POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea Magistrale

Image-to-recipe, classificazione del cibo a partire da immagini

Relatori

Prof. Maurizio MORISIO

Prof. Andrea BOTTINO

Candidato

Matteo MAGNALDI

Ottobre 2023

Sommario

Il cibo riveste un ruolo indispensabile per la sopravvivenza umana. Oltre a fornirci energia, esso è anche una parte essenziale della nostra identità e cultura. Nel corso del tempo però i modelli alimentari e la cultura della cucina stanno evolvendo. In passato, il cibo veniva principalmente preparato in casa, ma oggi spesso ci affidiamo ad altri, come take-away, fast food e ristoranti. Di conseguenza, l'accesso a informazioni dettagliate sui cibi consumati è limitato, rendendo difficile conoscere esattamente cosa stiamo mangiando. In molti contesti, come la prevenzione di malattie, il fitness, il mantenimento di una dieta bilanciata e la corretta gestione di allergie e intolleranze specifiche, risulta molto importante monitorare quello che una persona mangia, in particolare identificare i diversi ingredienti che compongono un piatto e le loro quantità, grazie alle quali, è possibile calcolare i valori nutrizionali (grassi, proteine, carboidrati, ecc.) presenti in qualsiasi cibo consumato. Un modo pratico per effettuare queste analisi sarebbe scattare una foto del piatto che si intende consumare, attraverso uno smartphone e utilizzare un classificatore automatico in grado di riconoscere il cibo e gli ingredienti di cui è composto, con elevata precisione. Lo scopo della presente tesi di laurea è quindi quello di realizzare un classificatore che, ricevuto come input l'immagine di un cibo, sia in grado di riconoscerlo e identificare gli ingredienti che lo compongono. Per fare ciò, viene analizzata e studiata una delle più grandi collezioni pubblicamente disponibili di dati sulle ricette, Recipe1M[1], che nella sua ultima versione, Recipe1M+[2], contiene oltre un milione di ricette culinarie e 13 milioni di immagini di cibo. Sulla base di questo dataset, viene quindi realizzata una collezione di ricette più incentrata sui piatti tipici della tradizione italiana, che viene utilizzata come base per gli esperimenti. Inoltre vengono analizzate diverse tecniche di apprendimento automatico/deep learning e i relativi modelli, con lo scopo di ottenere un classificatore in grado di riconoscere con buona precisione il cibo rappresentato nell'immagine e gli ingredienti da cui è composto. I modelli analizzati in particolare sono:

- Un modello neurale multimodale, in grado di apprendere congiuntamente la rappresentazione delle ricette e delle relative immagini in uno spazio comune, con l'aggiunta di un high-level classification task, per la regolarizzazione semantica.[2];

- Un sistema che genera una ricetta culinaria, contenente un titolo, gli ingredienti e le istruzioni di cottura direttamente dall'immagine.[3];
- Un modello multimodale, implementato tramite Transformer gerarchici che supera significativamente i modelli basati su LSTM, rappresentando lo state of the art per i modelli di recupero delle ricette.[4];

Ringraziamenti

Tabella dei contenuti

Elenco delle tabelle VIII

Elenco delle figure IX

Acronyms XVIII

1 Introduzione 1
- 1.1 Analisi del problema . . . 1
- 1.2 Obiettivi della tesi . . . 2
- 1.3 Contributo della tesi . . . 2

2 State of the Art 4
- 2.1 Recipe1M . . . 4
- 2.2 Img2recipe . . . 7
  - 2.2.1 Modello . . . 7
  - 2.2.2 Metriche . . . 10
  - 2.2.3 Esperimenti e risultati . . . 11
- 2.3 Inverse Cooking . . . 12
  - 2.3.1 Modello . . . 14
  - 2.3.2 Metriche . . . 16
  - 2.3.3 Esperimenti e risultati . . . 18
- 2.4 Revamping Cross-Modal Recipe Retrieval with Hierarchical Transformers and Self-supervised Learning . . . 19
  - 2.4.1 Modello . . . 20
  - 2.4.2 Metriche . . . 24
  - 2.4.3 Esperimenti e risultati . . . 25

3 Soluzioni proposte 29
- 3.1 Approccio 0 . . . 29
  - 3.1.1 Implementazione . . . 29
  - 3.1.2 Esperimenti e Risultati . . . 30

3.2 Approccio 1 31
3.2.1 Nuovo dataset 31
3.2.2 Esperimenti con img2recipe 34
3.2.3 Esperimenti con inverseCooking 43
3.2.4 Principali problemi 51
3.3 Approccio 2 52
3.3.1 Idea e Implementazione 52
3.3.2 Metriche ed Esperimenti 53
3.3.3 Risultati 53
3.4 Approccio 3 58
3.4.1 Idea e implementazione 58
3.4.2 Metriche 59
3.4.3 Risultati 59
3.5 Approccio 4 64
3.5.1 Nuova versione del dataset 64
3.5.2 Esperimenti con img2recipe 68
3.5.3 Esperimenti con inverseCooking 74
3.6 Approccio 5 79
3.6.1 Implementazione 79
3.6.2 Esperimenti e metriche 80
3.6.3 Risultati 81

4 Conclusioni 106

A Categorie Food-101 108

Bibliografia 113

Elenco delle tabelle

2.1 Tabella di confronto tra i diversi modelli analizzati 28

3.1 Risultati degli esperimenti sul modello img2recipe 43
3.2 Risultati degli esperimenti sul modello img2recipe escludendo le istruzioni dalla composizione delle ricette 58
3.3 Risultati degli esperimenti sul modello img2recipe sulle ricette italiane tradotte in lingua inglese, partendo dal checkpoint fornito dagli autori 64
3.4 Risultati degli esperimenti sul modello img2recipe addestrato sulla nuova versione del dataset delle ricette italiane 74
3.5 Risultati degli esperimenti $Model_{15\_}rec_{v1}$ e $Model_{100\_}rec_{v1}$ 102
3.6 Risultati degli esperimenti $Model_{125\_}recs_{v2}$ e $Model_{R50\_}rec_{v2}$ . . . 103
3.7 Tabella riassuntiva esperimenti sulla prima versione del dataset delle ricette italiane 104
3.8 Tabella riassuntiva esperimenti sulla seconda versione del dataset delle ricette italiane 104
3.9 Tabella riassuntiva dei modelli utilizzati 105

Elenco delle figure

2.1 Statistiche Recipe1M. Distribuzione delle ricette in termini di portate.[1] 5
2.2 Statistiche Recipe1M. Partizionamento delle ricette.[1] 5
2.3 Statistiche Recipe1M. Distribuzione del numero di ingredienti e numero di istruzioni per ricetta.[1] 6
2.4 Statistiche Recipe1M+. Distribuzione del numero di immagini per ricetta.[2] 7
2.5 Statistiche Recipe1M+. Confronto tra il partizionamento di Recipe1M e Recipe1M+.[2] 7
2.6 Estrazione delle informazioni dalle ricette.[2] 10
2.7 Rappresentazione del modello Img2recipe di Recipe1M.[2] 10
2.8 Modello di Skip-instructions.[2] 11
2.9 Confronto tra architettura VGG-16 e Resnet-50.[2] 12
2.10 Confronto del modello introdotto rispetto al modello CCA.[1] . . . 12
2.11 Confronto tra modelli addestrati su Recipe1M e Recipe1M+. Con Recipe1M in questo caso si fa riferimento alle ricette e immagini comuni tra le due versioni del dataset.[2] 13
2.12 Risultati del processo di image-to-recipe. Sono riportati, da sinistra a destra, le immagini di input, i reali ingredienti della ricetta, gli ingredienti e l'immagine restituiti dal processo.[2] 13
2.13 Rappresentazione del decoder utilizzato per la generazione degli ingredienti relativi ad un'immagine di input.[3] 17
2.14 Encoder delle istruzioni. Sulla sinistra (a) viene mostrato il modello del trasformatore utilizzato, mentre nella parte di destra (b, c, d) sono riportate le strategie di fusione adottate.[3] 17
2.15 Architettura del modello InverseCooking.[3] 18
2.16 Confronto tra i migliori modelli proposti dagli autori e il modello R_{I2LR}. R_{I2L} indica una versione alternativa del modello che considera solamente gli ingredienti, tralasciando il titolo e le istruzioni.[3] . . . 19

