sna-project1

June 16, 2023

1 Image labeling on social network metadata

Package import

[ ]:

```
import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing import image
from tensorflow.keras.layers import Flatten, Dense, Dropout, BatchNormalization, Conv2D, MaxPool2D
print(tf.__version__)
```

```
2.12.0
```

[ ]:

```
import numpy as np
import pandas as pd
from tqdm import tqdm
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

Dataset import

[ ]:

```
data = pd.read_csv('/content/drive/MyDrive/Movies-Poster_Dataset-master/train.csv')
data.shape
```

[ ]:

```
(99, 27)
```

[ ]:

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

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

Tags overview

[ ]:

```
data.head()
```

```
[ ]:          Id                                 Genre  Action  Adventure  \
     0  tt0086425                   ['Comedy', 'Drama']       0          0
     1  tt0085549        ['Drama', 'Romance', 'Music']       0          0
     2  tt0086465                            ['Comedy']       0          0
     3  tt0086567                 ['Sci-Fi', 'Thriller']       0          0
     4  tt0086034  ['Action', 'Adventure', 'Thriller']       1          1

        Animation  Biography  Comedy  Crime  Documentary  Drama  …  N/A  News  \
     0          0          0       1      0            0      1  …    0     0
     1          0          0       0      0            0      1  …    0     0
     2          0          0       1      0            0      0  …    0     0
     3          0          0       0      0            0      0  …    0     0
     4          0          0       0      0            0      0  …    0     0

        Reality-TV  Romance  Sci-Fi  Short  Sport  Thriller  War  Western
     0           0        0       0      0      0         0    0        0
     1           0        1       0      0      0         0    0        0
     2           0        0       0      0      0         0    0        0
     3           0        0       1      0      0         1    0        0
     4           0        0       0      0      0         1    0        0

     [5 rows x 27 columns]
```

Processing images in batch

```
[ ]: img_width = 350
     img_height = 350

     X = []

     for i in tqdm(range(data.shape[0])):
       path = '/content/drive/MyDrive/Movies-Poster_Dataset-master/Images/' +␣
      ↪data['Id'][i] + '.jpg'
       img = image.load_img(path, target_size=(img_width, img_height, 3))
       img = image.img_to_array(img)
       img = img/255.0
       X.append(img)

     X = np.array(X)
```

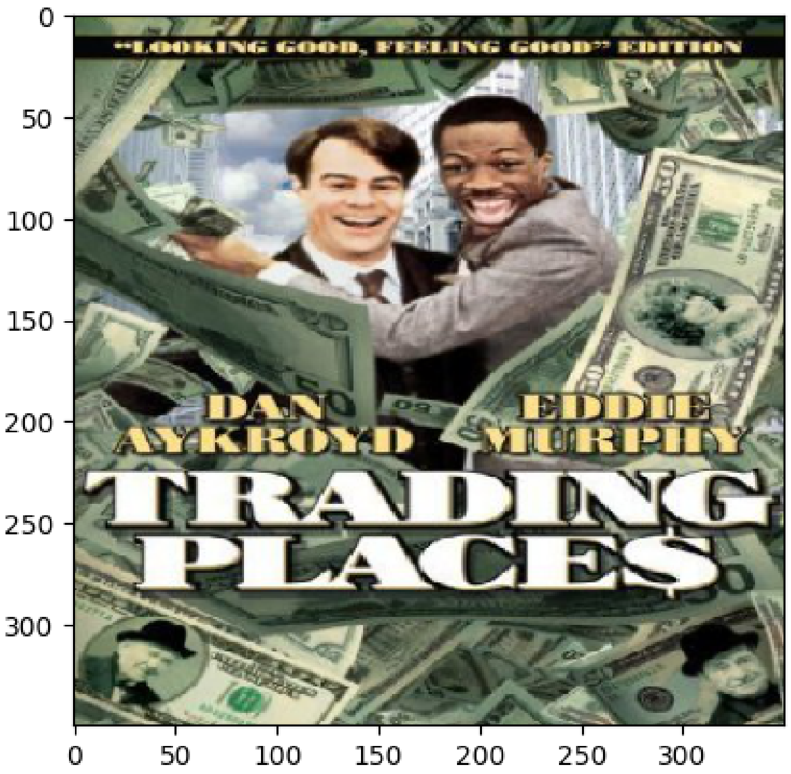
```
100%|      | 99/99 [00:00<00:00, 163.72it/s]
```

