

Applied Deep Learning Final Project

Movie Genre Classification

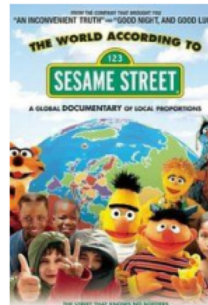
TEAM

Rafael Antonio Niño Vargas
 Santiago Nicolás Barco Gómez

Julio, 2018

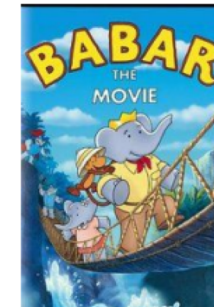
Input

The world according to sesame street



A documentary which examines the creation and co-production of the popular children's television program in three developing countries: Bangladesh, Kosovo and South Africa.

Babar: The movie



In his spectacular film debut, young Babar, King of the Elephants, must save his homeland from certain destruction by Rataxes and his band of invading rhinos.

Prediction

Comedy, Adventure,
 Family, Animation

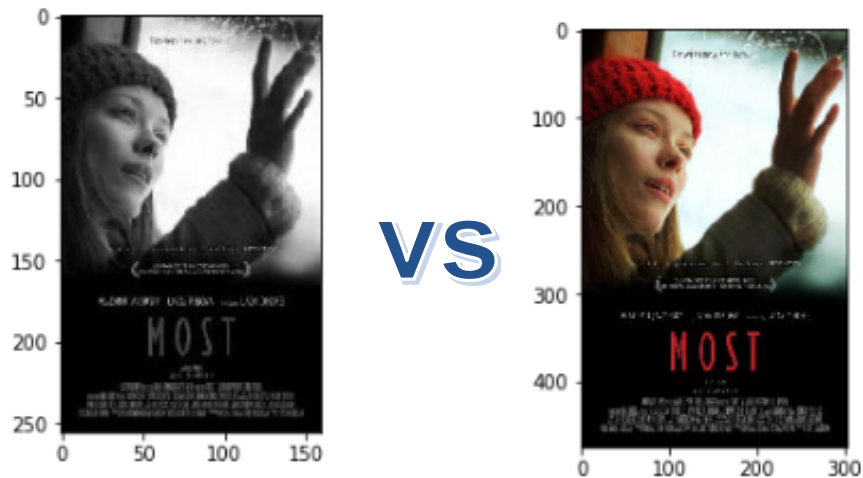
Documentary,
 History

Comedy, Adventure,
 Family, Animation

Adventure, War,
 Documentary, Music

Output:

1. Probability of the movie belong to each genre.



How?:

1. Use GPU, limited by the laptop RAM.
2. Use COLAB, limited by many users.

Issue:

1. Computer capacity CPU / GPU / RAM.
2. Machine Learning algorithm
3. Deep Learning algorithm

```
In [9]: from tensorflow.python.client import device_lib
print(device_lib.list_local_devices())

[name: "/device:CPU:0"
 device_type: "CPU"
 memory_limit: 268435456
 locality {
 }
 incarnation: 6378999212706407437
, name: "/device:GPU:0"
 device_type: "GPU"
 memory_limit: 1741082624
 locality {
   bus_id: 1
   links {
 }
 }
 incarnation: 15355355965038652028
 physical_device_desc: "device: 0, name: GeForce"]
```

```
incarnation: 6378999212706407437
, name: "/device:GPU:0"
 device_type: "GPU"
 memory_limit: 1741082624
```

How we do?

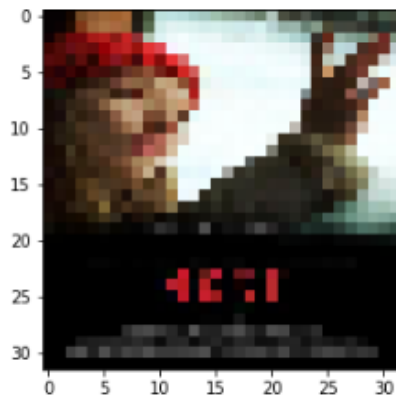
Color Image Reshape 32,32,3

```
In [72]: image.shape
```