2.17 Confronto tra i diversi modelli proposti in termini di cardinality prediction error. Sulla destra viene anche riportata la media del numero di ingredienti predetti.[3] 19
2.18 Confronto tra i diversi modelli proposti in termini di IoU e F1.[3] . 20
2.19 Confronto delle prestazioni dei diversi modelli in termini di precisione su diversi valori di K,.., e F1 per i singoli ingredienti, ordinati in base al punteggio ottenuto.[3] 20
2.20 Esempi della procedura di generazione degli ingredienti. Partendo da sinistra sono riportati, l'immagine di input, gli ingredienti predetti, gli ingredienti predetti dal modello R_{I2LR} e gli ingredienti reali. Sono evidenziati in rosso gli ingredienti non presenti nella ricetta originale, mentre in blu quelli corretti.[3] 21
2.21 Struttura del modello.[4] 24
2.22 Rappresentazione dei trasformatori utilizzati. TR (a) indica la struttura del singolo trasformatore, mentre HTR (b) indica l'architettura del trasformatore gerarchico.[4] 25
2.23 Funzione di perdita supervisionata per la sola ricetta. I cerchi colorati indicano le possibili combinazioni delle diverse componenti.[4] 26
2.24 Performance del modello considerando diverse combinazioni delle componenti di una ricetta. Gli ingredienti risultano essere le componenti che forniscono la maggiore quantità di informazioni.[4] . 27
2.25 Performance delle architetture con trasformatori rispetto a quelle con LSTM.[4] 27
2.26 Confronto tra diversi modelli per la realizzazione dell'encoder dell'immagine.[4] 28
2.27 Confronto tra i modelli proposti e lavori rilasciati negli anni precedenti.[4] 28

3.1 Distribuzione delle ricette secondo la portata di appartenenza per le partizioni di training, validation e test 32
3.2 Distribuzione del numero di ingredienti differenti per ogni ricetta . . 33
3.3 Distribuzione del numero di istruzioni per ogni ricetta 33
3.4 Distribuzione del numero di parole che compongono le diverse istruzioni di ogni ricetta 34
3.5 Andamento della cosine loss, sul training set, durante l'addestramento del modello unicamente su ricette italiane 37
3.6 Andamento della cosine loss, sul validation set, durante l'addestramento del modello unicamente su ricette italiane 37
3.7 Andamento della image loss, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane 38

3.8 Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane 38
3.9 Andamento della MedR, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane 39
3.10 Andamento del recall rate, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette 40
3.11 Andamento della cosine loss, sul training set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori 40
3.12 Andamento della cosine loss, sul validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori . . . 40
3.13 Andamento della image loss, sul training e validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori 41
3.14 Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori 41
3.15 Andamento della MedR, sul training e validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori 42
3.16 Andamento del recall rate, sul training e validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette . . 42
3.17 Andamento della Loss, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane 46
3.18 Andamento della Ingredient loss, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane 46
3.19 Andamento della Iou, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane 47
3.20 Andamento della Cardinality prediction error, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane 47
3.21 Andamento della F1, sul validation set, durante l'addestramento del modello unicamente su ricette italiane 48
3.22 Andamento della Loss, sul training e validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori 48

3.23 Andamento della Ingredient loss, sul training e validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori 49
3.24 Andamento della Iou, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane 49
3.25 Andamento della Cardinality prediction error, sul training e validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori 50
3.26 Andamento della F1, sul validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori 51
3.27 Andamento della cosine loss, sul training set, durante l'addestramento del modello, unicamente su ricette italiane, considerando solamente gli ingredienti che lo compongono ed escludendo le istruzioni 54
3.28 Andamento della cosine loss, sul validation set, durante l'addestramento del modello unicamente su ricette italiane, considerando solamente gli ingredienti che lo compongono ed escludendo le istruzioni 54
3.29 Andamento della image loss, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane, considerando solamente gli ingredienti che lo compongono ed escludendo le istruzioni 55
3.30 Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane, considerando solamente gli ingredienti che lo compongono ed escludendo le istruzioni 56
3.31 Andamento della MedR, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane, considerando solamente gli ingredienti che lo compongono ed escludendo le istruzioni 57
3.32 Andamento del recall rate, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane, considerando solamente gli ingredienti che lo compongono ed escludendo le istruzioni. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette . . 57
3.33 Andamento della cosine loss, sul training set, durante l'addestramento del modello sulle ricette italiane tradotte in lingua inglese, partendo dal checkpoint fornito dagli autori 60
3.34 Andamento della cosine loss, sul validation set, durante l'addestramento del modello sulle ricette italiane tradotte in lingua inglese, partendo dal checkpoint fornito dagli autori 60

3.35 Andamento della image loss, sul training e validation set, durante l'addestramento del modello sulle ricette italiane tradotte in lingua inglese, partendo dal checkpoint fornito dagli autori 61
3.36 Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello sulle ricette italiane tradotte in lingua inglese, partendo dal checkpoint fornito dagli autori 62
3.37 Andamento della MedR, sul training e validation set, durante l'addestramento del modello sulle ricette italiane tradotte in lingua inglese, partendo dal checkpoint fornito dagli autori 63
3.38 Andamento del recall rate, sul training e validation set, durante l'addestramento del modello sulle ricette italiane tradotte in lingua inglese, partendo dal checkpoint fornito dagli autori. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette 63
3.39 Nuova versione del database: distribuzione delle ricette secondo la portata di appartenenza per le partizioni di training, validation e test 66
3.40 Nuova versione del database: distribuzione del numero di ingredienti differenti per ogni ricetta 66
3.41 Nuova versione del database: distribuzione del numero di istruzioni per ogni ricetta 67
3.42 Nuova versione del database: distribuzione del numero di parole che compongono le diverse istruzioni di ogni ricetta 67
3.43 Andamento della cosine loss, sul training set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane 70
3.44 Andamento della cosine loss, sul validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane 70
3.45 Andamento della image loss, sul training e validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane 71
3.46 Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane 71
3.47 Andamento della MedR, sul training e validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane 72

3.48 Andamento del recall rate, sul training e validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette 73
3.49 Andamento della Loss, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane 76
3.50 Andamento della Ingredient loss, sul training e validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane 77
3.51 Andamento della Iou, sul training e validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane 77
3.52 Andamento della Cardinality prediction error, sul training e validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane 78
3.53 Andamento della F1, sul validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane 79
3.54 Andamento della loss, sul training e validation set, durante l'addestramento del modello $Model_{15\_}rec_{v1}$ 82
3.55 Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello $Model_{15\_}rec_{v1}$ 83
3.56 Andamento della paired loss, sul training e validation set, durante l'addestramento del modello $Model_{15\_}rec_{v1}$ 83
3.57 Andamento della medR, sul training set, durante l'addestramento del modello $Model_{15\_}rec_{v1}$ 84
3.58 Andamento della medR, sul validation set, durante l'addestramento del modello $Model_{15\_}rec_{v1}$ 85
3.59 Andamento del recall rate, sul training set, durante l'addestramento del modello $Model_{15\_}rec_{v1}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette 86
3.60 Andamento del recall rate, sul validation set, durante l'addestramento del modello $Model_{15\_}rec_{v1}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette 87
3.61 Andamento della loss, sul training e validation set, durante l'addestramento del modello $Model_{100\_}rec_{v1}$ 88

3.62 Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello $Model_{100\_rec_{v1}}$ 88
3.63 Andamento della paired loss, sul training e validation set, durante l'addestramento del modello $Model_{100\_rec_{v1}}$ 89
3.64 Andamento della medR, sul training set, durante l'addestramento del modello $Model_{100\_rec_{v1}}$ 89
3.65 Andamento della medR, sul validation set, durante l'addestramento del modello $Model_{100\_rec_{v1}}$ 90
3.66 Andamento del recall rate, sul training set, durante l'addestramento del modello $Model_{100\_rec_{v1}}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette 91
3.67 Andamento del recall rate, sul validation set, durante l'addestramento del modello $Model_{100\_rec_{v1}}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette 92
3.68 Andamento della loss, sul training e validation set, durante l'addestramento del modello $Model_{125\_rec_{v2}}$ 93
3.69 Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello $Model_{125\_rec_{v2}}$ 93
3.70 Andamento della paired loss, sul training e validation set, durante l'addestramento del modello $Model_{125\_rec_{v2}}$ 94
3.71 Andamento della medR, sul training set, durante l'addestramento del modello $Model_{125\_rec_{v2}}$ 94
3.72 Andamento della medR, sul validation set, durante l'addestramento del modello $Model_{125\_rec_{v2}}$ 95
3.73 Andamento del recall rate, sul training set, durante l'addestramento del modello $Model_{125\_rec_{v2}}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette 96
3.74 Andamento del recall rate, sul validation set, durante l'addestramento del modello $Model_{125\_rec_{v2}}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette 97
3.75 Andamento della loss, sul training e validation set, durante l'addestramento del modello $Model_{R50\_rec_{v2}}$ 98