```
[ ]: X.shape
```

```
[ ]: (99, 350, 350, 3)
```

```
[ ]: plt.imshow(X[2])
```

[ ]:
```
<matplotlib.image.AxesImage at 0x7f8b3aff8460>
```



Labelling image

[ ]:
```
data['Genre'][2]
```

[ ]:
```
"['Comedy']"
```

[ ]:
```
y = data.drop(['Id', 'Genre'], axis = 1)
y = y.to_numpy()
y.shape
```

[ ]:
```
(99, 25)
```

Splitting dataset

[ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 0,␣
  ↪test_size = 0.15)
```

[ ]:
```
X_train[0].shape
```

[ ]:
```
(350, 350, 3)
```

Building CNN network

[ ]:

```
model = Sequential()
model.add(Conv2D(16, (3,3), activation='relu', input_shape = X_train[0].shape))
model.add(BatchNormalization())
model.add(MaxPool2D(2,2))
model.add(Dropout(0.3))

model.add(Conv2D(32, (3,3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(2,2))
model.add(Dropout(0.3))

model.add(Conv2D(64, (3,3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(2,2))
model.add(Dropout(0.4))

model.add(Conv2D(128, (3,3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPool2D(2,2))
model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(128, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))


model.add(Dense(128, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))


model.add(Dense(25, activation='sigmoid'))
```

Parameter summary

[ ]:

```
model.summary()
```

```
Model: "sequential_1"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 348, 348, 16)      448


 batch_normalization_6 (Batc  (None, 348, 348, 16)     64
 hNormalization)

 max_pooling2d_4 (MaxPooling  (None, 174, 174, 16)     0
 2D)

 dropout_6 (Dropout)         (None, 174, 174, 16)      0

 conv2d_5 (Conv2D)           (None, 172, 172, 32)      4640

 batch_normalization_7 (Batc  (None, 172, 172, 32)     128
 hNormalization)

 max_pooling2d_5 (MaxPooling  (None, 86, 86, 32)       0
 2D)

 dropout_7 (Dropout)         (None, 86, 86, 32)        0

 conv2d_6 (Conv2D)           (None, 84, 84, 64)        18496

 batch_normalization_8 (Batc  (None, 84, 84, 64)       256
 hNormalization)

 max_pooling2d_6 (MaxPooling  (None, 42, 42, 64)       0
 2D)

 dropout_8 (Dropout)         (None, 42, 42, 64)        0

 conv2d_7 (Conv2D)           (None, 40, 40, 128)       73856

 batch_normalization_9 (Batc  (None, 40, 40, 128)      512
 hNormalization)

 max_pooling2d_7 (MaxPooling  (None, 20, 20, 128)      0
 2D)

 dropout_9 (Dropout)         (None, 20, 20, 128)       0

 flatten_1 (Flatten)         (None, 51200)             0

 dense_3 (Dense)             (None, 128)               6553728

 batch_normalization_10 (Bat  (None, 128)              512
 chNormalization)

 dropout_10 (Dropout)        (None, 128)               0

 dense_4 (Dense)             (None, 128)               16512
```

```
 batch_normalization_11 (Bat  (None, 128)              512
 chNormalization)

 dropout_11 (Dropout)        (None, 128)               0

 dense_5 (Dense)             (None, 25)                3225

=================================================================
Total params: 6,672,889
Trainable params: 6,671,897
Non-trainable params: 992
_________________________________________________________________
```

