UNIVERSITÀ DELLA CALABRIA
DIPARTIMENTO DI
INGEGNERIA INFORMATICA,
MODELLISTICA, ELETTRONICA
E SISTEMISTICA
DIMES

Corso di Laurea Magistrale in

Ingegneria Informatica

Progetto Analisi di Social Network e Media

Approccio multimodale per la classificazione di meme

Candidati

Amirato Simone matr. 235520

D'Atri Fulvio matr. 235344

De Luca Francesco matr. 235300

Anno Accademico 2022/2023

Sommario

1 Introduzione

Lo scopo del presente progetto è quello di realizzare un modello di Deep Learning per la classificazione di meme che faccia uso di un approccio multimodale. Il dataset, costituito da 9000 campioni (8500 per il train set e 500 per il test set), è stato rilasciato da Facebook :con lo scopo di indire una challenge; ogni partecipante ha sviluppato ed addestrato un modello capace di classificare i meme in "Hateful" e "Not-Hateful"; alla fine della challenge, Facebook ha presentato :una classifica basata sulla bontà degli score di ogni modello.

Il dataset è costituito da due attributi e da una etichetta di classe; nello specifico, i due attributi rappresentano l'immagine e la trascrizione del testo nella stessa. L'etichetta assume valore 0 se il campione è "not-hateful", 1 altrimenti.

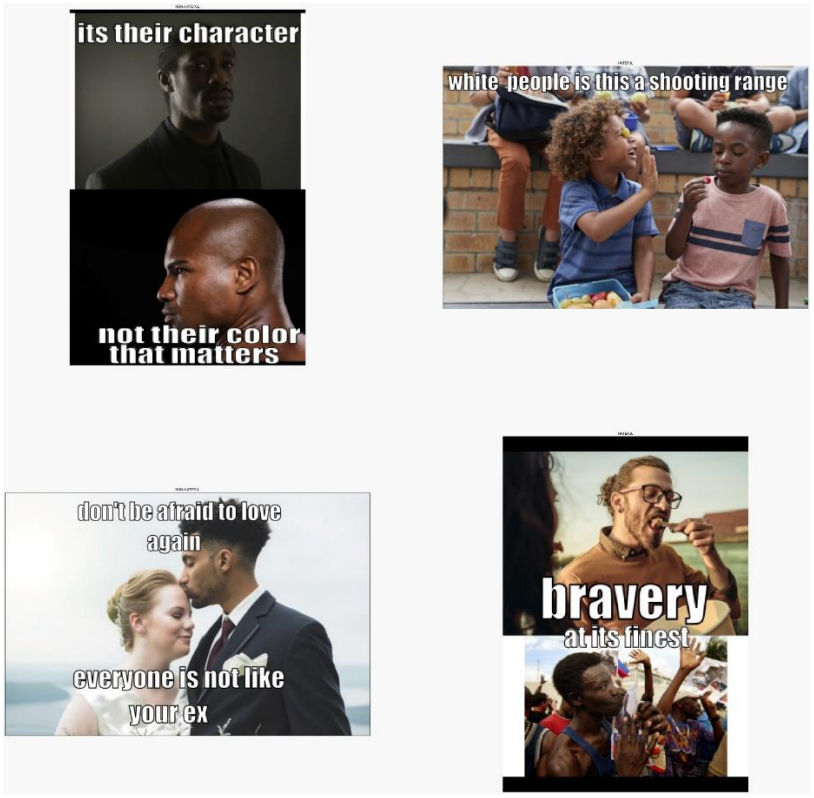

Figure 1: Esempio di meme non-hatefull (sinistra) e hatefull (destra)

2 Sviluppo

In fase di progettazione è stato definito lo schema del modello da sviluppare; l'idea è stata quella di utilizzare due modelli preaddestrati per la feature extraction, uno per le immagini ed uno per il testo, e di passare in input i loro output ad una rete fully connected per la classificazione.

Sono stati sviluppati più modelli, allo scopo di confrontare le prestazioni e scegliere quello che offrisse le performance migliori; ad ogni modo, bisogna sottolineare che i primi cinque classificati, nella challenge di Facebook, hanno raggiunto una accuratezza media di circa 0.8 "https://github.com/drivendataorg/hateful-memes/tree/main" .

2.1 Preprocessing sui dati

Da una prima analisi sui dati si è notato che il training set, costituito da 8500 istanze, fosse sbilanciato (5450 istanze "not-hateful" e 3050 "hateful"). Le immagini sono in formato RGB, ma con size molto diverse tra loro.