3.76 Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello $Model_{R50\_rec_{v2}}$ 98
3.77 Andamento della paired loss, sul training e validation set, durante l'addestramento del modello $Model_{R50\_rec_{v2}}$ 99
3.78 Andamento della medR, sul training set, durante l'addestramento del modello $Model_{R50\_rec_{v2}}$ 99
3.79 Andamento della medR, sul validation set, durante l'addestramento del modello $Model_{R50\_rec_{v2}}$ 100
3.80 Andamento del recall rate, sul training set, durante l'addestramento del modello $Model_{R50\_rec_{v2}}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette 101
3.81 Andamento del recall rate, sul validation set, durante l'addestramento del modello $Model_{R50\_rec_{v2}}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette 102

Acronyms

AI
Artificial intelligence

LSTM
Long short-term memory

GRU
Gated recurrent unit

RNN
Recurrent neural network

LMDB
Lightning memory-mapped database

CCA
Canonical correlation analysis

EOS
End of sentence

NLP
Natural language processing

LR
Learning rate

Capitolo 1

Introduzione

1.1 Analisi del problema

Il cibo è uno degli aspetti fondamentali dell'esperienza umana. Il suo consumo è intrinsecamente legato alla nostra salute, alle nostre emozioni e alla nostra cultura. Come si dice "siamo ciò che mangiamo" e le attività connesse al cibo come cucinare, mangiare e discuterne, occupano una parte significativa della nostra quotidianità. Nell'era digitale attuale, l'accesso a una vasta gamma di risorse multimediali ha facilitato la diffusione della cultura culinaria. I video, siti web e tutorial dedicati alla preparazione di ricette varie sono diventati sempre più diffusi. Queste risorse forniscono informazioni dettagliate e passo-passo su come realizzare piatti di ogni genere, consentendo alle persone di esplorare e sperimentare in cucina. Inoltre, i social media, permettono alle persone di condividere foto e video dei piatti che preparano e consumano. Una semplice ricerca su Instagram con l'hashtag #food porta a più di 500 milioni di post, evidenziando il valore innegabile che il cibo ha nella nostra società. Inoltre, è necessario sottolineare come, soprattutto negli ultimi anni, si è assistito ad una diffusione sempre più ampia del concetto di alimentazione salutare. La consapevolezza dell'importanza di una dieta equilibrata e nutriente per il benessere generale è in costante aumento. Secondo uno studio, pubblicato nel 2022 su Nature Food, della Friedman School of Nutrition Science and Policy della Tuft University di Boston, in collaborazione con la McMaster University di Hamilton (Canada), nel trentennio esaminato, (1990-2018), si è beneficiato, in molte nazioni, di un crescente consumo di prodotti alimentari salutari, quali frutta secca, vegetali e legumi [5]. Tale tendenza nei confronti di una dieta più sana la si può riscontrare anche attraverso i social network, per esempio, l'hashtag #healtyfood produce oltre 110 milioni di risultati su Instagram. L'adozione di una dieta salutare non solo impatta positivamente sulla salute delle persone, ma ha anche importanti implicazioni a livello sociale ed ambientale, promuovendo

pratiche agricole sostenibili e il benessere degli animali. Per il mantenimento di uno stile di vita e di un regime alimentare più salutare è sempre più necessario quindi, tenere traccia dei diversi valori nutrizionali, (proteine, zuccheri, carboidrati), che caratterizzano in piatti consumati, operazione complessa che viene complicata ulteriormente quando si scelgono cibi preparati senza la cura domestica, come i piatti di ristoranti, fast food e take away, per i quali l'accesso a informazioni dettagliate sui cibi preparati è molto più limitato.

1.2 Obiettivi della tesi

Per aiutare quindi le persone in questo processo di tracciamento dei valori nutrizionali dei cibi consumati, si vuole realizzare un'applicazione che, attraverso una fotografia o un'immagine fornita dall'utente, sia in grado di riconoscere il cibo rappresentato, gli ingredienti di cui è composto e il volume di quest'ultimi, per poterne calcolare i valori nutrizionali associati. L'idea si ispira ad un'applicazione già esistente, chiamata FatSecret [6], che fornisce delle statiche basate sui valori nutrizionali dei piatti consumati dall'utente, attraverso l'inserimento testuale del nome e delle quantità di ogni piatto. La presente tesi di laurea si focalizza in particolare sullo studio e sulla realizzazione di un classificatore, in grado di stimare, con una buona precisione, gli ingredienti da cui è composto un piatto, rendendo più semplice il tracciamento dei valori nutrizionali e il mantenimento di una dieta bilanciata.

1.3 Contributo della tesi

Per il raggiungimento di tale obiettivo, lo sviluppo è stato suddiviso in diverse fasi, riguardanti principalmente:

- La ricerca e raccolta di ricette tipiche della cucina italiana, attraverso l'utilizzo di un apposito strumento di data mining, ParseHub [7];
- La raccolta di fotografie e immagini da associare alle ricette, attraverso un opportuno strumento di scraping, Google Image Scraper [8];
- La creazione di un database personalizzato, realizzato partendo dalla struttura del dataset Recipe1M [1], utilizzato come base per gli esperimenti sui modelli analizzati;
- Lo studio della struttura e del funzionamento di diversi modelli [2], [3], [4], con conseguente valutazione delle prestazioni sulle ricette italiane;
- L'analisi delle principali problematiche e limiti di questi modelli;

In particolare, nel capitolo seguente (2) vengono analizzati nel dettaglio i dataset e i modelli citati in precedenza, mentre nel capitolo 3 sono riportati i risultati degli esperimenti eseguiti per verificare le performance dei diversi modelli studiati.

Capitolo 2

State of the Art

2.1 Recipe1M

A causa della loro complessità, sia dal punto di vista visivo che da quello testuale, la comprensione delle ricette culinarie richiede una vasta raccolta di dati a supporto. Questo è dovuto principalmente all'elevata variabilità a cui è soggetta ogni singola ricetta, che può essere modificata in termini di possibili presentazioni del piatto, diversi ingredienti utilizzati o ancora diversi metodi di preparazione. Nel corso degli anni sono stati pubblicati alcuni dataset, caratterizzati da un discreto numero di ricette, [9], [10], [11], [12], [13], che tuttavia contenevano solamente immagini categorizzate, oppure semplicemente il testo delle ricette. Negli anni seguenti vengono rilasciate alcune raccolte di dati che presentano delle ricette, associate anche a delle immagini, nel 2015, per esempio, viene pubblicato un dataset, chiamato Food-101 [14], che contiene circa 101.000 immagini suddivise in modo equo tra 101 categorie diverse di cibo. Le ricette tuttavia erano rappresentate in formato HTML grezzo. L'anno seguente viene presentato un database [15] contenente 65.284 ricette, tipiche della cucina cinese, caratterizzate da una breve introduzione, un elenco di ingredienti e le istruzioni relative alla preparazione del piatto e 110.241 immagini annotate con 353 etichette di ingredienti. Sebbene i dataset sopra citati rappresentino un passo avanti verso un migliore apprendimento delle rappresentazioni delle ricette, sono ancora limitati sia in termini di dimensioni che di generalità. Questo fattore rappresenta un problema per l'apprendimento di rappresentazioni efficaci, in quanto, questa capacità dipende principalmente dalla quantità e dalla qualità dei dati disponibili. Al fine di limitare le problematiche descritte sopra, nel 2017, viene rilasciato un nuovo dataset su larga scala, chiamato Recipe1M [1], che, nella sua prima versione, è composto da oltre 1 milione di ricette e 800.000 immagini, contenente quindi il doppio delle ricette e un numero di immagini otto volte superiore, rispetto al database più ricco rilasciato fino a

quel momento. I dati, sono stati raccolti consultando più di due dozzine di famosi siti di cucina, sfruttando un processo di elaborazione che, ha estratto il testo significativo dall'HTML, ha scaricato le immagini correlate e ha assemblato i dati in uno schema JSON, assegnando un identificativo univoco ad ogni ricetta. In particolare sono stati realizzati due layer per contenere le informazioni relative alle diverse ricette, il primo, contiene informazioni basilari, come il titolo, la lista degli ingredienti, la descrizione della preparazione del piatto e l'identificativo univoco, mentre il secondo layer contiene i riferimenti alle diverse immagini raccolte per una determinata ricetta. Per un sottoinsieme di questi dati inoltre è possibile identificare la portata di appartenenza, consentendo, in parte, di verificare la distribuzione delle ricette raccolte, rappresentata nella Fig. 2.1. L'intero dataset è stato poi suddiviso in tre partizioni, come mostrato nella tabella della Fig. 2.2, dove circa il 70% delle ricette è stato etichettato per il training set, mentre le rimanenti sono state suddivise in maniera equa tra validation e test set.

