file_folder] = get_images_labels(list(train.glob("file_folder/*.*")), 3)
```

```
[form, Y_form] = get_images_labels(list(train.glob("form/*.*")), 4)
[handwritten, Y_handwritten] = get_images_labels(list(train.glob("handwritten/*.*")), 5)
[invoice, Y_invoice] = get_images_labels(list(train.glob("invoice/*.*")), 6)
[letter, Y_letter] = get_images_labels(list(train.glob("letter/*.*")), 7)
[memo, Y_memo] = get_images_labels(list(train.glob("memo/*.*")), 8)
[news_article, Y_news_article] = get_images_labels(list(train.glob("news_article/*.*")),
9)
[presentation, Y_presentation] = get_images_labels(list(train.glob("presentation/*.*")),
10)
[questionnaire, Y_questionnaire] = get_images_labels(list(train.glob("questionnaire/*.*")
), 11)
[resume, Y_resume] = get_images_labels(list(train.glob("resume/*.*")), 12)
[scientific_report, Y_scientific_report] = get_images_labels(list(train.glob("scientific
_report/*.*")), 13)
[specification, Y_specification] = get_images_labels(list(train.glob("specification/*.*")
), 14)
```

In [44]:

```
advertisement[0]
```

Out[44]:

```
array([[[ 0.52162696,  0.52162696,  0.52162696, ...,  0.52162696,
          0.52162696,  0.52162696],
        [ 0.47366433,  0.52162696,  0.52162696, ...,  0.52162696,
          0.52162696,  0.52162696],
        [ 0.39715251,  0.52162696,  0.52162696, ...,  0.52162696,
          0.52162696,  0.52162696],
        ...,
        [-0.11216497,  0.52162696,  0.52162696, ...,  0.52162696,
          0.52162696,  0.52162696],
        [-0.11521022,  0.52162696,  0.52162696, ...,  0.52162696,
          0.52162696,  0.52162696],
        [-0.11787481,  0.52162696,  0.52162696, ...,  0.52162696,
          0.52162696,  0.52162696]],

       [[-1.71396463,  0.52162696,  0.52162696, ..., -0.01738167,
         -0.09731939, -0.07790595],
        [-1.70330627,  0.52162696,  0.52162696, ..., -2.1734162 ,
         -2.5731048 , -2.47603757],
        [-1.71662922,  0.52162696,  0.52162696, ...,  0.52162696,
          0.52162696,  0.52162696],
        ...,
        [ 0.4816581 ,  0.52162696,  0.52162696, ..., -0.07790595,
         -0.07790595, -0.98386678],
        [ 0.52162696,  0.52162696,  0.52162696, ...,  0.40172038,
          0.40172038,  0.52162696],
        [ 0.52162696,  0.52162696,  0.52162696, ..., -0.89593529,
         -0.80267461, -1.7219584 ]],

       [[ 0.42874694, -0.20999351, -0.22331647, ...,  0.52162696,
          0.52162696,  0.52162696],
        [ 0.05722687, -3.13647541, -3.20309017, ...,  0.52162696,
          0.52162696,  0.52162696],
        [ 0.52162696,  0.52162696,  0.52162696, ...,  0.52162696,
          0.52162696,  0.52162696],
        ...,
        [-2.31235557, -1.8822145 , -2.02496043, ...,  0.52162696,
          0.52162696,  0.52162696],
        [ 0.52162696,  0.52162696,  0.52162696, ...,  0.52162696,
          0.52162696,  0.52162696],
        [ 0.12993213,  0.15391344, -1.16505895, ...,  0.52162696,
          0.52162696,  0.52162696]],

       [[ 0.52162696,  0.52162696,  0.52162696, ...,  0.52162696,
          0.52162696,  0.52162696],
        [ 0.52162696,  0.52162696,  0.52162696, ...,  0.52162696,
          0.52162696,  0.52162696],
        [ 0.52162696,  0.52162696,  0.52162696, ...,  0.52162696,
          0.52162696,  0.52162696],
        ...,
        [ 0.52162696,  0.52162696,  0.52162696, ...,  0.52162696,
```

```
        [ 0.52162696,  0.52162696,  0.52162696, ...,  0.52162696,
          0.52162696,  0.52162696],
        [-2.97279341, -2.97279341, -2.97279341, ..., -2.97279341,
         -2.97279341,  0.52162696],
        [-5.69067591, -5.69067591, -5.69067591, ..., -5.69067591,
         -5.69067591,  0.52162696]]])
```

In [45]:

```
images = advertisement + budget + email + file_folder + form + handwritten + invoice + l
etter + memo + news_article + presentation + questionnaire + resume + scientific_report
+ specification
labels = Y_advertisement + Y_budget + Y_email + Y_file_folder + Y_form + Y_handwritten +
Y_invoice + Y_letter + Y_memo + Y_news_article + Y_presentation + Y_questionnaire + Y_re
sume + Y_scientific_report + Y_specification
```

In [46]:

```
images = np.asarray(images)
labels = np.asarray(labels)
```

In [47]:

```
images.shape
```

Out[47]:

```
(37427, 4, 48, 48)
```

In [48]:

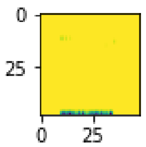
```
labels.shape
```

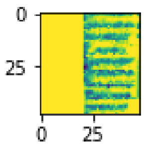
Out[48]:

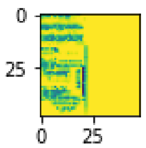
```
(37427,)
```

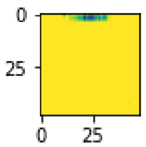
In [49]:

```
for i in range(4):
  plt.figure(figsize = (1, 1))
  plt.imshow(images[4][i])
  plt.grid(False)
  plt.show()
```









In [50]:

```
labels[0]
```

Out[50]:

```
0
```

In [51]:

```
labels = to_categorical(labels)
```

In [52]:

```
labels
```

Out[52]:

