

Esercitazione 8

Progetto

Progetto

- Classificazione multi-classe e multi-label di trailer
- Task: predire le classi associate ad ogni film
- Costruire un modello di classificazione

Dataset

- Il progetto è distribuito in 4 file:
 - progetto_2021_dataset_labeled.tgz - **TRAIN+TEST**
 - progetto_2021_dataset_UNlabeled1.tgz
 - progetto_2021_dataset_UNlabeled2.tgz
 - progetto_2021_dataset_UNlabeled3.tgz
- Il dataset è composto da 5404 trailer. Il dataset è suddiviso in train (4292) e test (1112)
- Sono definiti 85 tag con numerosità differente
- Download: link nel file .md allegato alle slide

Dataset

- Nel file *progetto_2021_dataset_labeled.tgz* sono memorizzate delle folder con nome «ttXXXXXXXXXX» e un file json «train_test_split_dict.json»
 - Ogni folder contiene i keyframe estratti dal trailer, non tutti i trailer hanno lo stesso numero di frame
 - Il file json contiene un dizionario con due chiavi *train* e *test*: ad ognuna è associato un dizionario con chiavi il nome della folder del trailer e con valore la lista dei tag associati al trailer. Non tutti i trailer hanno lo stesso numero di tag
 - Le folder del test non devono essere usate per il train

Esempio:

```
'tt4060866': ['crime', 'drama', 'action']
```

Dataset

- Nei file `progetto_2021_dataset_UNlabeled{1,2,3}.tgz` sono memorizzate delle folder con nome «ttXXXXXXXXXX» e ognuna contiene i keyframe dei trailer
- Sono dati non etichettati, quindi non sono associate delle etichette
- Possono essere utilizzati per migliorare gli algoritmi di estrazione delle feature, ed esempio tramite approcci di self supervision

Dataset

- I file hanno dimensione
 - progetto_2021_dataset_labeled.tgz 6,6GB
 - progetto_2021_dataset_UNlabeled1.tgz 5,7GB
 - progetto_2021_dataset_UNlabeled2.tgz 5,6GB
 - progetto_2021_dataset_UNlabeled3.tgz. 5,6GB

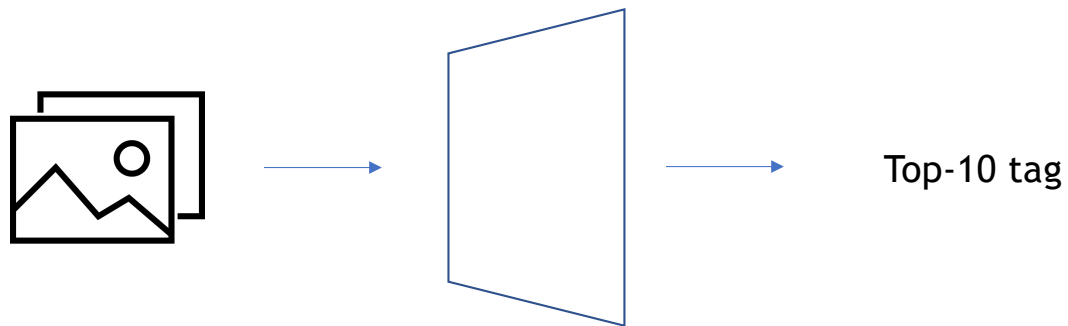
Progetto

- 5k video con etichetta (4k/1k training/test, 85 etichette)
 - Keyframe -> 360x240
- 15k video senza etichetta
- Protocollo:
 - Train on 4k (+ unlabeled data)
 - Test on 1k test dataset, predizione delle top-10 classi
- Valutazione tramite micro-precision, micro-recall e micro-F1 score

Suggerimenti

Un batch è formato da k trailer, per ogni trailer si ha un numero massimo di frame da valutare (es. 30)

Un modello può essere utilizzato per estrarre delle features da ogni frame, si combinano le FM di tutti i frame del trailer e su queste si calcola la classificazione