La prima operazione svolta sul training set è stata quella di downsampling al fine di bilanciare le due classi. Successivamente, come suggerito nella presentazione relativa al progetto vincitore del contest, è stata eseguita la rimozione del testo dalle immagini al fine di ridurre possibili elementi di distrazione per la fase di image feature extraction.

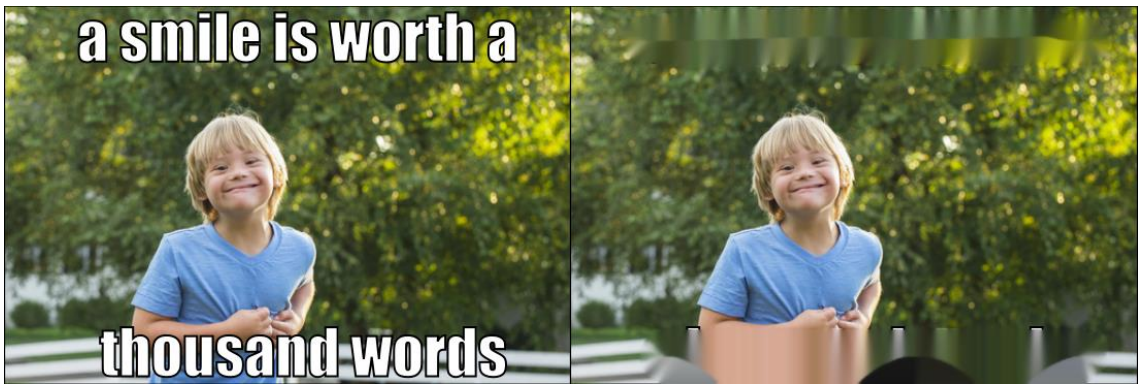

Figure 2: Esempio di rimozione del testo da un meme.

In fase di caricamento del dataset delle immagini, sono state rese omogenee le size, ovvero sono state tutte ridimensionate a 224x224.

Sul testo non sono state eseguite operazioni, al fine di non manomettere la natura retorica ed i significati multipli delle varie affermazioni.

2.2 Prima implementazione

Nel primo modello implementato è stato utilizzato il Vision Trasformer pre-addestrato *dinov2-base* per l'image feature extraction ed il Sentence Trasformer *sentence-t5-large* per il text embedding. La parte di fusion è stata realizzata attraverso layer densi che ricevevano in input la concatenazione dei due embedding, ognuno dei quali con size pari a 768. Il modello, dopo un training di 50 epoche ed un valore di loss raggiunto pari a 0.021, ha restituito risultati prossimi a quelli di un classificatore dummy.

2.3 Seconda implementazione

Nella seconda implementazione, sono stati sostituiti i modelli pre-addestrati con architetture più performanti. Nello specifico, è stato usato il *google/vit-base-patch16-224-in21k* per la gestione delle immagini, mentre per il text embedding *all-mpnet-base-v2*; la size degli embedding rimane di 768. Non avendo ottenuto cambiamenti significativi negli score, abbiamo deciso di modificare completamente architettura, passando ad una implementazione basata su CLIP.

2.4 Terza implementazione: Modello CLIP

Un modello CLIP è un tipo di rete neurale artificiale che combina immagini e testo per comprenderli insieme. Questo modello è addestrato per rappresentare concetti visivi e linguistici in uno spazio comune, al fine di effettuare ricerche e comprensione attraverso entrambi i tipi di dati, consentendo di cercare immagini

basate su descrizioni testuali e viceversa. Il modello realizzato approssima la struttura presente nel repository https://github.com/gokulkarthik/hateclipper .

