

# ***PROGETTO ANALISI DI IMMAGINI E VIDEO***

Docenti:

Prof. Giuseppe Manco

Prof. Francesco Sergio Pisani

Studente:

Giorgio Simone

matricola 214575



# Introduzione

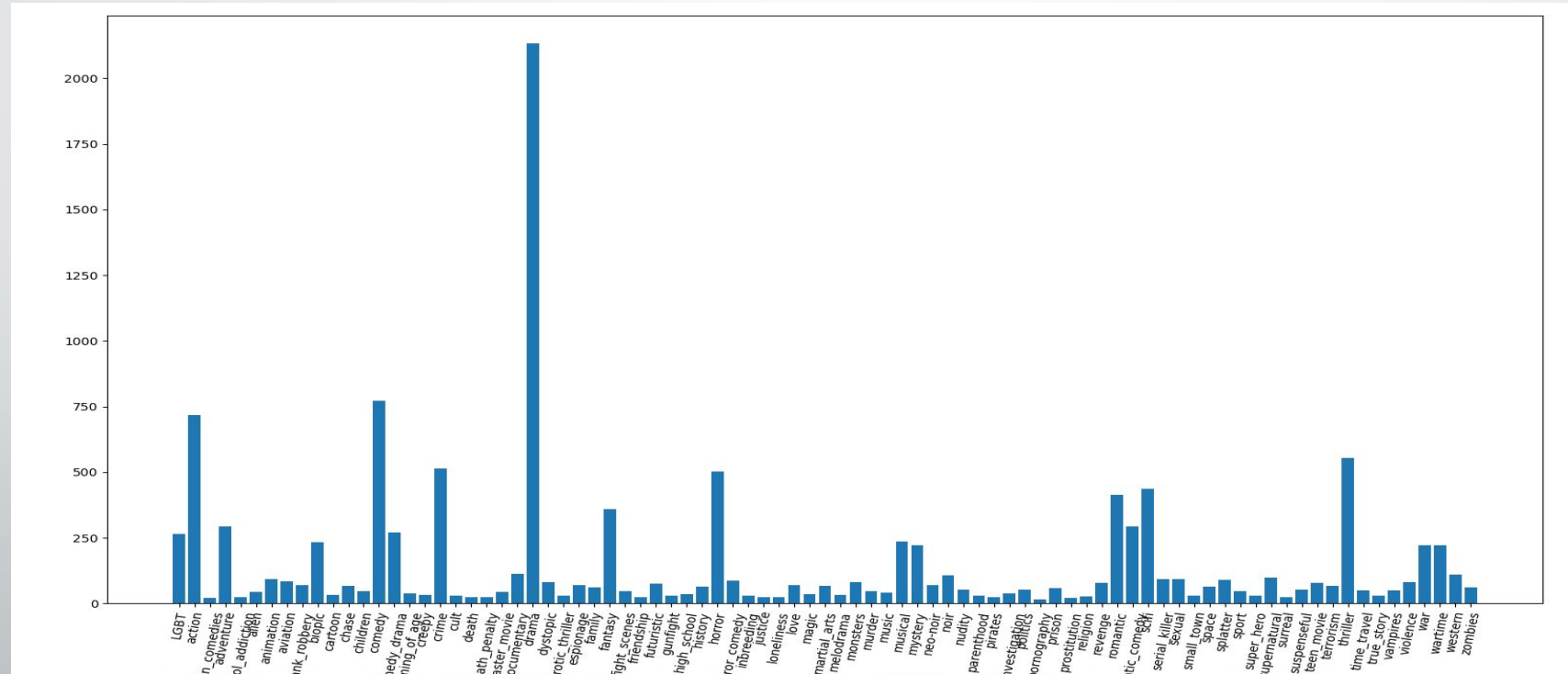
L'obiettivo del progetto è quello di creare un modello per la classificazione multi-classe e multi-label di trailer di film

Il dataset utilizzato contiene **5404** trailer di film, suddivisi in 4292 per il Training set e 1112 per il Test set