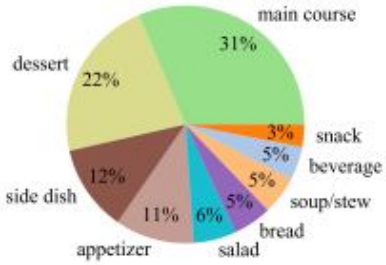


Figura 2.1: Statistiche Recipe1M. Distribuzione delle ricette in termini di portate.[1]

Partition	# Recipes	# Images
Training	720,639	619,508
Validation	155,036	133,860
Test	154,045	134,338
Total	1,029,720	887,706

Figura 2.2: Statistiche Recipe1M. Partizionamento delle ricette.[1]

Una volta terminata la raccolta dei dati, sono state eseguite una serie di analisi statistiche, che hanno evidenziato come solamente 0.4% delle ricette e il 2% delle immagini raccolte risultino duplicate. Questi dati, sottolineano gli autori, sono stati rimossi, per evitare la sovrapposizione tra training e test set. Nella Fig. 2.3 vengono evidenziate le distribuzioni dei dati in termini di numero di ingredienti e

numero di istruzioni. Inoltre, sono stati identificati 16.000 ingredienti unici, di cui però, soltanto 4.000 rappresentano il 95% delle occorrenze. Questa tendenza viene riscontrata anche nelle immagini raccolte, dove l'1% delle ricette include il 25% di esse, mentre la metà di tutte le immagini è associata al 10% delle ricette. Di conseguenza, è importante sottolineare come, di tutti i dati disponibili, solamente 333.000 ricette siano effettivamente associate con delle immagini.

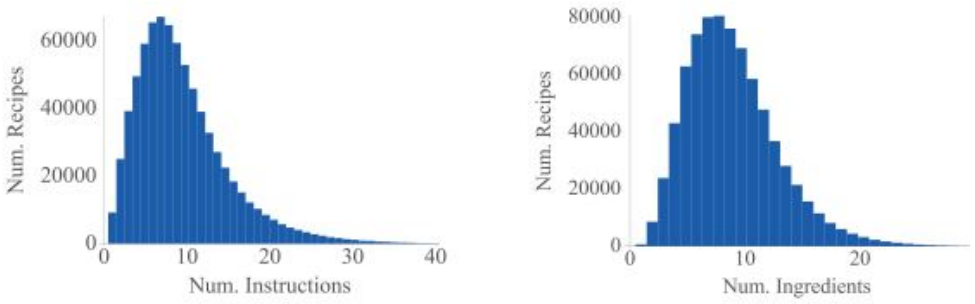


Figura 2.3: Statistiche Recipe1M. Distribuzione del numero di ingredienti e numero di istruzioni per ricetta.[1]

Nonostante il dataset prodotto fosse il più grande e ricco rilasciato fino a quel momento nel campo delle ricette culinarie, il numero totale di immagini era inferiore alle più grandi raccolte di dati pubblicamente disponibili, come ImageNet [16] e Places [17], che ne includono decine di milioni. Per superare questo limite, gli autori hanno prodotto e rilasciato una versione aggiornata del dataset, chiamata Recipe1M+, [2], che presenta lo stesso numero di ricette della precedente versione, ma che incrementa notevolmente il numero di immagini, passando da 800.000 a 13 milioni. Questi dati sono stati raccolti sfruttando Google come motore di ricerca e il titolo di ogni ricetta come query. Per ogni ricetta sono stati considerati gli URL delle prime 50 immagini trovate, collezionandone in totale circa 50 milioni. Una volta scaricate localmente, a causa di alcuni di indirizzi corrotti o non più disponibili, il numero è stato ridotto a 47 milioni. In seguito, è stata eseguita un'operazione di pulizia, che ha rimosso le immagini duplicate o molto simili, quelle in cui era possibile riconoscere dei volti umani e quelle che presentavano del testo, ottenendo come valore finale 13.735.679. Nella tabella della Fig. 2.5, vengono mostrati con maggiore dettaglio i dati relativi alle due versioni del dataset. La colonna *intersection*, come specificato dagli autori, rappresenta il numero di immagini in comune tra i due. Il rilascio di questa nuova versione permette di risolvere uno dei principali limiti del lavoro precedente, ovvero il numero di immagini associate alla singola ricetta, che risultava essere circa 0.86, ovvero meno di una per ricetta. In Recipe1M+ invece, questo valore per singola ricetta è decisamente superiore, come si può chiaramente vedere nella Fig. 2.4, dove il numero di immagini per più di mezzo milione di ricette è superiore a 12.

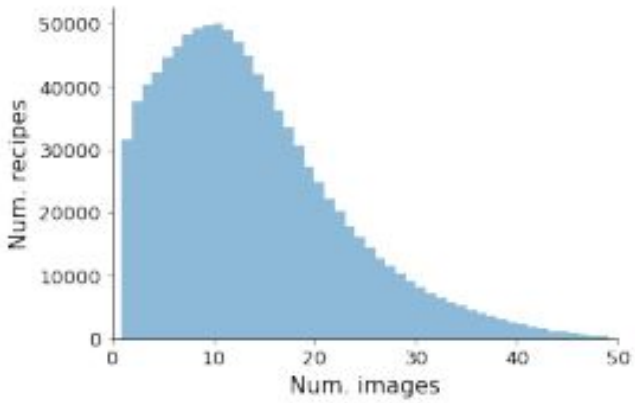


Figura 2.4: Statistiche Recipe1M+. Distribuzione del numero di immagini per ricetta.[2]

		Recipe1M	intersection	Recipe1M+
Partition	# Recipes	# Images	# Images	# Images
Training	720,639	619,508	493,339	9,727,961
Validation	155,036	133,860	107,708	1,918,890
Test	154,045	134,338	115,373	2,088,828
Total	1,029,720	887,706	716,480	13,735,679

Figura 2.5: Statistiche Recipe1M+. Confronto tra il partizionamento di Recipe1M e Recipe1M+.[2]

2.2 Img2recipe

Insieme al rilascio del database Recipe1M e Recipe1M+, gli autori hanno sviluppato anche un modello, chiamato img2recipe, che tenta di associare le immagini delle ricette con la ricetta stessa, per cercare di risolvere il difficile task del passaggio da immagine a ricetta e viceversa [2].

2.2.1 Modello

Img2recipe è un modello multimodale, ovvero un modello in grado di processare input di diversa modalità, in questo caso, immagini e testo. In particolare il modello, partendo dalle ricette e dalle immagini ad esse associate, tenta di elaborarle al fine di apprendere uno spazio comune di rappresentazione, come illustrato nella Fig. 2.6. In termini più semplici, il modello tenta di associare nella maniera più precisa possibile il testo delle ricette con le relative immagini, permettendo così di ricavare la ricetta a partire dalla sola immagine e viceversa. Per raggiungere questo

obiettivo, gli autori hanno realizzato una struttura che, prendendo in considerazione le diverse parti di una ricetta, quali, ingredienti, istruzioni e immagini estrae per ogni componente una rappresentazione, attraverso degli appositi encoder. Quest'ultime vengono utilizzate poi per ricavare uno spazio di rappresentazione comune, che viene regolato tramite l'utilizzo di apposite funzioni di perdita. La Fig. 2.7 mostra con maggiore chiarezza quanto descritto sopra. Nel seguito vengono approfonditi gli aspetti più rilevanti della struttura sopra descritta.

Encoder dell'immagine

Per ottenere una rappresentazione efficace delle immagini vengono utilizzati i modelli VGG-16 [18] e Resnet50 [19], ovvero, due reti convoluzionali profonde all'avanguardia. In particolare, queste reti risultano molto efficaci in termini di prestazioni, grazie all'utilizzo di mappe di identità ubiquitarie, che consentono l'addestramento di architetture molto più profonde (ad esempio, con 50 o 101 strati) con migliori prestazioni. Come mostrato sul lato destro della Fig. 2.7, questi modelli vengono incorporati, rimuovendo prima l'ultimo strato di classificazione softmax e collegandoli poi con il resto del modello.

Encoder degli ingredienti

Come illustrato nella Fig. 2.7, da ogni ricetta viene estratta la lista degli ingredienti di cui è composta. Per ogni ingrediente viene ricavata una rappresentazione word2vec [20], ovvero costituita da vettori numerici densi, grazie alla quale, parole uguali o simili ottengono una rappresentazione vettoriale simile. Prima della conversione, dal testo degli ingredienti viene estratto solamente il nominativo dell'ingrediente stesso, ripulendolo da elementi aggiuntivi come, le quantità e le unità di misura. Per essere chiari, dal testo *100 ml di latte* verrà estratta solamente la parola *latte*. Per fare ciò è stato utilizzato un LSTM bidirezionale che esegue una regressione logistica su tutte le parole nel testo degli ingredienti. Il training di questa architettura viene eseguito su un sottoinsieme del training set, ottenendo una precisione nell'estrazione degli ingredienti pari al 99.5%.