Suggerimenti

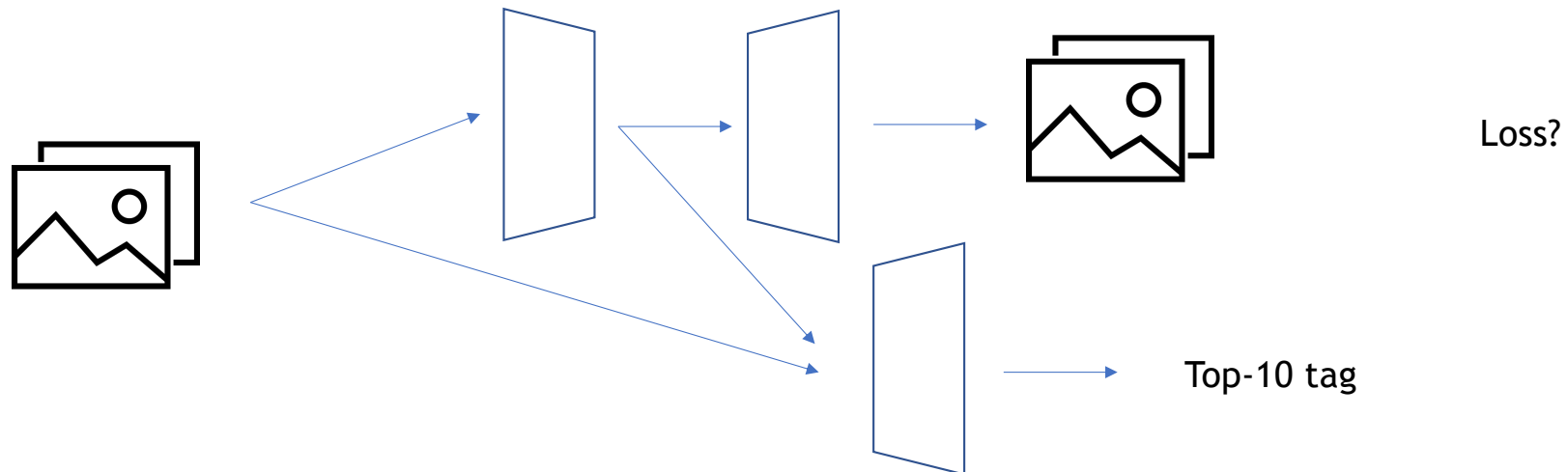
- Per ogni frame una FM
 - Rete pre-addestrata (vgg, resnet, ecc.)?
- Come combinare le FM
 - Somma
 - Media
 - Pooling
 - ???
- Come costruire il classificatore?
- Colab/Azure/ecc. con salvataggio dello stato
- Test veloci su porzioni del dataset

Suggerimenti

Come si possono sfruttare le immagini unlabeled?

Un modello può essere utilizzato per estrarre le features dalle immagini ed uno per calcolare la classificazione

Durante il train alternare un batch labeled e uno unlabeled



Suggerimenti

- Analisi del dataset
 - Quali frame selezionare? Tutti? Una parte?
 - Applicare delle trasformazioni? Normalizzazione?
 - Aggiunta di operazioni di preprocessing? Trasformazione dei frame?
 - Risoluzione dei frame?
- Scelta del modello
 - Quale scegliere?
 - Estrazione delle feature con reti pre-trained?
 - Suddividere i due task (estrazione delle feature e classificazione)?
 - Quali architetture? Combinare più modelli?
- Analisi degli errori
 - Su quali tag il modello commette più errori? Perché?
 - Feature ottimizzate per le classi di minoranza
 - Classi sbilanciate!!!
- Tuning dei modelli
 - Analisi della loss

Requisiti

- Definire e addestrare un modello di classificazione
- Descrivere in una presentazione le scelte progettuali e tutti i parametri utilizzati nella sperimentazione.
- Consegnare notebook (e/o file sorgente) utilizzati per il training, il notebook di test, presentazione e dump del modello
- Il notebook di test deve caricare il modello e processare il test set
- Info:
 - https://pytorch.org/tutorials/beginner/saving_loading_models.html

Protocollo di valutazione

- Valutazione generale del progetto
- Loading del modello e calcolo delle metriche micro-precision, micro-recall e micro-f1 sul test set considerando le Top-10 predizioni

Baseline-1 - 8 pt - Valore micro-F1 0.20 con micro-precision ≥ 0.10 e micro-recall ≥ 0.45

Baseline-2 - 7 pt - Valore micro-F1 0.25 con micro-precision ≥ 0.15 e micro-recall ≥ 0.50

Per accedere all'orale sono necessari almeno 8 pt.

Assegnazione dei punteggi (micro-F1):

- Se il modello supera la Baseline-1: 8 pt
- Se il modello non supera la Baseline-2: $(p - \text{Baseline-1}) / (\text{Baseline-2} - \text{Baseline-1}) * 7$ pt
- Se il modello supera la Baseline-2: 7 pt + bonus
- Bonus $(p - \text{Baseline-2}) / (\text{BESTMODEL} - \text{Baseline-2}) * 5$ pt

Protocollo di valutazione

y_true = etichette vere

Y_pred = Top-10 predizioni del modello

`sklearn.metrics.classification_report(y_true, y_pred)`

precision	recall	f1-score	support		
	action	0.22	0.70	0.34	191
	action_comedies	0.10	0.33	0.15	6
	adventure	0.09	0.56	0.16	75
	alcohol_addiction	0.00	0.00	0.00	6
	alien	0.00	0.00	0.00	12
	animation	0.08	0.43	0.14	21
	...				
	war	0.11	0.28	0.16	58
	wartime	0.08	0.31	0.12	49
	western	0.05	0.24	0.08	33
	zombies	0.11	0.27	0.16	15
	micro avg	0.14	0.48	0.21	3167
	macro avg	0.05	0.18	0.08	3167
	weighted avg	0.17	0.48	0.25	3167
	samples avg	0.14	0.52	0.21	3167

For x in Tag1 ... TagN

True positive (TPx)

False positive (FPx)

False negative (FNx)

$$\text{Micro-Prec} = \frac{\sum TPx}{\sum TPx + FPx}$$

$$\text{Micro-Prec} = \frac{\sum TPx}{\sum TPx + FNx}$$

$$F1 = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$$

Appelli

- XX - luglio - Progetto attuale
 - Consegna entro «4gg prima dell'esame»
- XX - luglio - Nuovo Progetto (stesse modalità)
 - Rilascio indicativo fine giugno