# Data Analysis

Ai trailer sono associati **85 tag**

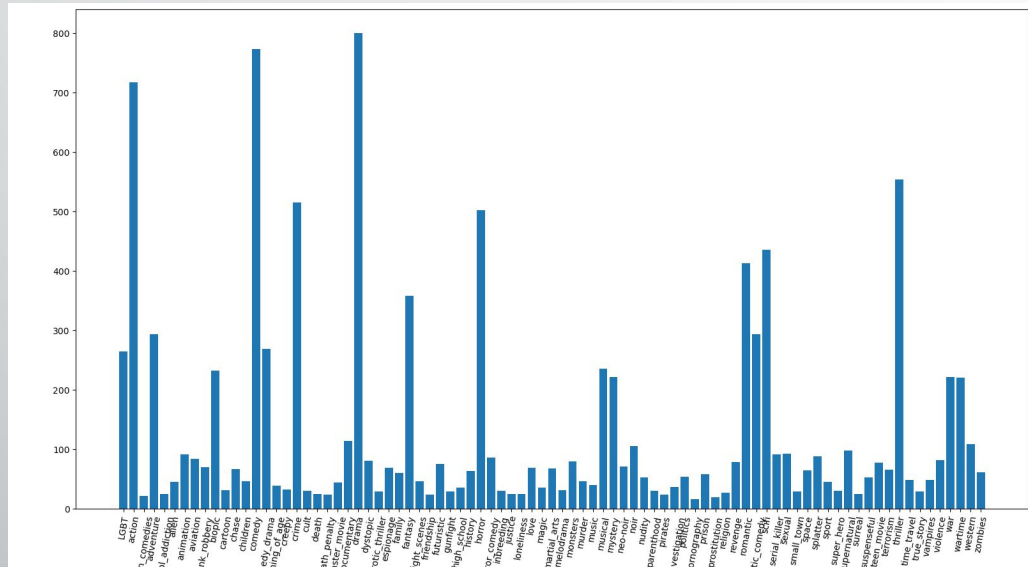
altamente sbilanciati che rendono più complessa l'analisi



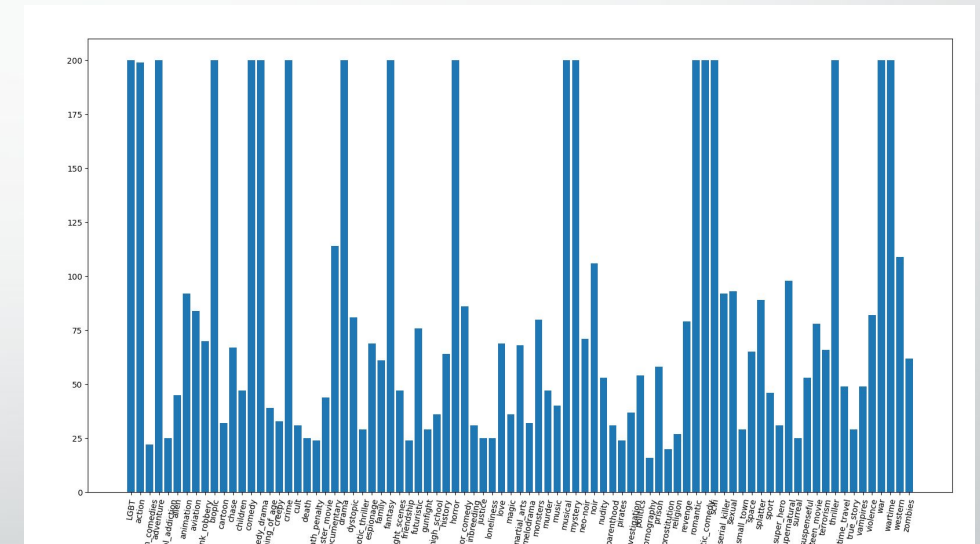
Per il bilanciamento dei dati sono state testate diverse soluzioni.

Soluzione migliore: undersampling su classe maggioritaria (drama)