Encoder delle istruzioni

Ogni ricetta include anche le istruzioni per la preparazione e la cottura del piatto. A causa della loro lunghezza, in media circa 208 parole, un singolo LSTM non è adatto a rappresentarle, a causa dell'eccessiva riduzione dei gradienti dovuta ai molti passaggi temporali. Di conseguenza, viene proposto un modello LSTM a due fasi, in grado quindi di codificazione una sequenza di sequenze. Nella prima fase, vengono creati dei vettori chiamati *skip-instructions* per ogni frase o istruzione. In seguito, un LSTM viene addestrato sulla sequenza di questi vettori per ottenere una

codifica a lunghezza fissa, che viene poi unita alla rappresentazione degli ingredienti per formare la codifica finale della ricetta. Gli *skip-instructions*, sono il risultato di un modello sequence to sequence [21] che utilizza la tecnica degli skipthoughts [22], che permette di codificare una frase e utilizzarla per predire e decodificare le frasi precedenti e successive, come mostrato nella Fig. 2.8. In particolare gli autori hanno apportato delle modifiche a questo metodo utilizzando un LSTM al posto di un GRU e introducendo delle istruzioni di inizio e fine della ricetta.

Rappresentazione in uno spazio condiviso

Utilizzando le rappresentazioni delle ricette e delle immagini descritte in precedenza, il modello tenta di imparare le trasformazioni necessarie per rendere queste rappresentazioni il più vicine possibile per una coppia di immagine-ricetta. La Fig. 2.7 illustra bene questo processo, nel quale i risultati degli encoder degli ingredienti e delle istruzioni vengono incorporati e concatenati in uno spazio condiviso tra la ricetta e l'immagine. Quest'ultima poi viene proiettata in questo spazio attraverso una trasformazione lineare. Per verificare la similarità tra l'immagine e la ricetta viene utilizzata una tecnica euristica chiamata similarità del coseno, che calcola la similitudine tra due vettori basandosi sul valore del coseno tra di essi. Definendo quindi come (ϕ^r, ϕ^v) la rappresentazione nello spazio condiviso rispettivamente, della ricetta e dell'immagine, si definisce la funzione di perdita della similarità coseno come segue:

$$L_{\cos}(\phi^r, \phi^v, y) = \begin{cases} 1 - \cos(\phi^r, \phi^v), & \text{if } y = 1 \\ \max(0, \cos(\phi^r, \phi^v) - \alpha), & \text{if } y = -1 \end{cases}$$

dove il cos(.) rappresenta la similarità coseno normalizzata e α il margine.

Regolarizzazione semantica

Nella risoluzione del problema descritto precedentemente, gli autori hanno deciso di introdurre una regolarizzazione aggiuntiva, chiamata regolarizzazione semantica, basata sulla condivisione di pesi discriminativi ad alto livello, in modo che gli encoder delle immagini e delle ricette utilizzino questi pesi in modo simile, aggiungendo un ulteriore livello di allineamento basato sulla discriminazione. In termini più semplici, il modello tenta di imparare come inserire le codifiche delle ricette o delle immagini all'interno di categorie semantiche relative al cibo. Queste ultime vengono estratte partendo dai titoli delle ricette presenti nel training set e considerando le categorie definite nel dataset Food-101 [14]. Per incorporare questo elemento aggiuntivo all'interno dell'architettura viene inserito un fully connected layer. Viene poi definita la perdita di regolarizzazione semantica come $L_{\text{reg}}(\phi^r, \phi^v, c_r, c_v)$, dove (ϕ^r, ϕ^v) rappresentano la codifica nello spazio condiviso rispettivamente, della ricetta

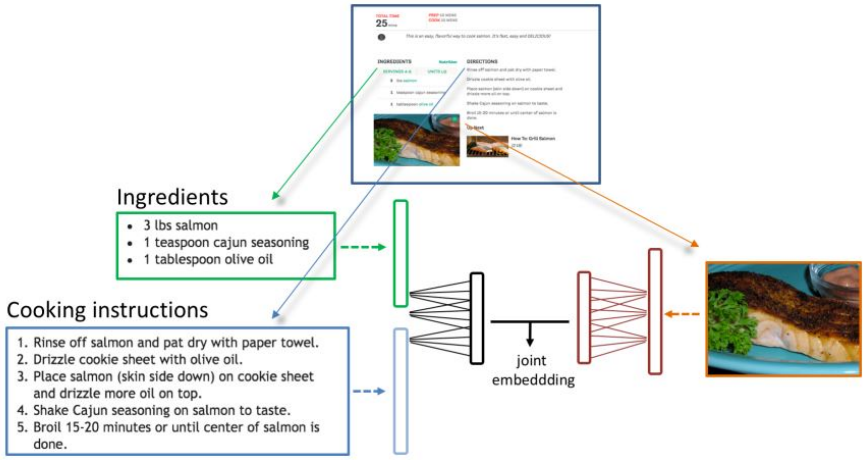


Figura 2.6: Estrazione delle informazioni dalle ricette.[2]

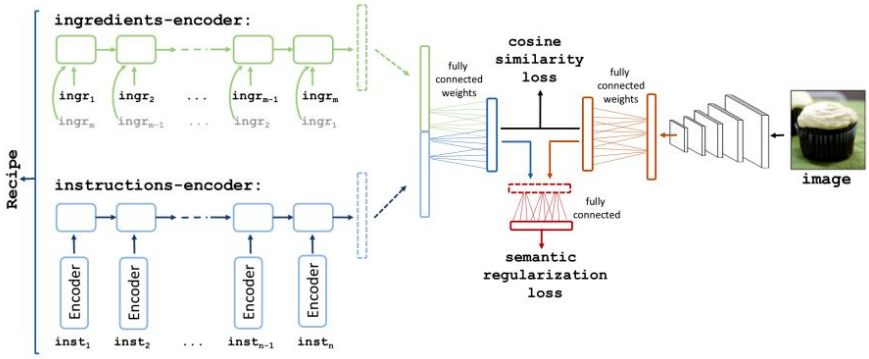


Figura 2.7: Rappresentazione del modello Img2recipe di Recipe1M.[2]

e dell'immagine, mentre (c_r, c_v) sono le label delle categorie semantiche della ricetta e dell'immagine, rispettivamente. Infine, è possibile descrivere la funzione obiettivo, come somma della perdita della similarità coseno e della perdita di regolarizzazione semantica, pesata attraverso un parametro λ:

$$L(\phi^r, \phi^v, c_r, c_v, y) = L_{\cos}(\phi^r, \phi^v, y) + \lambda L_{\text{reg}}(\phi^r, \phi^v, c_r, c_v)$$

2.2.2 Metriche

Nella sezione degli esperimenti verranno riportati i risultati dei procedimenti di image-to-recipe e recipe-to-image, ovvero la selezione delle ricette più simili, in termini di similarità coseno all'interno dello spazio comune, ottenute partendo da un'immagine di input e viceversa. Come metodo di valutazione, gli autori hanno deciso di seguire la procedura descritta dall'*image2caption retrivial task* [23] [24],

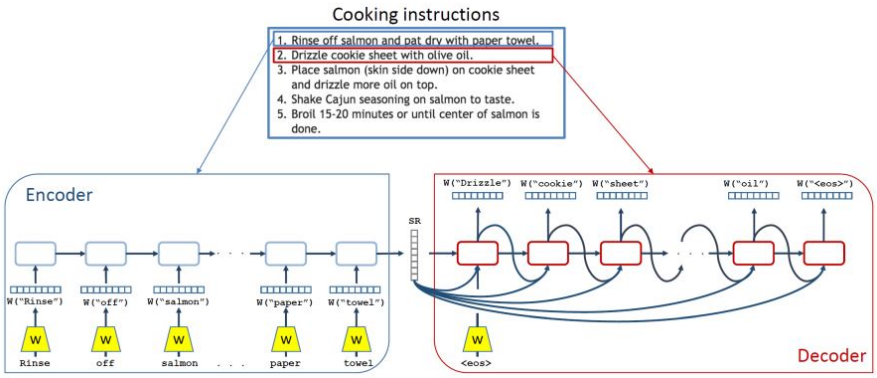


Figura 2.8: Modello di Skip-instructions.[2]

nella quale, si calcolano i risultati su un sottoinsieme di 1000 coppie immagine-ricetta, prese dal validation set. Vengono utilizzate come metriche il median rank (MedR), che rappresenta la media delle posizioni in cui si trova la ricetta che è associata correttamente all'immagine di input e il recall rate di K (R@K), ovvero la percentuale delle immagini per cui la ricetta correttamente associata si trovava nelle prime K posizioni. In particolare vengono definite la R@1, R@5, R@10. Queste valutazioni vengono ripetute 10 volte, su sottoinsiemi differenti e vengono poi considerati i valori medi.