```
array([[1., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 0., 1.]], dtype=float32)
```

In [53]:

```
images.shape
```

Out[53]:

```
(37427, 4, 48, 48)
```

In [61]:

```
m = tf.keras.models.Sequential()
m.add(tf.keras.layers.TimeDistributed(tf.keras.layers.Conv2D(512, 3, activation = "relu"
), input_shape=(4, 48, 48, 1)))
m.add(tf.keras.layers.TimeDistributed(tf.keras.layers.MaxPooling2D()))
m.add(tf.keras.layers.TimeDistributed(tf.keras.layers.Dropout(0.2)))
m.add(tf.keras.layers.TimeDistributed(tf.keras.layers.Conv2D(256, 3, activation = "relu"
)))
m.add(tf.keras.layers.TimeDistributed(tf.keras.layers.MaxPooling2D()))
m.add(tf.keras.layers.TimeDistributed(tf.keras.layers.Dropout(0.2)))
m.add(tf.keras.layers.TimeDistributed(tf.keras.layers.Flatten()))
m.add(tf.keras.layers.GlobalAveragePooling1D())
m.add(tf.keras.layers.Dense(1024, activation = "sigmoid"))
m.add(tf.keras.layers.Dropout(0.2))
m.add(tf.keras.layers.Dense(15, activation = "softmax"))
```

In [62]:

```
m.summary()
```

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 time_distributed_35 (TimeDi  (None, 4, 46, 46, 512)   5120
 stributed)

 time_distributed_36 (TimeDi  (None, 4, 23, 23, 512)   0
 stributed)

 time_distributed_37 (TimeDi  (None, 4, 23, 23, 512)   0
 stributed)

 time_distributed_38 (TimeDi  (None, 4, 21, 21, 256)   1179904
 stributed)

 time_distributed_39 (TimeDi  (None, 4, 10, 10, 256)   0
 stributed)

 time_distributed_40 (TimeDi  (None, 4, 10, 10, 256)   0
 stributed)
```

```
 time_distributed_41 (TimeDi  (None, 4, 25600)         0
 stributed)

 global_average_pooling1d_3   (None, 25600)            0
 (GlobalAveragePooling1D)

 dense_5 (Dense)             (None, 1024)              26215424

 dropout_13 (Dropout)        (None, 1024)              0

 dense_6 (Dense)             (None, 15)                15375

=================================================================
Total params: 27,415,823
Trainable params: 27,415,823
Non-trainable params: 0
_________________________________________________________________
```

In [63]:

```
m.compile(optimizer= "adam", loss = 'categorical_crossentropy',
              metrics = [ TruePositives(name='tp'),
                         FalsePositives(name='fp'),
                         TrueNegatives(name='tn'),
                         FalseNegatives(name='fn'),
                         "accuracy",
                         Precision(name='precision'),
                         Recall(name='recall'),
                         AUC(name='auc')])
```

In [59]:

```
from keras.callbacks import TensorBoard, EarlyStopping
earlyStopping = EarlyStopping(monitor = 'loss', patience = 16, mode = 'min', restore_bes
t_weights = True)
```