best

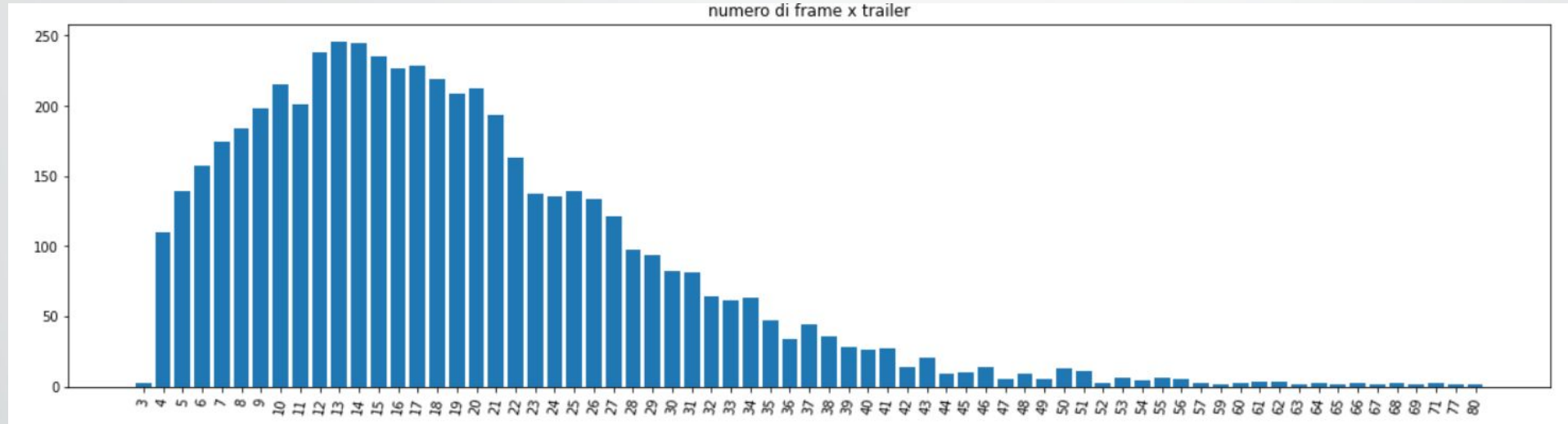


estremo



Sono state effettuate delle prove anche utilizzando una loss pesata sulle classi, con scarsi risultati  
 $w_j = n_{\text{samples}} / (n_{\text{classes}} * n_{\text{samples}}_j)$

A seguito di alcune analisi si è optato per utilizzare **15** frame per trailer



Sui frame è stato effettuato un CenterCrop(224)

Altre tecniche di data augmentation testate:

RandomRotation(degrees=(-20, 20)),  
RandomHorizontalFlip(p=0.5),  
Normalize([0.485, 0.456, 0.406],[0.229, 0.224, 0.225])) (rispetto ad Imagenet)

# Model

Dopo diverse simulazioni, si è optato per utilizzare:

- **VGG19** pre-addestrata su ImageNet, per feature extraction.
- Blocco di classificazione personalizzato formato da livelli densi

Le **15** feature map di dimensione **512x7x7** ottenute dalla rete convoluzionale, prima di essere passate al classificatore, sono state aggregate effettuandone la **media**.

## Reti testate:

- VGG16
- VGG19
- RESNET50
- RESNET152

```
self.classifier= nn.Sequential(  
    nn.Linear(25088 , 6656),  
    nn.ReLU(True),  
    nn.Dropout(),  
    nn.Linear(6656 , 1024),  
    nn.ReLU(True),  
    nn.Linear(1024 , 1024),  
    nn.ReLU(True),  
    nn.Dropout(),  
    nn.Linear(1024 , 85),  
    nn.Sigmoid()  
)
```

# Train

## Loss sul training set:

Tempo addestramento singola epoca:

11h30m

Problematica stop imprevisti dovuti a Google colab, le epoche «early stopped» sono state comunque tenute perché hanno superato le 8.30h di addestramento.

Parametri utilizzati:

- **Loss:** Binary Cross Entropy
- **Ottimizzatore:** Adam
- **Learning rate:** 0.0001  
testata anche 1e-5 parallelamente in diverse epoche

END Epoch 1/10 Train loss: 0.140153

END Epoch 2/10 Train loss: 0.128752

END Epoch 3/10 Train loss: 0.125942

END Epoch 4/10 Train loss: 0.117826

END Epoch 5/10 train loss: 0.123577 (early stopped)

END Epoch 6/10 Train loss: 0.113836 (early stopped)

END Epoch 7/10 Train loss: 0.105162 (early stopped)

END Epoch 8/10 Train loss: 0.120767

END Epoch 9/10 Train loss: 0.110056 (early stopped)

END Epoch 10/10 Train loss: 0.108614

## Test

Miglior risultato ottenuto dopo 10 epoche

A seguire i risultati del Classification report sui top10:

|              |      |      |      |      |
|--------------|------|------|------|------|
| micro avg    | 0.17 | 0.60 | 0.27 | 3147 |
| macro avg    | 0.04 | 0.17 | 0.06 | 3147 |
| weighted avg | 0.17 | 0.60 | 0.25 | 3147 |
| samples avg  | 0.17 | 0.65 | 0.26 | 3147 |

Micro-avg F1 **0.27** con **25** classi con f1-score > 0





**GRAZIE PER L'ATTENZIONE**