Training dataset

```
[ ]: model.compile(optimizer='adam', loss = 'binary_crossentropy',␣
      ↪metrics=['accuracy'])
```

```
[ ]: history = model.fit(X_train, y_train, epochs=5, validation_data=(X_test,␣
      ↪y_test))
```

```
Epoch 1/5
3/3 [==============================] - 25s 7s/step - loss: 0.9768 - accuracy:
0.0833 - val_loss: 0.6755 - val_accuracy: 0.4667
Epoch 2/5
3/3 [==============================] - 20s 6s/step - loss: 0.9406 - accuracy:
0.0595 - val_loss: 0.6901 - val_accuracy: 0.0000e+00
Epoch 3/5
3/3 [==============================] - 20s 6s/step - loss: 0.9265 - accuracy:
0.0833 - val_loss: 0.7903 - val_accuracy: 0.0000e+00
Epoch 4/5
3/3 [==============================] - 23s 7s/step - loss: 0.9340 - accuracy:
0.0357 - val_loss: 0.9271 - val_accuracy: 0.0000e+00
Epoch 5/5
3/3 [==============================] - 19s 7s/step - loss: 0.8965 - accuracy:
0.0952 - val_loss: 1.0717 - val_accuracy: 0.0000e+00
```

Plot for training and validation loss

```
[ ]: def plot_learningCurve(history, epoch):
       epoch_range = range(1, epoch+1)

       plt.plot(epoch_range, history.history['loss'])
       plt.plot(epoch_range, history.history['val_loss'])
       plt.title('Model loss')
       plt.ylabel('Loss')
       plt.xlabel('Epoch')
       plt.legend(['Train', 'Val'], loc='upper left')
       plt.show()
```

```
plot_learningCurve(history, 5)
```