In [64]:

```
history = m.fit(images, labels, epochs=400, batch_size= 16,
                callbacks =[earlyStopping])
```

```
Epoch 1/400
2340/2340 [==============================] - 173s 73ms/step - loss: 1.9226 - tp: 6904.000
0 - fp: 2477.0000 - tn: 521501.0000 - fn: 30523.0000 - accuracy: 0.3895 - precision: 0.73
60 - recall: 0.1845 - auc: 0.8414
Epoch 2/400
2340/2340 [==============================] - 171s 73ms/step - loss: 1.5966 - tp: 11099.00
00 - fp: 3464.0000 - tn: 520514.0000 - fn: 26328.0000 - accuracy: 0.4968 - precision: 0.7
621 - recall: 0.2966 - auc: 0.8957
Epoch 3/400
2340/2340 [==============================] - 171s 73ms/step - loss: 1.4525 - tp: 13386.00
00 - fp: 3872.0000 - tn: 520106.0000 - fn: 24041.0000 - accuracy: 0.5414 - precision: 0.7
756 - recall: 0.3577 - auc: 0.9146
Epoch 4/400
2340/2340 [==============================] - 171s 73ms/step - loss: 1.3376 - tp: 15381.00
00 - fp: 4177.0000 - tn: 519801.0000 - fn: 22046.0000 - accuracy: 0.5778 - precision: 0.7
864 - recall: 0.4110 - auc: 0.9279
Epoch 5/400
2340/2340 [==============================] - 171s 73ms/step - loss: 1.2449 - tp: 16708.00
00 - fp: 4330.0000 - tn: 519648.0000 - fn: 20719.0000 - accuracy: 0.6037 - precision: 0.7
942 - recall: 0.4464 - auc: 0.9377
Epoch 6/400
2340/2340 [==============================] - 171s 73ms/step - loss: 1.1578 - tp: 18106.00
00 - fp: 4440.0000 - tn: 519538.0000 - fn: 19321.0000 - accuracy: 0.6299 - precision: 0.8
031 - recall: 0.4838 - auc: 0.9464
Epoch 7/400
2340/2340 [==============================] - 171s 73ms/step - loss: 1.0983 - tp: 19107.00
00 - fp: 4485.0000 - tn: 519493.0000 - fn: 18320.0000 - accuracy: 0.6501 - precision: 0.8
099 - recall: 0.5105 - auc: 0.9517
Epoch 8/400
2340/2340 [==============================] - 171s 73ms/step - loss: 1.0195 - tp: 20317.00
```

```
2340/2340 [==============================] - 171s 73ms/step - loss: 1.0193 - tp: 20317.00
00 - fp: 4503.0000 - tn: 519475.0000 - fn: 17110.0000 - accuracy: 0.6735 - precision: 0.8
186 - recall: 0.5428 - auc: 0.9585
Epoch 9/400
2340/2340 [==============================] - 171s 73ms/step - loss: 0.9645 - tp: 21182.00
00 - fp: 4612.0000 - tn: 519366.0000 - fn: 16245.0000 - accuracy: 0.6902 - precision: 0.8
212 - recall: 0.5660 - auc: 0.9628
Epoch 10/400
2340/2340 [==============================] - 170s 73ms/step - loss: 0.9164 - tp: 22098.00
00 - fp: 4543.0000 - tn: 519435.0000 - fn: 15329.0000 - accuracy: 0.7058 - precision: 0.8
295 - recall: 0.5904 - auc: 0.9663
Epoch 11/400
2340/2340 [==============================] - 170s 73ms/step - loss: 0.8701 - tp: 22911.00
00 - fp: 4592.0000 - tn: 519386.0000 - fn: 14516.0000 - accuracy: 0.7196 - precision: 0.8
330 - recall: 0.6122 - auc: 0.9695
Epoch 12/400
2340/2340 [==============================] - 170s 73ms/step - loss: 0.8169 - tp: 23730.00
00 - fp: 4495.0000 - tn: 519483.0000 - fn: 13697.0000 - accuracy: 0.7347 - precision: 0.8
407 - recall: 0.6340 - auc: 0.9731
Epoch 13/400
2340/2340 [==============================] - 170s 73ms/step - loss: 0.7801 - tp: 24439.00
00 - fp: 4484.0000 - tn: 519494.0000 - fn: 12988.0000 - accuracy: 0.7450 - precision: 0.8
450 - recall: 0.6530 - auc: 0.9754
Epoch 14/400
2340/2340 [==============================] - 171s 73ms/step - loss: 0.7507 - tp: 25061.00
00 - fp: 4496.0000 - tn: 519482.0000 - fn: 12366.0000 - accuracy: 0.7556 - precision: 0.8
479 - recall: 0.6696 - auc: 0.9769
Epoch 15/400
2340/2340 [==============================] - 170s 73ms/step - loss: 0.7225 - tp: 25538.00
00 - fp: 4492.0000 - tn: 519486.0000 - fn: 11889.0000 - accuracy: 0.7638 - precision: 0.8
504 - recall: 0.6823 - auc: 0.9784
Epoch 16/400
2340/2340 [==============================] - 170s 73ms/step - loss: 0.6912 - tp: 26069.00
00 - fp: 4373.0000 - tn: 519605.0000 - fn: 11358.0000 - accuracy: 0.7737 - precision: 0.8
563 - recall: 0.6965 - auc: 0.9802
Epoch 17/400
2340/2340 [==============================] - 170s 73ms/step - loss: 0.6716 - tp: 26496.00
00 - fp: 4434.0000 - tn: 519544.0000 - fn: 10931.0000 - accuracy: 0.7802 - precision: 0.8
566 - recall: 0.7079 - auc: 0.9814
Epoch 18/400
2340/2340 [==============================] - 170s 73ms/step - loss: 0.6426 - tp: 26932.00
00 - fp: 4385.0000 - tn: 519593.0000 - fn: 10495.0000 - accuracy: 0.7894 - precision: 0.8
600 - recall: 0.7196 - auc: 0.9827
Epoch 19/400
2340/2340 [==============================] - 170s 73ms/step - loss: 0.6170 - tp: 27352.00
00 - fp: 4282.0000 - tn: 519696.0000 - fn: 10075.0000 - accuracy: 0.7950 - precision: 0.8
646 - recall: 0.7308 - auc: 0.9841
Epoch 20/400
2340/2340 [==============================] - 170s 73ms/step - loss: 0.5941 - tp: 27724.00
00 - fp: 4211.0000 - tn: 519767.0000 - fn: 9703.0000 - accuracy: 0.8039 - precision: 0.86
81 - recall: 0.7407 - auc: 0.9848
Epoch 21/400
2340/2340 [==============================] - 170s 73ms/step - loss: 0.5766 - tp: 28253.00
00 - fp: 4198.0000 - tn: 519780.0000 - fn: 9174.0000 - accuracy: 0.8112 - precision: 0.87
06 - recall: 0.7549 - auc: 0.9856
Epoch 22/400
2340/2340 [==============================] - 168s 72ms/step - loss: 0.5625 - tp: 28429.00
00 - fp: 4170.0000 - tn: 519808.0000 - fn: 8998.0000 - accuracy: 0.8133 - precision: 0.87
21 - recall: 0.7596 - auc: 0.9861
Epoch 23/400
2340/2340 [==============================] - 168s 72ms/step - loss: 0.5354 - tp: 28863.00
00 - fp: 4019.0000 - tn: 519959.0000 - fn: 8564.0000 - accuracy: 0.8242 - precision: 0.87
78 - recall: 0.7712 - auc: 0.9874
Epoch 24/400
2340/2340 [==============================] - 169s 72ms/step - loss: 0.5125 - tp: 29258.00
00 - fp: 3830.0000 - tn: 520148.0000 - fn: 8169.0000 - accuracy: 0.8310 - precision: 0.88
42 - recall: 0.7817 - auc: 0.9884
Epoch 25/400
2340/2340 [==============================] - 168s 72ms/step - loss: 0.5072 - tp: 29476.00
00 - fp: 3897.0000 - tn: 520081.0000 - fn: 7951.0000 - accuracy: 0.8344 - precision: 0.88
32 - recall: 0.7876 - auc: 0.9883
Epoch 26/400
2340/2340 [==============================] - 168s 72ms/step - loss: 0.4855 - tp: 29798.00
```

```
2340/2340 [==============================] - 168s 72ms/step - loss: 0.4855 - tp: 29798.00
00 - fp: 3849.0000 - tn: 520129.0000 - fn: 7629.0000 - accuracy: 0.8413 - precision: 0.88
56 - recall: 0.7962 - auc: 0.9891
Epoch 27/400
2340/2340 [==============================] - 168s 72ms/step - loss: 0.4727 - tp: 30091.00
00 - fp: 3737.0000 - tn: 520241.0000 - fn: 7336.0000 - accuracy: 0.8448 - precision: 0.88
95 - recall: 0.8040 - auc: 0.9895
Epoch 28/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.4582 - tp: 30270.00
00 - fp: 3708.0000 - tn: 520270.0000 - fn: 7157.0000 - accuracy: 0.8505 - precision: 0.89
09 - recall: 0.8088 - auc: 0.9902
Epoch 29/400
2340/2340 [==============================] - 168s 72ms/step - loss: 0.4524 - tp: 30402.00
00 - fp: 3681.0000 - tn: 520297.0000 - fn: 7025.0000 - accuracy: 0.8505 - precision: 0.89
20 - recall: 0.8123 - auc: 0.9904
Epoch 30/400
2340/2340 [==============================] - 167s 72ms/step - loss: 0.4321 - tp: 30701.00
00 - fp: 3536.0000 - tn: 520442.0000 - fn: 6726.0000 - accuracy: 0.8586 - precision: 0.89
67 - recall: 0.8203 - auc: 0.9911
Epoch 31/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.4255 - tp: 30850.00
00 - fp: 3597.0000 - tn: 520381.0000 - fn: 6577.0000 - accuracy: 0.8591 - precision: 0.89
56 - recall: 0.8243 - auc: 0.9911
Epoch 32/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.4088 - tp: 31108.00
00 - fp: 3466.0000 - tn: 520512.0000 - fn: 6319.0000 - accuracy: 0.8652 - precision: 0.89
98 - recall: 0.8312 - auc: 0.9917
Epoch 33/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.4081 - tp: 31106.00
00 - fp: 3511.0000 - tn: 520467.0000 - fn: 6321.0000 - accuracy: 0.8646 - precision: 0.89
86 - recall: 0.8311 - auc: 0.9919
Epoch 34/400
2340/2340 [==============================] - 168s 72ms/step - loss: 0.3979 - tp: 31303.00
00 - fp: 3392.0000 - tn: 520586.0000 - fn: 6124.0000 - accuracy: 0.8688 - precision: 0.90
22 - recall: 0.8364 - auc: 0.9922
Epoch 35/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3937 - tp: 31441.00
00 - fp: 3400.0000 - tn: 520578.0000 - fn: 5986.0000 - accuracy: 0.8708 - precision: 0.90
24 - recall: 0.8401 - auc: 0.9921
Epoch 36/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3798 - tp: 31674.00
00 - fp: 3342.0000 - tn: 520636.0000 - fn: 5753.0000 - accuracy: 0.8753 - precision: 0.90
46 - recall: 0.8463 - auc: 0.9927
Epoch 37/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3865 - tp: 31584.00
00 - fp: 3421.0000 - tn: 520557.0000 - fn: 5843.0000 - accuracy: 0.8722 - precision: 0.90
23 - recall: 0.8439 - auc: 0.9923
Epoch 38/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3601 - tp: 32002.00
00 - fp: 3176.0000 - tn: 520802.0000 - fn: 5425.0000 - accuracy: 0.8807 - precision: 0.90
97 - recall: 0.8551 - auc: 0.9932