2.2.3 Esperimenti e risultati

L'implementazione dei modelli e degli esperimenti è stata realizzata utilizzando il framework PyTorch [25], impostando un valore di 0.1 per il margine α e un valore di 0.02 per l'iperparametro λ. Il training dei modelli e gli esperimenti sono stati condotti su 4 NVIDIA GTX 1080 con 8GB di memoria per circa 60 ore per quanto riguarda il dataset Recipe1M, mentre per Recipe1M+, a causa della maggiore complessità, ha impiegato più di una settimana, utilizzando batch con una dimensione di 256. Le immagini, per motivi di efficienza, vengono caricate utilizzando la funzione di DataLoader della libreria PyTorch, mentre il dataset è stato salvato in formato LMDB. Nelle tabelle seguenti vengono mostrate le sperimentazioni sull'efficacia del modello sopra descritto. In particolare, nella Fig. 2.10 vengono riportati i risultati ottenuti dal modello sul dataset Recipe1M, confrontati con quelli ottenuti da CCA, uno dei modelli statistici più efficaci per imparare una rappresentazione comune per diversi spazi di caratteristiche. Dai dati è possibile osservare come il nuovo modello risulti superare di molto le performance del CCA, sia per quanto riguarda la image-to-recipe, sia per la recipe-to-image. Nella tabella riportata nella Fig. 2.9 invece, viene mostrato un

confronto tra l'utilizzo di un'architettura VGG-16, rispetto ad una Resnet-50, con diverse configurazioni. Nello specifico, vengono confrontati i valori di MedR riferiti a 1K, 5K e 10K elementi presi in considerazione e si può notare come Resnet-50 ottenga prestazioni superiori in tutti gli esperimenti. Nella Fig. 2.11, sono stati confrontati due modelli, trainati uno sul dataset Recipe1M e l'altro su Recipe1M+. Le performance sono state poi testate, sia sul test set di Recipe1M, dove i valori di MedR e R@K sono simili, sia su quello di Recipe1M+, nel quale, il modello trainato su di esso ottiene risultati di MedR, R@5 e R@10 decisamente superiori, dimostrando l'efficacia dell'introduzione di numerose immagini per la singola ricetta. Infine, nella Fig. 2.12 vengono mostrati degli esempi dei risultati del processo di image-to-recipe.

	Joint emb. methods	im2recipe			recipe2im		
		medR-1K	medR-5K	medR-10K	medR-1K	medR-5K	medR-10K
VGG-16	fixed vision	15.3	71.8	143.6	16.4	76.8	152.8
	finetuning (ft)	12.1	56.1	111.4	10.5	51.0	101.4
	ft + semantic reg.	8.2	36.4	72.4	7.3	33.4	64.9
ResNet-50	fixed vision	7.9	35.7	71.2	9.3	41.9	83.1
	finetuning (ft)	7.2	31.5	62.8	6.9	29.8	58.8
	ft + semantic reg.	5.2	21.2	41.9	5.1	20.2	39.2

Figura 2.9: Confronto tra architettura VGG-16 e Resnet-50.[2]

	im2recipe				recipe2im			
	medR	R@1	R@5	R@10	medR	R@1	R@5	R@10
random ranking	500	0.001	0.005	0.01	500	0.001	0.005	0.01
CCA w/ skip-thoughts + word2vec (GoogleNews) + image features	25.2	0.11	0.26	0.35	37.0	0.07	0.20	0.29
CCA w/ skip-instructions + ingredient word2vec + image features	15.7	0.14	0.32	0.43	24.8	0.09	0.24	0.35
joint emb. only	7.2	0.20	0.45	0.58	6.9	0.20	0.46	0.58
joint emb. + semantic	5.2	0.24	0.51	0.65	5.1	0.25	0.52	0.65

Figura 2.10: Confronto del modello introdotto rispetto al modello CCA.[1]

2.3 Inverse Cooking

Il secondo modello analizzato è Inverse Cooking [3], un progetto rilasciato nel giugno del 2019, la cui finalità è quella di generare una nuova ricetta, caratterizzata da titolo, ingredienti e istruzioni a partire solamente da un'immagine. Il sistema viene addestrato e valutato sul dataset Recipe1M, ampiamente descritto nella sezione precedente 2.1.

	Recipe1M test set				Recipe1M+ test set			
	im2recipe							
	medR	R@1	R@5	R@10	medR	R@1	R@5	R@10
Recipe1M training set	5.1	0.24	0.52	0.64	13.6	0.15	0.35	0.46
Recipe1M+ training set	5.7	0.21	0.49	0.62	8.6	0.17	0.42	0.54
	recipe2im							
	medR	R@1	R@5	R@10	medR	R@1	R@5	R@10
Recipe1M training set	4.8	0.27	0.54	0.65	11.9	0.17	0.38	0.48
Recipe1M+ training set	4.6	0.26	0.54	0.66	6.8	0.21	0.46	0.58

Figura 2.11: Confronto tra modelli addestrati su Recipe1M e Recipe1M+. Con Recipe1M in questo caso si fa riferimento alle ricette e immagini comuni tra le due versioni del dataset.[2]

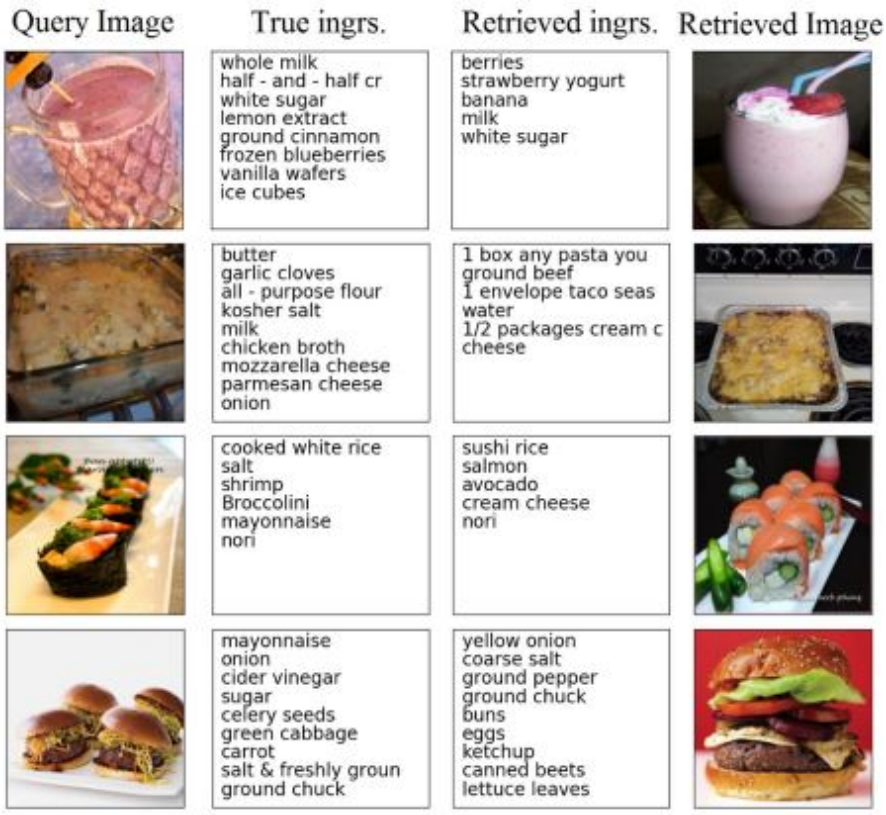


Figura 2.12: Risultati del processo di image-to-recipe. Sono riportati, da sinistra a destra, le immagini di input, i reali ingredienti della ricetta, gli ingredienti e l'immagine restituiti dal processo.[2]

2.3.1 Modello

L'obiettivo di questo modello è quello di generare una ricetta (titolo, ingredienti e istruzioni) partendo da un'immagine. Un compito decisamente arduo, in quanto è necessaria sia una comprensione degli ingredienti che compongono il piatto sia delle lavorazioni che questi hanno subito. Per far fronte a questo difficile task, il modello presenta una pipeline di generazione di ricette, suddivisa principalmente in due stage. Nel primo si tenta di predire l'elenco degli ingredienti partendo dall'immagine, nel secondo invece, utilizzando il risultato ottenuto nello stage precedente e l'immagine, viene generata la sequenza di istruzioni relative alla preparazione della ricetta. La divisione in fasi, secondo gli autori, può fornire delle informazioni aggiuntive sulle modalità di elaborazione degli ingredienti, rispetto all'analisi della sola immagine. Nella Fig. 2.15 viene mostrata una rappresentazione della struttura del modello realizzato, il quale, come detto in precedenza, riceve l'immagine di un piatto come input e genera una sequenza di istruzioni per la preparazione. Questa sequenza viene creata attraverso l'utilizzo di un decoder che lavora a partire da due rappresentazioni. La prima contiene le caratteristiche visive estratte dall'immagine, mentre la seconda, la codifica degli ingredienti estratti precedentemente. Nel seguito vengono descritti con maggiore approfondimento i diversi elementi che compongono l'architettura del modello.