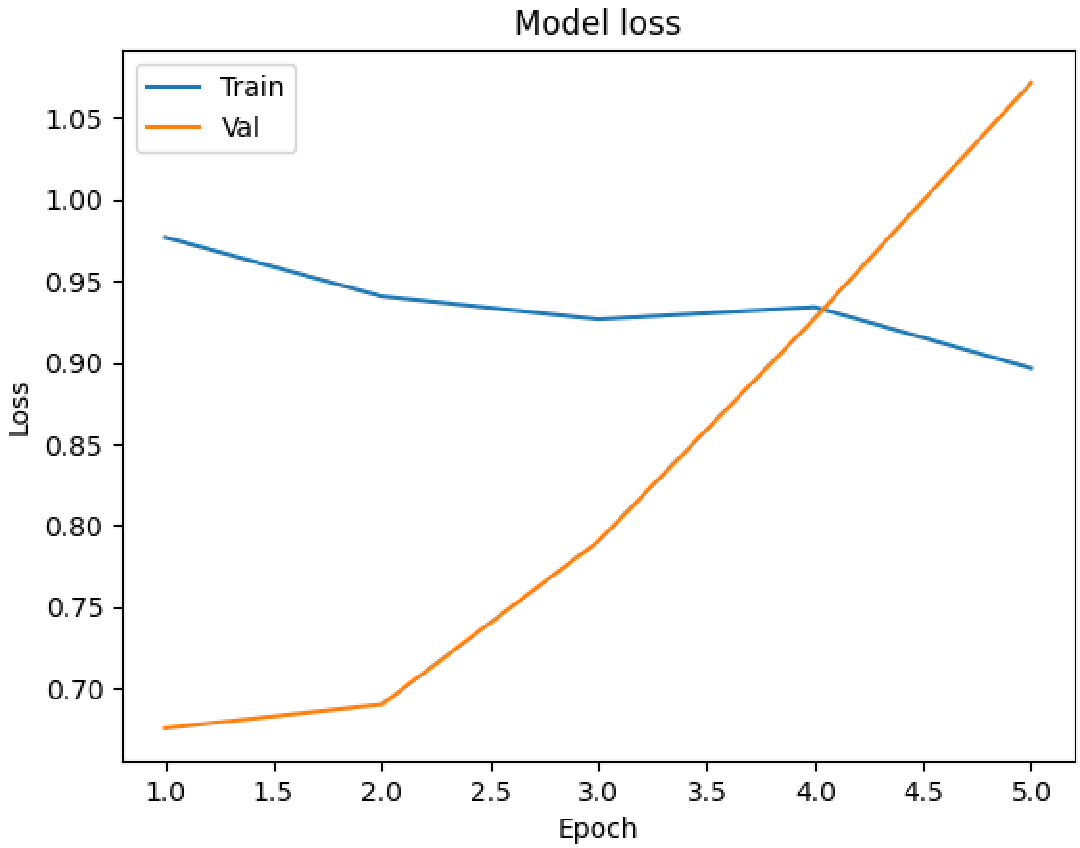


Image labelling on sample movie poster

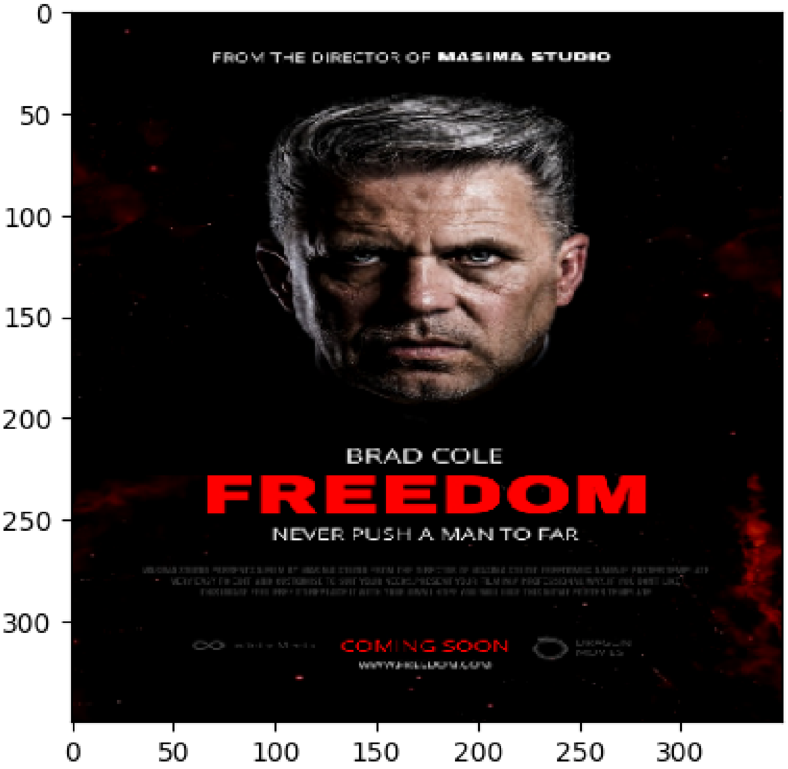
```
[ ]: img = image.load_img('free.jpg', target_size=(img_width, img_height, 3))
     plt.imshow(img)
     img = image.img_to_array(img)
     img = img/255.0

     img = img.reshape(1, img_width, img_height, 3)

     classes = data.columns[2:]
     print(classes)
     y_prob = model.predict(img)
     top3 = np.argsort(y_prob[0])[:-4:-1]

     for i in range(3):
       print(classes[top3[i]])
```

```
Index(['Action', 'Adventure', 'Animation', 'Biography', 'Comedy', 'Crime',
       'Documentary', 'Drama', 'Family', 'Fantasy', 'History', 'Horror',
       'Music', 'Musical', 'Mystery', 'N/A', 'News', 'Reality-TV', 'Romance',
       'Sci-Fi', 'Short', 'Sport', 'Thriller', 'War', 'Western'],
      dtype='object')
1/1 [==============================] - 1s 929ms/step
N/A
Western
Crime
```


FROM THE DIRECTOR OF MASIMA STUDIO
BRAD COLE
FREEDOM
NEVER PUSH A MAN TO FAR
COMING SOON
0
50
100
150
200
250
300
0
50
100
150
200
250
300