Epoch 39/400
2340/2340 [==============================] - 168s 72ms/step - loss: 0.3629 - tp: 31944.00
00 - fp: 3265.0000 - tn: 520713.0000 - fn: 5483.0000 - accuracy: 0.8795 - precision: 0.90
73 - recall: 0.8535 - auc: 0.9931
Epoch 40/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3607 - tp: 32018.00
00 - fp: 3262.0000 - tn: 520716.0000 - fn: 5409.0000 - accuracy: 0.8810 - precision: 0.90
75 - recall: 0.8555 - auc: 0.9930
Epoch 41/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3502 - tp: 32227.00
00 - fp: 3129.0000 - tn: 520849.0000 - fn: 5200.0000 - accuracy: 0.8846 - precision: 0.91
15 - recall: 0.8611 - auc: 0.9934
Epoch 42/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3424 - tp: 32264.00
00 - fp: 3182.0000 - tn: 520796.0000 - fn: 5163.0000 - accuracy: 0.8856 - precision: 0.91
02 - recall: 0.8621 - auc: 0.9933
Epoch 43/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3438 - tp: 32292.00
00 - fp: 3114.0000 - tn: 520864.0000 - fn: 5135.0000 - accuracy: 0.8870 - precision: 0.91
20 - recall: 0.8628 - auc: 0.9934
Epoch 44/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3323 - tp: 32521.00
```

```
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3323 - tp: 32521.00
00 - fp: 3024.0000 - tn: 520954.0000 - fn: 4906.0000 - accuracy: 0.8913 - precision: 0.91
49 - recall: 0.8689 - auc: 0.9938
Epoch 45/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3222 - tp: 32677.00
00 - fp: 2954.0000 - tn: 521024.0000 - fn: 4750.0000 - accuracy: 0.8948 - precision: 0.91
71 - recall: 0.8731 - auc: 0.9938
Epoch 46/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3326 - tp: 32561.00
00 - fp: 3047.0000 - tn: 520931.0000 - fn: 4866.0000 - accuracy: 0.8912 - precision: 0.91
44 - recall: 0.8700 - auc: 0.9934
Epoch 47/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3270 - tp: 32643.00
00 - fp: 3041.0000 - tn: 520937.0000 - fn: 4784.0000 - accuracy: 0.8925 - precision: 0.91
48 - recall: 0.8722 - auc: 0.9937
Epoch 48/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3178 - tp: 32835.00
00 - fp: 2929.0000 - tn: 521049.0000 - fn: 4592.0000 - accuracy: 0.8964 - precision: 0.91
81 - recall: 0.8773 - auc: 0.9938
Epoch 49/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.3200 - tp: 32798.00
00 - fp: 3004.0000 - tn: 520974.0000 - fn: 4629.0000 - accuracy: 0.8959 - precision: 0.91
61 - recall: 0.8763 - auc: 0.9937
Epoch 50/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3297 - tp: 32701.00
00 - fp: 3078.0000 - tn: 520900.0000 - fn: 4726.0000 - accuracy: 0.8928 - precision: 0.91
40 - recall: 0.8737 - auc: 0.9932
Epoch 51/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3189 - tp: 32897.00
00 - fp: 2980.0000 - tn: 520998.0000 - fn: 4530.0000 - accuracy: 0.8979 - precision: 0.91
69 - recall: 0.8790 - auc: 0.9934
Epoch 52/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3060 - tp: 33027.00
00 - fp: 2892.0000 - tn: 521086.0000 - fn: 4400.0000 - accuracy: 0.8996 - precision: 0.91
95 - recall: 0.8824 - auc: 0.9943
Epoch 53/400
2340/2340 [==============================] - 167s 71ms/step - loss: 0.3087 - tp: 32925.00
00 - fp: 2974.0000 - tn: 521004.0000 - fn: 4502.0000 - accuracy: 0.8973 - precision: 0.91
72 - recall: 0.8797 - auc: 0.9940
Epoch 54/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2974 - tp: 33132.00
00 - fp: 2828.0000 - tn: 521150.0000 - fn: 4295.0000 - accuracy: 0.9022 - precision: 0.92
14 - recall: 0.8852 - auc: 0.9946
Epoch 55/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2973 - tp: 33132.00
00 - fp: 2868.0000 - tn: 521110.0000 - fn: 4295.0000 - accuracy: 0.9035 - precision: 0.92
03 - recall: 0.8852 - auc: 0.9945
Epoch 56/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.3010 - tp: 33105.00
00 - fp: 2877.0000 - tn: 521101.0000 - fn: 4322.0000 - accuracy: 0.9008 - precision: 0.92
00 - recall: 0.8845 - auc: 0.9941
Epoch 57/400
2340/2340 [==============================] - 165s 71ms/step - loss: 0.2984 - tp: 33195.00
00 - fp: 2830.0000 - tn: 521148.0000 - fn: 4232.0000 - accuracy: 0.9035 - precision: 0.92
14 - recall: 0.8869 - auc: 0.9942
Epoch 58/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.3013 - tp: 33183.00
00 - fp: 2883.0000 - tn: 521095.0000 - fn: 4244.0000 - accuracy: 0.9019 - precision: 0.92
01 - recall: 0.8866 - auc: 0.9940
Epoch 59/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.3025 - tp: 33150.00
00 - fp: 2891.0000 - tn: 521087.0000 - fn: 4277.0000 - accuracy: 0.9022 - precision: 0.91
98 - recall: 0.8857 - auc: 0.9941
Epoch 60/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.3004 - tp: 33192.00
00 - fp: 2899.0000 - tn: 521079.0000 - fn: 4235.0000 - accuracy: 0.9019 - precision: 0.91
97 - recall: 0.8868 - auc: 0.9940
Epoch 61/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2997 - tp: 33251.00
00 - fp: 2810.0000 - tn: 521168.0000 - fn: 4176.0000 - accuracy: 0.9034 - precision: 0.92
21 - recall: 0.8884 - auc: 0.9941
Epoch 62/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2899 - tp: 33362.00
```

```
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2899 - tp: 33362.00
00 - fp: 2777.0000 - tn: 521201.0000 - fn: 4065.0000 - accuracy: 0.9060 - precision: 0.92
32 - recall: 0.8914 - auc: 0.9943
Epoch 63/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.3044 - tp: 33213.00
00 - fp: 2938.0000 - tn: 521040.0000 - fn: 4214.0000 - accuracy: 0.9018 - precision: 0.91
87 - recall: 0.8874 - auc: 0.9937
Epoch 64/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2936 - tp: 33321.00
00 - fp: 2862.0000 - tn: 521116.0000 - fn: 4106.0000 - accuracy: 0.9048 - precision: 0.92
09 - recall: 0.8903 - auc: 0.9942
Epoch 65/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2789 - tp: 33460.00
00 - fp: 2675.0000 - tn: 521303.0000 - fn: 3967.0000 - accuracy: 0.9086 - precision: 0.92
60 - recall: 0.8940 - auc: 0.9949
Epoch 66/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2899 - tp: 33355.00
00 - fp: 2787.0000 - tn: 521191.0000 - fn: 4072.0000 - accuracy: 0.9065 - precision: 0.92
29 - recall: 0.8912 - auc: 0.9942
Epoch 67/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2885 - tp: 33406.00
00 - fp: 2839.0000 - tn: 521139.0000 - fn: 4021.0000 - accuracy: 0.9066 - precision: 0.92
17 - recall: 0.8926 - auc: 0.9942
Epoch 68/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2836 - tp: 33462.00
00 - fp: 2782.0000 - tn: 521196.0000 - fn: 3965.0000 - accuracy: 0.9079 - precision: 0.92
32 - recall: 0.8941 - auc: 0.9945
Epoch 69/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2836 - tp: 33470.00
00 - fp: 2756.0000 - tn: 521222.0000 - fn: 3957.0000 - accuracy: 0.9083 - precision: 0.92
39 - recall: 0.8943 - auc: 0.9942
Epoch 70/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2794 - tp: 33554.00
00 - fp: 2704.0000 - tn: 521274.0000 - fn: 3873.0000 - accuracy: 0.9097 - precision: 0.92
54 - recall: 0.8965 - auc: 0.9945
Epoch 71/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2876 - tp: 33488.00
00 - fp: 2755.0000 - tn: 521223.0000 - fn: 3939.0000 - accuracy: 0.9078 - precision: 0.92
40 - recall: 0.8948 - auc: 0.9943
Epoch 72/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2891 - tp: 33439.00
00 - fp: 2812.0000 - tn: 521166.0000 - fn: 3988.0000 - accuracy: 0.9065 - precision: 0.92
24 - recall: 0.8934 - auc: 0.9945
Epoch 73/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2662 - tp: 33678.00
00 - fp: 2654.0000 - tn: 521324.0000 - fn: 3749.0000 - accuracy: 0.9115 - precision: 0.92
70 - recall: 0.8998 - auc: 0.9951
Epoch 74/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2723 - tp: 33651.00
00 - fp: 2701.0000 - tn: 521277.0000 - fn: 3776.0000 - accuracy: 0.9123 - precision: 0.92
57 - recall: 0.8991 - auc: 0.9945
Epoch 75/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2781 - tp: 33568.00
00 - fp: 2738.0000 - tn: 521240.0000 - fn: 3859.0000 - accuracy: 0.9095 - precision: 0.92
46 - recall: 0.8969 - auc: 0.9943
Epoch 76/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2821 - tp: 33525.00
00 - fp: 2754.0000 - tn: 521224.0000 - fn: 3902.0000 - accuracy: 0.9090 - precision: 0.92
41 - recall: 0.8957 - auc: 0.9942
Epoch 77/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2804 - tp: 33602.00
00 - fp: 2681.0000 - tn: 521297.0000 - fn: 3825.0000 - accuracy: 0.9110 - precision: 0.92
61 - recall: 0.8978 - auc: 0.9943
Epoch 78/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2841 - tp: 33543.00
00 - fp: 2800.0000 - tn: 521178.0000 - fn: 3884.0000 - accuracy: 0.9088 - precision: 0.92
30 - recall: 0.8962 - auc: 0.9942
Epoch 79/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2795 - tp: 33601.00
00 - fp: 2763.0000 - tn: 521215.0000 - fn: 3826.0000 - accuracy: 0.9095 - precision: 0.92
40 - recall: 0.8978 - auc: 0.9942
Epoch 80/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2894 - tp: 33517.00
```

