

PROGETTO ANALISI DI IMMAGINI E VIDEO

Docenti:

Giuseppe Manco
Francesco Sergio Pisani

Studenti:

Serafino Salatino 214455
Francesco Maria Granata 216648

Dataset

- Classi fortemente non bilanciate
- Numero di frame non uniforme

Uso del Dataset

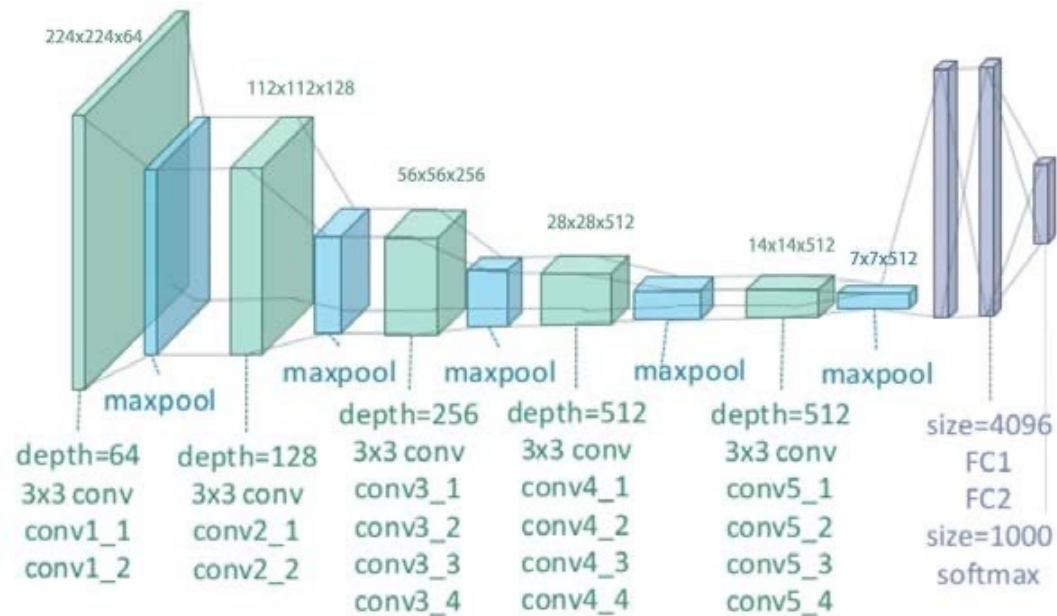
- Scelta di 20 frame per Trailer dettata da considerazioni sperimentali
- Scelta di una Loss non pesata sulle classi (spiegato nelle slide successive)

Modello

- Tecnica scelta per l'addestramento basata sul Transfer Learning
- Aggregazione di tutte le feature map dei frame ottenute dallo step precedente per generare le feature map del trailer
- Il livello denso di output si occuperà della predizione del trailer

Modello

- Il Transfer Learning è stato eseguito, utilizzando la rete VGG19



Modello

- Le immagini, prima di essere passate alla VGG19 sono state ridotte in dimensione 224 x 224.
- E' stato eliminato il livello denso di output e sostituito con un'altra rete convoluzionale 1d.
- Questa rete prende in input 20 canali (che sono i frame), ciascun canale è una sequenza di 1000 elementi restituita dalla VGG19.

Modello

Motivazioni sulla scelta della rete convoluzionale 1d:

Invece di utilizzare un metodo statico (pooling, somma, media, ecc..) per aggregare le feature dei singoli frame si fa in modo che la rete stessa apprenda come effettuare l'aggregazione durante la fase di training per ottenere le feature del trailer

Modello

La rete presenta:

- 3 livelli convoluzionali 1d con funzione relu e uso della batch normalization
- 1 livello di MaxPooling 1d
- 1 livello di Dropout
- 2 livelli convoluzioni 1d con funzione relu e uso della batch normalization
- 1 livello di MaxPooling 1d
- 1 livello di Dropout

Livello di classificazione che porta le 384 feature in input alle 85 di output (classi da predire)

```
Sequential(  
  (0): Conv1d(20, 20, kernel_size=(3,), stride=(2,), padding=(1,))  
  (1): ReLU()  
  (2): BatchNorm1d(20, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
  (3): Conv1d(20, 32, kernel_size=(5,), stride=(2,), padding=(1,))  
  (4): ReLU()  
  (5): BatchNorm1d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
  (6): Conv1d(32, 64, kernel_size=(5,), stride=(2,), padding=(1,))  
  (7): ReLU()  
  (8): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
  (9): MaxPool1d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
  (10): Dropout(p=0.5, inplace=False)  
  (11): Conv1d(64, 64, kernel_size=(7,), stride=(2,), padding=(1,))  
  (12): ReLU()  
  (13): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
  (14): Conv1d(64, 64, kernel_size=(7,), stride=(2,), padding=(1,))  
  (15): ReLU()  
  (16): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
  (17): MaxPool1d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
  (18): Dropout(p=0.5, inplace=False)  
  (19): Flatten(start_dim=1, end_dim=-1)  
  (20): Linear(in_features=384, out_features=85, bias=True)  
  (21): Sigmoid()  
)
```

Modello

- Loss utilizzata: Binary Cross Entropy.
- E' stata implementata anche la Focal Loss ma con scarsi risultati
- Ottimizzatore scelto: Adam
- Learning rate pari a 0.0001

Risultati

Loss sul training set:

```
Start epoch 33
epoch 33/50 train loss: 0.111085 Labeled
epoch 33/50 train loss: 0.109665 Labeled
epoch 33/50 train loss: 0.109351 Labeled
epoch 33/50 train loss: 0.110608 Labeled
epoch 33/50 train loss: 0.110490 Labeled
epoch 33/50 train loss: 0.110672 Labeled
epoch 33/50 train loss: 0.110847 Labeled
epoch 33/50 train loss: 0.110519 Labeled
epoch 33/50 train loss: 0.110649 Labeled
epoch 33/50 train loss: 0.111004 Labeled
epoch 33/50 train loss: 0.111312 Labeled
epoch 33/50 train loss: 0.111516 Labeled
epoch 33/50 train loss: 0.111536 Labeled
END Epoch 33/50 Train loss: 0.111422
Start epoch 34
```

Risultati

Classification Report sul test set

micro avg	0.17	0.61	0.27	3148
macro avg	0.05	0.19	0.07	3148
weighted avg	0.17	0.61	0.26	3148
samples avg	0.17	0.65	0.26	3148