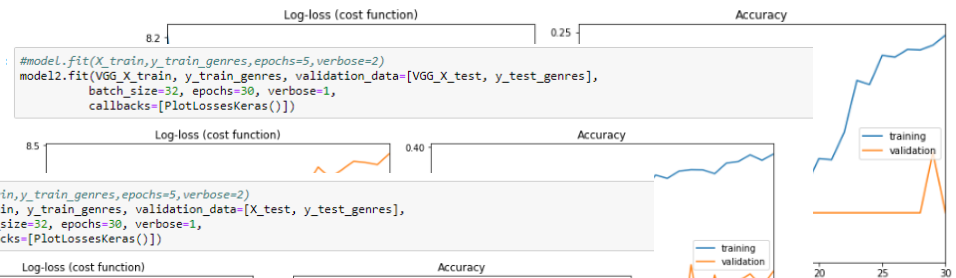
```
Out[72]: (32, 32, 3)
```

```
In [73]: fig, ax = plt.subplots(nrows=1, ncols=1)
ax.imshow(image, cmap='gray')
```

```
Out[73]: <matplotlib.image.AxesImage at 0x46ccd15668>
```



```
In [473]: #model.fit(X_train,y_train_genres,epochs=5,verbose=2)
model.fit(X_train, y_train_genres, validation_data=[X_test, y_test_genres],
          batch_size=32, epochs=30, verbose=1,
          callbacks=[PlotLossesKeras()])
```



CONCLUSIONES

```
: print('The accuracy of the model RF ML is ',RF)

print('The accuracy of the model KERAS is ',KERAS)

print('The accuracy of the model2 VGG is ',VGG)

print('The accuracy of the model3 CNN is ',CNN)
```

The accuracy of the model RF ML is 0.8893419033000768
The accuracy of the model KERAS is 0.8866557943207981
The accuracy of the model2 VGG is 0.8868956254796623
The accuracy of the model3 CNN is 0.8883665899206958

ML: Random Forest.
DL: Neuronal Network – Keras.
DL: VGG.
DL: Transfer Learning con CNN

Text Analytics

Deep Learning: Neuronal Network – Keras.

Machine Learning: Random Forest.

```
clf.fit(X_dtm, y_genres)
```

```
OneVsRestClassifier(estimator=RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',  
max_depth=10, max_features='auto', max_leaf_nodes=None,  
min_impurity_decrease=0.0, min_impurity_split=None,  
min_samples_leaf=1, min_samples_split=2,  
min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=-1,  
oob_score=False, random_state=42, verbose=0, warm_start=False),  
n_jobs=1)
```

Accuracy: 0.9370

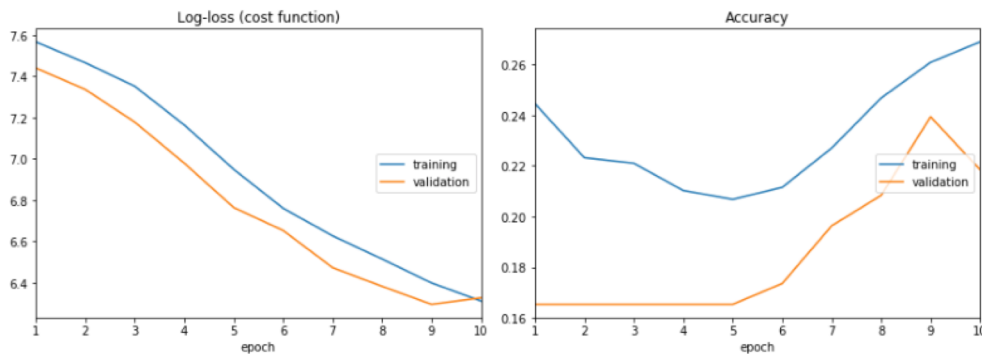
```
model.summary()
```

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 2100)	4202100
activation_1 (Activation)	(None, 2100)	0
batch_normalization_1 (Batch Normalization)	(None, 2100)	8400
dropout_1 (Dropout)	(None, 2100)	0
dense_2 (Dense)	(None, 24)	50424
activation_2 (Activation)	(None, 24)	0

Total params: 4,260,924
Trainable params: 4,256,724
Non-trainable params: 4,200

Accuracy: 0.4976

Deep Learning: Transfer Learning con word2vec



Accuracy: 0.2184

"THE BEST WAY TO
PREDICT YOUR FUTURE
IS TO CREATE IT."

- ABRAHAM LINCOLN

GRACIAS