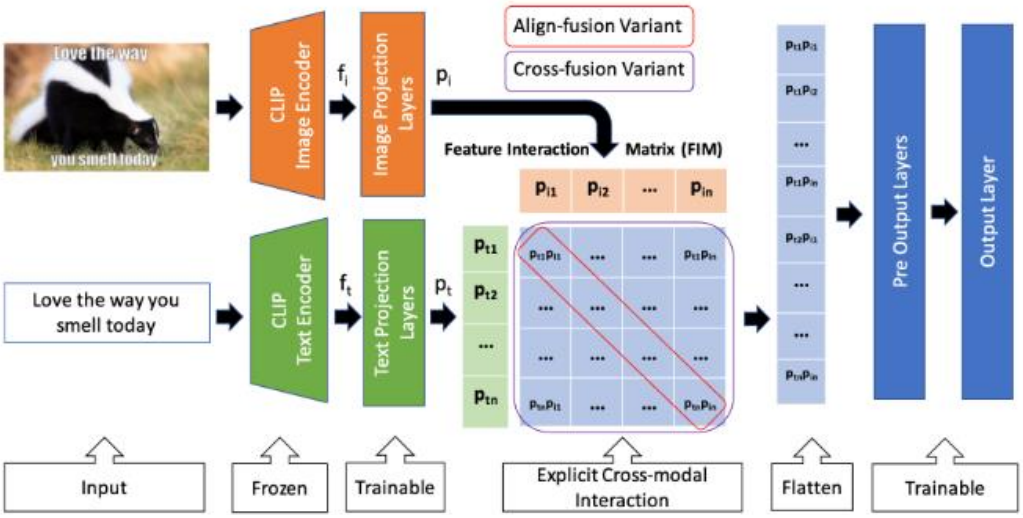

Figure 3: Struttura della rete CLIP implementata.

I modelli utilizzati in fase di encoding sono quelli presenti nel modello CLIP pre-addestrato *openai/clip-vit-large-patch14*, il quale utilizza l'architettura Transformer *ViT-L/14* per l'elaborazione delle immagini ed un Transformer con self attention mascherata per il testo.

Sono state addestrate e testate varie versioni di tale modello, ognuna delle quali presentava un specifico approccio per la fusion dei due embedding; più in dettaglio, sono stati testate le seguenti varianti:

- **Concatenation-fusion Variant**: effettua la concatenazione dei due embedding.
- **Align-fusion Variant**: effettua il prodotto elemento per elemento dei due embedding.
- **Cross-fusion Variant**: effettua il prodotto matriciale dei due embedding e restituisce la matrice risultato attraverso un'operazione di flattening.

In fase di valutazione, il modello basato su **cross-fusion** ha restituito i risultati più alti e dunque è stato selezionato come modello finale.

Nelle figure sottostanti sono mostrati i risultati ottenuti da questo modello. È importante sottolineare che a partire dalla ventesima epoca è stato utilizzato un learning rate variabile con parametri *gamma* pari a 0.95 e *step* pari ad 1.

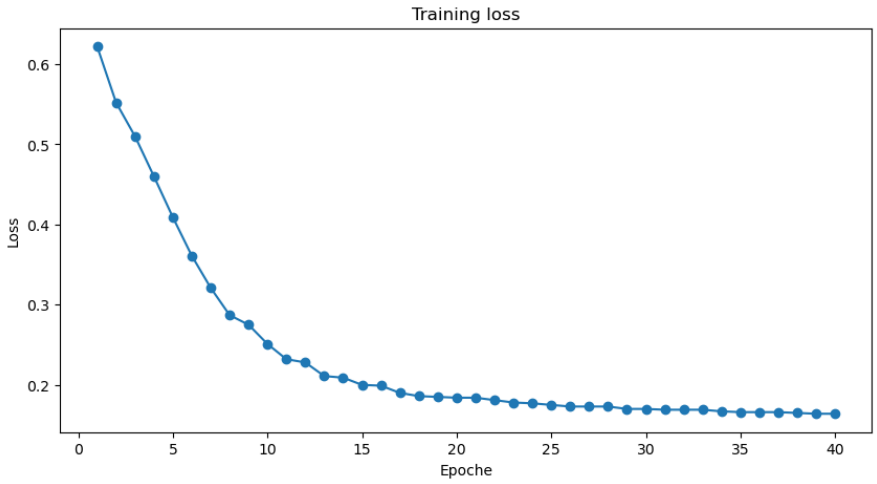

Figure 4: Valori della loss al variare del numero di epoche.

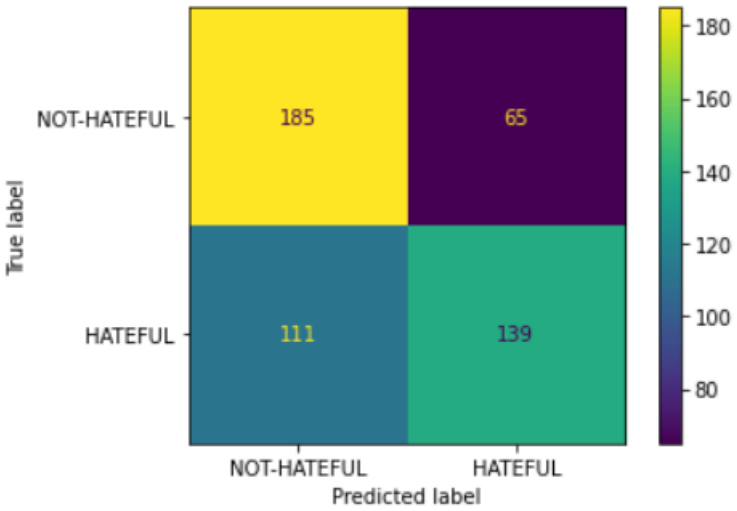

Figure 5: Confusion Matrix per il modello CLIP Cross-fusion.