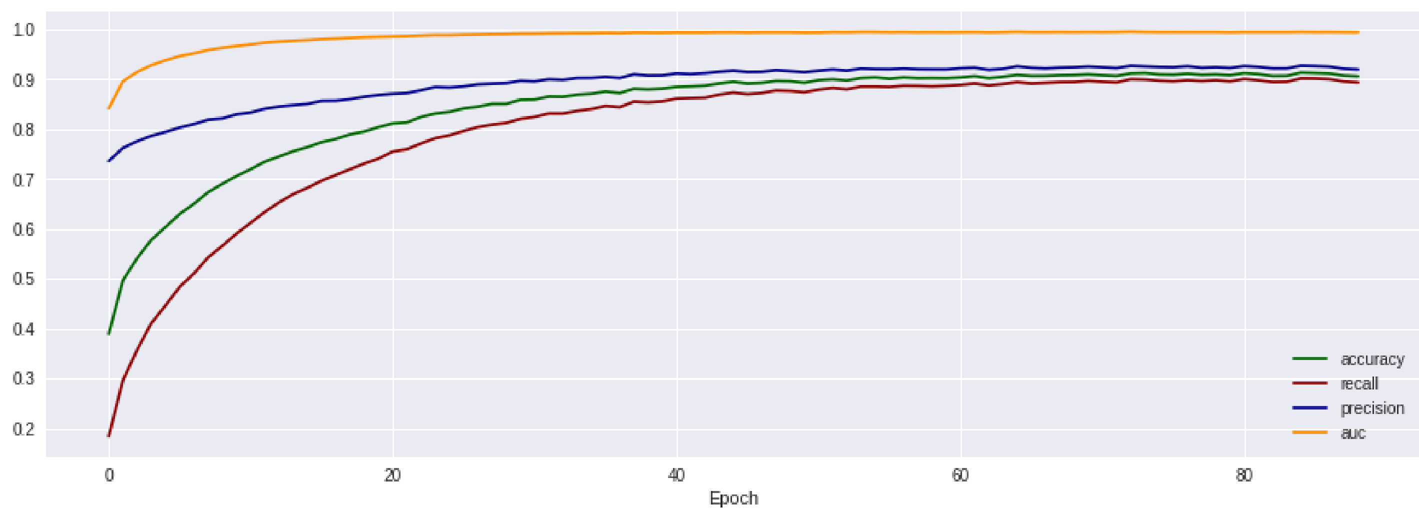
```
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2891 - tp: 33517.00
00 - fp: 2810.0000 - tn: 521168.0000 - fn: 3910.0000 - accuracy: 0.9080 - precision: 0.92
26 - recall: 0.8955 - auc: 0.9938
Epoch 81/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2741 - tp: 33693.00
00 - fp: 2678.0000 - tn: 521300.0000 - fn: 3734.0000 - accuracy: 0.9121 - precision: 0.92
64 - recall: 0.9002 - auc: 0.9942
Epoch 82/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2808 - tp: 33594.00
00 - fp: 2737.0000 - tn: 521241.0000 - fn: 3833.0000 - accuracy: 0.9101 - precision: 0.92
47 - recall: 0.8976 - auc: 0.9942
Epoch 83/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2866 - tp: 33471.00
00 - fp: 2841.0000 - tn: 521137.0000 - fn: 3956.0000 - accuracy: 0.9063 - precision: 0.92
18 - recall: 0.8943 - auc: 0.9943
Epoch 84/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2847 - tp: 33487.00
00 - fp: 2842.0000 - tn: 521136.0000 - fn: 3940.0000 - accuracy: 0.9067 - precision: 0.92
18 - recall: 0.8947 - auc: 0.9942
Epoch 85/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2736 - tp: 33738.00
00 - fp: 2650.0000 - tn: 521328.0000 - fn: 3689.0000 - accuracy: 0.9132 - precision: 0.92
72 - recall: 0.9014 - auc: 0.9946
Epoch 86/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2756 - tp: 33730.00
00 - fp: 2680.0000 - tn: 521298.0000 - fn: 3697.0000 - accuracy: 0.9123 - precision: 0.92
64 - recall: 0.9012 - auc: 0.9943
Epoch 87/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2739 - tp: 33681.00
00 - fp: 2722.0000 - tn: 521256.0000 - fn: 3746.0000 - accuracy: 0.9113 - precision: 0.92
52 - recall: 0.8999 - auc: 0.9944
Epoch 88/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2873 - tp: 33510.00
00 - fp: 2871.0000 - tn: 521107.0000 - fn: 3917.0000 - accuracy: 0.9072 - precision: 0.92
11 - recall: 0.8953 - auc: 0.9941
Epoch 89/400
2340/2340 [==============================] - 166s 71ms/step - loss: 0.2911 - tp: 33437.00
00 - fp: 2929.0000 - tn: 521049.0000 - fn: 3990.0000 - accuracy: 0.9057 - precision: 0.91
95 - recall: 0.8934 - auc: 0.9939
```