Decoder degli ingredienti

Gli autori hanno eseguito uno studio approfondito sulla migliore rappresentazione possibile per gli ingredienti, in particolare hanno studiato i vantaggi e gli svantaggi della visione di essi come un insieme o come una lista. Da un lato, la rappresentazione come un insieme sembrerebbe essere la più accurata, poiché cambiando l'ordine di scrittura non cambia il risultato di una ricetta. Al contrario però, nell'ideale comune, ci si riferisce spesso agli ingredienti come lista, un banale esempio sta nel nome comunemente utilizzato, ovvero *lista degli ingredienti*, che prevede quindi un ordine specifico. Nel seguito verranno descritti modelli che lavorano sia con gli insiemi sia con le liste. Nella rappresentazione come lista abbiamo che gli ingredienti vengono rappresentati in maniera univoca con un ordine specifico. Definendo quindi un dizionario di ingredienti D, è possibile ricavare una lista L estraendo K elementi e mantenendone l'ordine. L'obiettivo quindi è quello di associare ad ogni immagine una ricetta e per fare ciò si tenta di massimizzare la seguente funzione obiettivo:

$$\arg\max_{\theta_I,\theta_L} \sum_{i=0}^{M} \log p(\hat{L}^{(i)} = L^{(i)} | x^{(i)}; \theta_I, \theta_L)$$

dove $\hat{L}$ rappresenta la lista di ingredienti che deve essere generata, la $x^{(i)}$ indica l'i-esima immagine di cui vogliamo ricavare la ricetta, mentre θ_I e θ_L rappresentano i

parametri rispettivamente, dell'encoder dell'immagine e del decoder degli ingredienti, e M si riferisce al numero totale di immagini. Un possibile svantaggio di questa rappresentazione può essere l'introduzione di una penalizzazione per il mancato ordinamento, che tuttavia potrebbe non essere rilevante per gli ingredienti. Se invece consideriamo la rappresentazione come insieme, possiamo definire S, l'insieme di K elementi estratti dal dizionario D, senza un ordine specifico. In questo caso, si vuole predire il set di ingredienti, $\hat{s}$, associati ad un'immagine massimizzando la seguente funzione obiettivo:

$$\arg\max_{\theta_I,\theta_L} \sum_{i=0}^{M} \log p(\hat{s}^{(i)} = s^{(i)} | x^{(i)}; \theta_I, \theta_L)$$

dove θ_I, θ_L e M assumono lo stesso significato del caso precedente. Il possibile svantaggio di questa soluzione invece è quello di non considerare eventuali relazioni tra gli ingredienti di una ricetta, un esempio significativo sono il *sale* e il *pepe* che appaiono spesso insieme. Per l'implementazione di questi modelli viene utilizzato un trasformatore [26], un modello di deep learning, che utilizza il meccanismo dell'auto-attenzione, che permette di considerare l'intero insieme di dati di input contemporaneamente. A differenza di altre architetture, come le RNN, se per esempio l'input è rappresentato da una frase, il trasformatore non è limitato all'elaborazione di una singola parola alla volta, permettendo dunque una maggiore parallelizzazione e riducendo i tempi di addestramento. Nella Fig. 2.13 è rappresentata la struttura del trasformatore, chiamato *set tranformer* dagli autori, nel quale, per rimuovere l'ordine in cui vengono predetti gli ingredienti, vengono aggregati gli output prodotti in diversi istanti di tempo, utilizzando un'operazione di max pooling. Inoltre, per impedire che gli ingredienti risultino duplicati, viene impostata a $-\infty$ la probabilità di ogni ingrediente già scelto. Il modello viene addestrato con l'obiettivo di minimizzare una funzione di perdita, ottenuta tramite una somma pesata di diverse componenti, quali:

- Binary cross-entropy loss, utilizzata per misurare la differenza tra gli ingredienti predetti dal modello e quelli reali;
- Binary cross-entropy loss, chiamata anche *eos loss*, calcolata tra la probabilità predetta di trovare l'*eos* nei diversi istanti di tempo e la sua corretta posizione;
- Cardinality penality l_1, che come riportato dagli autori, risulta empiricamente utile, ovvero aiuta l'ottimizzazione del modello durante il training;

Un'altra soluzione proposta prevede invece l'utilizzo di una target distribution:

$$p(s^{(i)}|x^{(i)}) = \frac{s^{(i)}}{\sum_j s_j^{(i)}}$$

[27] [28] per modellare la distribuzione degli elementi, con l'obiettivo di minimizzare la cross-entropy loss tra $p(s^{(i)}|x^{(i)})$ e la distribuzione prodotta dal modello in output $p(\hat{s}^{(i)}|x^{(i)})$. Gli autori si riferisco a questo modello chiamandolo *feed forward* model.

Decoder delle istruzioni

Per predire una serie di istruzioni inerenti alla preparazione di una ricetta, a partire solamente dall'immagine, gli autori hanno deciso di implementare un decoder attraverso l'utilizzo di un trasformatore. Quest'ultimo riceve come input la rappresentazione dell'immagine, e_I , ottenuta tramite un encoder ResNet-50 [19] e la rappresentazione degli ingredienti, e_L, predetti attraverso il decoder descritto nella precedente sezione, con l'aggiunta di un layer per il mapping di ogni ingrediente in un vettore di lunghezza fissa. Nella Fig. 2.14 nella parte sinistra è riportata una rappresentazione del modello del trasformatore, formato da diversi blocchi chiamati *transformer blocks*, seguiti da un layer lineare e un sofmax, che forniscono come output la distribuzione, ad ogni tempo t, delle parole che compongono una ricetta. I *transformer blocks* sono formati da due attention layer, il primo dei quali si concentra sull'applicazione dell'auto-attenzione sui risultati precedentemente generati, mentre il secondo si occupa di modificare i parametri del modello sulla base dei valori di auto-attenzione prodotti. Per garantire che l'attenzione del trasformatore sia influenzata simultaneamente sia dalla rappresentazione dell'immagine, e_I , sia dalla rappresentazione degli ingredienti, e_L, sono state esplorate tre diverse strategie:

- *Attenzione concatenata*, che consiste prima, nella concatenazione dell' e_I e dell' e_L e in un secondo momento nell'applicazione dell'attenzione;
- *Attenzione indipendente*, che prevede l'utilizzo di due attention layers, che operano in maniera separata sulle due diverse rappresentazioni e_I e e_L, per poi combinarne i risultati in seguito;
- *Attenzione sequenziale*, che consiste nell'elaborazione sequenziale delle due rappresentazioni, considerando due possibili ordinamenti, il primo dei quali prevede di considerare prima e_I e in seguito l'e_L e viceversa per il secondo;

Queste tecniche sono rappresentate nella parte destra della Fig. 2.14

2.3.2 Metriche

Nella sezione degli esperimenti vengono riportati i risultati ottenuti nella generazione degli ingredienti e dell'intera ricetta. Per la valutazione vengono utilizzate come metriche l'Intersection over Union (IoU), che misura la sovrapposizione tra gli ingredienti generati e quelli reali, il cardinality prediction error, ovvero l'errore di previsione della cardinalità, che rappresenta la differenza tra la cardinalità

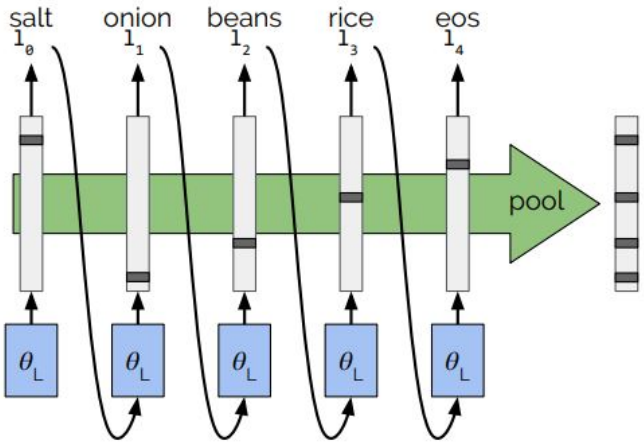
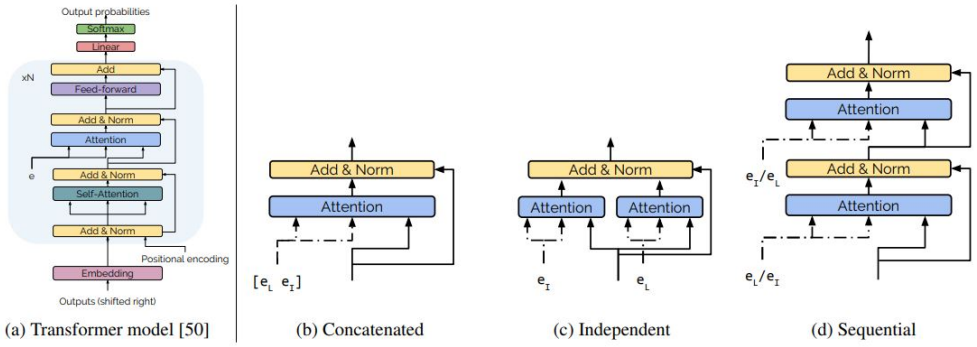