	precision	recall	f1-score	support
0	0.62	0.74	0.68	250
1	0.68	0.56	0.61	250
accuracy			0.65	500
macro avg	0.65	0.65	0.64	500
weighted avg	0.65	0.65	0.64	500

Figure 6: Score per il modello CLIP Cross-fusion.

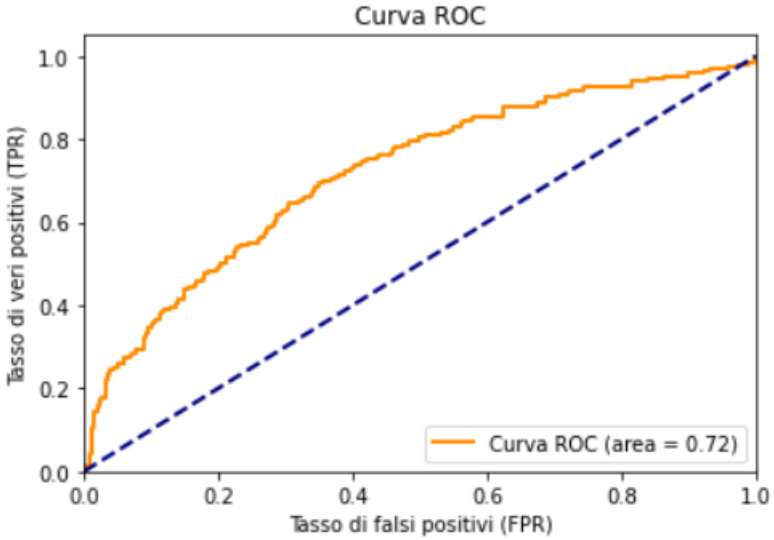

Figure 7: Curva ROC per il modello CLIP Cross-fusion.

3. Conclusioni

In quest'ultimo capitolo viene presentato un confronto tra i diversi modelli realizzati, sulle principali misure di valutazione per il task di classificazione.

Models	**N° params PVM+PLM**	**Fusion**	**Accuracy**	**Precision**	**Recall**	**F1**	**AUC**
sentence-t5-large	335M	-	0.55	0.59	0.55	0.51	0.60
dinov2-base **+** ***sentence-t5-large***	86.6M+335M	concat	0.56	0.57	0.56	0.54	0.59
google/vit-base-patch16-224-in21k **+** ***all-mpnet-base-v2***	86.6M+110M	concat	0.56	0.57	0.56	0.56	0.60
CLIP	428M	concat	0.56	0.56	0.55	0.55	0.60
CLIP	428M	align	0.57	0.57	0.55	0.56	0.62
CLIP	428M	cross	**0.65**	**0.65**	**0.65**	**0.65**	**0.72**

Un possibile sviluppo futuro del modello è dato dall'integrazione di una *Graph Neural Network* (*GNN*) a valle del processo di fusione degli embedding, con lo scopo di valutare, attraverso una rappresentazione a grafo, la relazione di similarità tra gli embedding e quindi classificare il meme corrente sulla base delle classificazioni dei meme simili ad esso (come avviene nel modello *MERLIN* "https://arxiv.org/ftp/arxiv/papers/2302/2302.01676.pdf").

In conclusione, nonostante le performance attuali del nostro modello siano al di sotto degli standard trovati in letteratura, ciò non dovrebbe scoraggiare ulteriori sforzi di ricerca e sviluppo. L'analisi di meme e contenuti online richiede una comprensione profonda del linguaggio naturale e del contesto culturale, e

migliorare la capacità di identificare contenuti hateful è un obiettivo cruciale per promuovere un ambiente online più sicuro e inclusivo. Le future ricerche dovrebbero concentrarsi sulla raccolta di dati più ampi e di alta qualità, oltre che sull'arricchimento del meme in oggetto con informazioni di contesto: commenti e feedback da parte degli utenti fornirebbero degli incrementi notevoli nella classificazione finale del meme.