In [65]:

```
import matplotlib as mpl
mpl.style.use('seaborn')
plt.figure(figsize = (15, 5))
plt.plot(history.history['accuracy'], "darkgreen", label= "accuracy")
plt.plot(history.history['recall'], "darkred", label= "recall")
plt.plot(history.history['precision'], "darkblue", label= "precision")
plt.plot(history.history['auc'], "darkorange", label= "auc")
plt.xlabel('Epoch')
plt.legend()
```

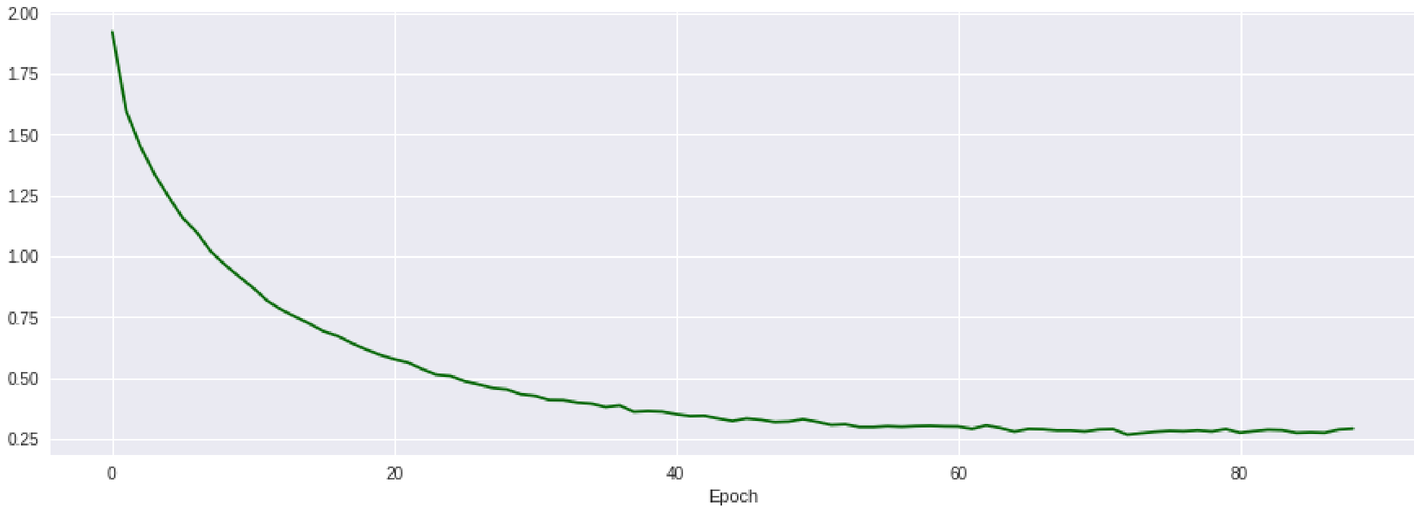
Out[65]:

<matplotlib.legend.Legend at 0x7fa34d46a690>



In [66]:

```
mpl.style.use('seaborn')
plt.figure(figsize = (15, 5))
plt.plot(history.history['loss'], "darkgreen", label= "accuracy")
plt.xlabel('Epoch')
plt.show()
```



In [69]:

```
m.evaluate(images, labels, batch_size=16)
```

```
2340/2340 [==============================] - 50s 21ms/step - loss: 0.0258 - tp: 37203.000
0 - fp: 130.0000 - tn: 523848.0000 - fn: 224.0000 - accuracy: 0.9955 - precision: 0.9965
- recall: 0.9940 - auc: 0.9999
```

Out[69]:

```
[0.025828829035162926,
 37203.0,
 130.0,
 523848.0,
 224.0,
 0.9954845309257507,
 0.9965178370475769,
 0.9940150380134583,
 0.9998602271080017]
```

In [67]:

```
y_pred = m.predict(images, batch_size=16, verbose= 1)
y_pred = np.argmax(y_pred, axis = 1)
```

```
2340/2340 [==============================] - 44s 19ms/step
```

In [70]:

```
y_test = np.argmax(labels, axis = 1)
y_test
```

Out[70]:

```
array([ 0,  0,  0, ..., 14, 14, 14])
```

In [76]:

```
from sklearn.metrics import accuracy_score, classification_report
accuracy_score(y_test, y_pred)
```

Out[76]:

```
0.995484543244182
```

In [77]:

```python
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      2515
           1       0.99      1.00      1.00      2505
           2       1.00      1.00      1.00      2516
           3       0.99      1.00      0.99      2527
           4       0.99      1.00      0.99      2506
           5       1.00      1.00      1.00      2532
           6       1.00      0.99      0.99      2477
           7       1.00      1.00      1.00      2464
           8       1.00      1.00      1.00      2492
           9       1.00      0.99      1.00      2463
          10       1.00      0.99      0.99      2489
          11       0.99      1.00      1.00      2435
          12       1.00      1.00      1.00      2536
          13       1.00      0.99      0.99      2498
          14       1.00      1.00      1.00      2472

    accuracy                           1.00     37427
   macro avg       1.00      1.00      1.00     37427
weighted avg       1.00      1.00      1.00     37427
```

In [72]:

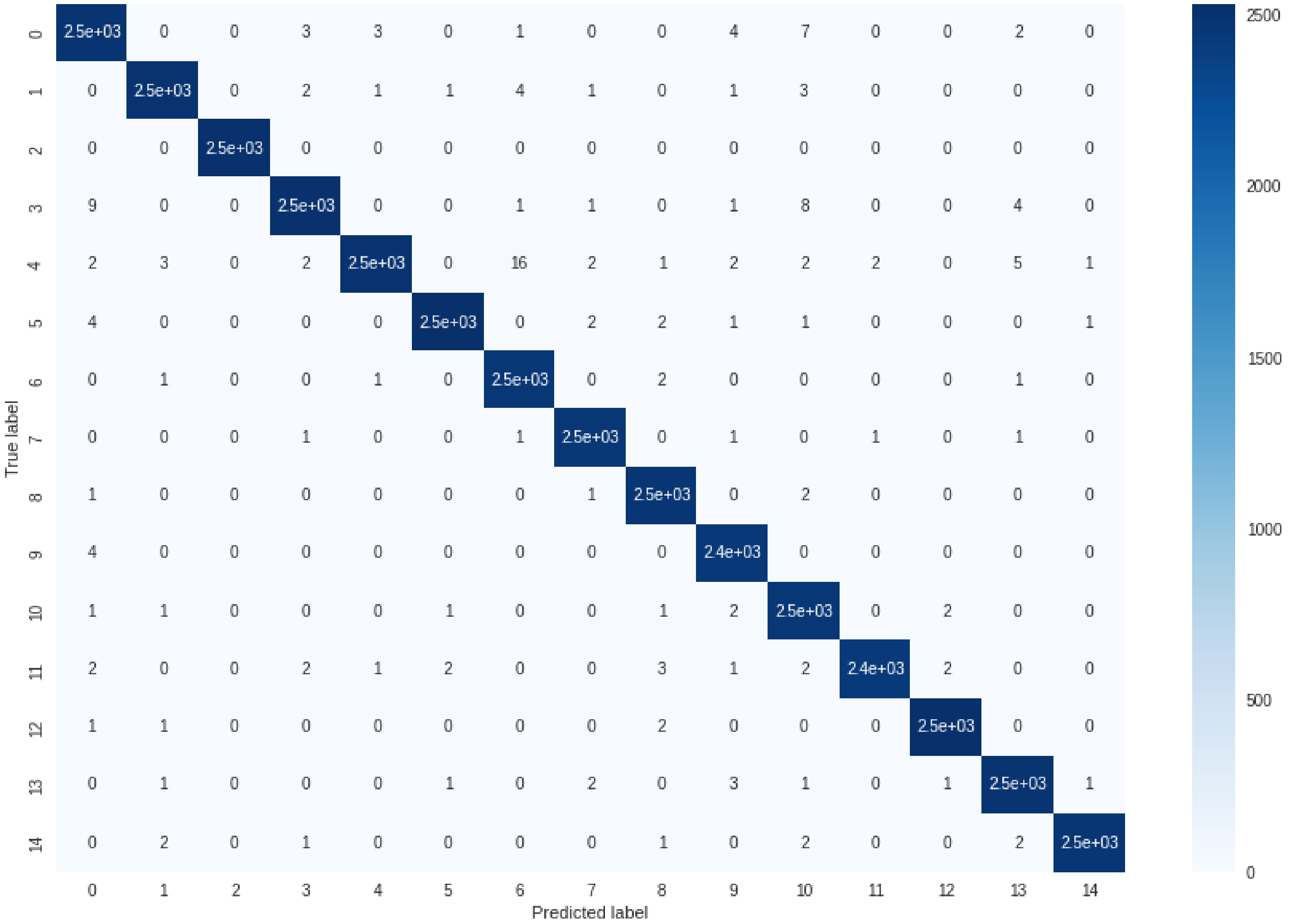
```python
classes = list(range(15))
res = tf.math.confusion_matrix(y_pred,y_test).numpy()
cm = pd.DataFrame(res,
                     index = classes,
                     columns = classes)
cm
```

Out[72]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	2491	0	0	3	3	0	1	0	0	4	7	0	0	2	0
1	0	2496	0	2	1	1	4	1	0	1	3	0	0	0	0
2	0	0	2516	0	0	0	0	0	0	0	0	0	0	0	0
3	9	0	0	2516	0	0	1	1	0	1	8	0	0	4	0
4	2	3	0	2	2500	0	16	2	1	2	2	2	0	5	1
5	4	0	0	0	0	2527	0	2	2	1	1	0	0	0	1
6	0	1	0	0	1	0	2454	0	2	0	0	0	0	1	0
7	0	0	0	1	0	0	1	2455	0	1	0	1	0	1	0
8	1	0	0	0	0	0	0	1	2480	0	2	0	0	0	0
9	4	0	0	0	0	0	0	0	0	2447	0	0	0	0	0
10	1	1	0	0	0	1	0	0	1	2	2461	0	2	0	0
11	2	0	0	2	1	2	0	0	3	1	2	2432	2	0	0
12	1	1	0	0	0	0	0	0	2	0	0	0	2531	0	0
13	0	1	0	0	0	1	0	2	0	3	1	0	1	2483	1
14	0	2	0	1	0	0	0	0	1	0	2	0	0	2	2469

In [74]:

```python
import seaborn as sns
figure = plt.figure(figsize=(15, 10))
sns.heatmap(cm, annot=True, cmap=plt.cm.Blues)
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.show()
```


True label
Predicted label
2500
2000
1500
1000
500
0