Figura 2.13: Rappresentazione del decoder utilizzato per la generazione degli ingredienti relativi ad un'immagine di input.[3]



(a) Transformer model [50] (b) Concatenated (c) Independent (d) Sequential

Figura 2.14: Encoder delle istruzioni. Sulla sinistra (a) viene mostrato il modello del trasformatore utilizzato, mentre nella parte di destra (b, c, d) sono riportate le strategie di fusione adottate.[3]

dell'insieme di dati generato e quella reale e l'F1, un punteggio che rappresenta una media tra le metriche di precisione e richiamo, utilizzata soprattutto su dati non bilanciati. Essa viene calcolata sulla base dei valori di TP (Veri positivi), FN (Falsi negativi) e FP (Falsi positivi), secondo la seguente formulazione:

$$\mathrm{F1} = 2 \times \frac{\text{precisione} \times \text{richiamo}}{\text{precisione} + \text{richiamo}}$$

dove la precisione e il richiamo sono ottenute tramite le seguenti espressioni:

$$\text{Precisione} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \qquad \text{Richiamo} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

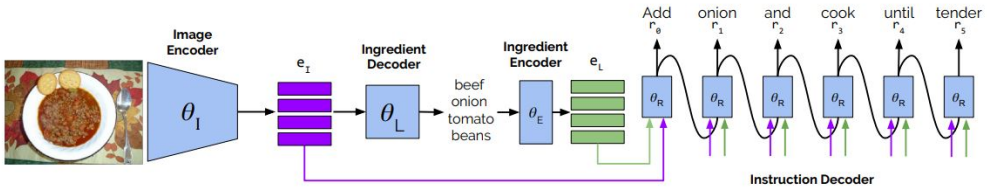


Figura 2.15: Architettura del modello InverseCooking.[3]

2.3.3 Esperimenti e risultati

L'implementazione del modello e dei diversi esperimenti è stata realizzata utilizzando il framework PyTorch [25]. In particolare, è stato utilizzato un trasformatore composto da 16 blocchi e 8 multi-head attentions, per il decoder delle istruzioni, mentre per il decoder degli ingredienti il trasformatore è composto da 4 blocchi e 2 multi-head attentions. La dimensione della rappresentazione degli ingredienti e delle immagini è stata impostata a 512, scegliendo 20 come numero massimo di ingredienti per ricetta e limitando la lunghezza delle istruzioni a non più di 150 parole. L'addestramento del modello è stato suddiviso in due fasi, la prima che consiste nell'pre-addestramento dell'encoder dell'immagine e del decoder degli ingredienti, mentre la seconda si concentra sull'encoder degli ingredienti e sul decoder delle istruzioni. Come accennato in precedenza il dataset utilizzato per i diversi esperimenti è Recipe1M, al quale però sono state apportate delle leggere modifiche. Nello specifico, è stata ridotta la dimensione del dizionario degli ingredienti, passando da 16k a 1488, attraverso l'unione di tutti gli ingredienti che condividevano le prime, o le ultime due parole, la rimozione dei plurali e infine non considerando gli ingredienti che apparivano meno di 10 volte. Nella tabella della Fig. 2.18 sono rappresentati i risultati ottenuti, in temrmi di IoU e F1 dei diversi modelli considerati dagli autori. Il modello *set transformer* in particolare risulta essere il migliore, sottolineando l'importanza della considerazione delle dipendenze tra ingredienti, senza però penalizzare l'ordinamento imposto. Questa superiorità viene confermata anche nei grafici riportati nella Fig. 2.19, che mostrano, nella parte sinistra, l'andamento della precisione per diversi valori di K, mentre nella parte destra i valori di F1 ottenuti dai diversi ingredienti. Tuttavia il modello proposto non eccelle nelle performance riferite alla valutazione del cardinality prediction error. Come si può osservare nella Fig. 2.17 infatti, il *set transformer* ottiene risultati inferiori rispetto a (FF_{IOU} e TF_{list}). Sulla base dei risultati ottenuti gli autori hanno infine selezionato il *set transformer* e *feed forward* per un confronto ulteriore con il modello analizzato nella precedente sezione 2.2, indicato come R_{I2LR}. I valori di questo confronto sono riportati nelle tabelle della Fig. 2.16, dalle quali risulta chiara la netta superiorità delle soluzioni proposte dagli autori. Infine,

vengono mostrati nella Fig. 2.20 alcuni esempi del processo di generazione degli ingredienti.

	IoU	F1
R_{I2L} [45]	18.92	31.83
R_{I2LR} [45]	19.85	33.13
FF_{TD} (ours)	29.82	45.94
TF_{set} (ours)	**32.11**	**48.61**

	Rec.	Prec.
R_{IL2R}	31.92	28.94
Ours	**75.47**	**77.13**

Figura 2.16: Confronto tra i migliori modelli proposti dagli autori e il modello R_{I2LR}. R_{I2L} indica una versione alternativa del modello che considera solamente gli ingredienti, tralasciando il titolo e le istruzioni.[3]

	Card. error	# pred. ingrs
FF_{BCE}	5.67 ± 3.10	2.37 ± 1.58
FF_{DC}	2.68 ± 2.07	9.18 ± 2.06
FF_{IOU}	2.46 ± 1.95	7.86 ± 1.72
FF_{TD}	3.02 ± 2.50	8.02 ± 3.24
TF_{list}	2.49 ± 2.11	7.05 ± 2.77
TF_{list} + shuffle	3.24 ± 2.50	5.06 ± 1.85
TF_{set}	2.56 ± 1.93	9.43 ± 2.35

Figura 2.17: Confronto tra i diversi modelli proposti in termini di cardinality prediction error. Sulla destra viene anche riportata la media del numero di ingredienti predetti.[3]

2.4 Revamping Cross-Modal Recipe Retrieval with Hierarchical Transformers and Self-supervised Learning

L'ultimo modello analizzato è stato sviluppato da un team di ricercatori per conto di Amazon e rilasciato nel giugno del 2021 [4]. Essendo il più recente tra i modelli considerati, utilizza tecniche innovative, grazie alle quali rappresenta lo state of the art per le architetture che si concentrano sul processo di associazione tra immagine e

Model	IoU	F1
FF_{BCE}	17.85	30.30
FF_{IOU}	26.25	41.58
FF_{DC}	27.22	42.80
FF_{TD}	**28.84**	**44.11**
TF_{list}	29.48	45.55
TF_{list} + shuf.	27.86	43.58
TF_{set}	**31.80**	**48.26**

Figura 2.18: Confronto tra i diversi modelli proposti in termini di IoU e F1.[3]

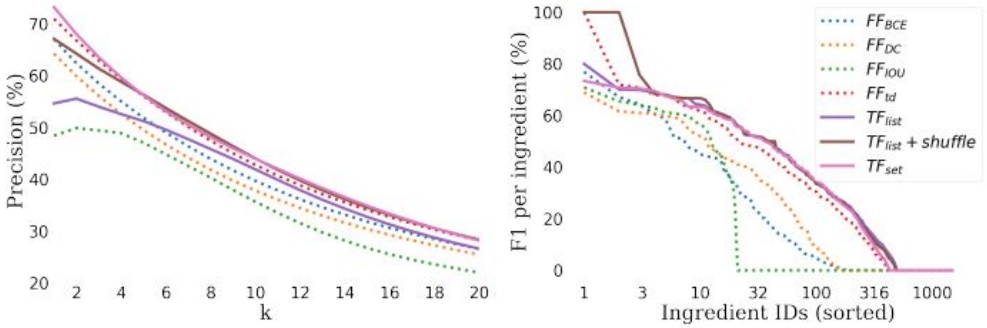


Figura 2.19: Confronto delle prestazioni dei diversi modelli in termini di precisione su diversi valori di K,.., e F1 per i singoli ingredienti, ordinati in base al punteggio ottenuto.[3]

ricetta (image-to-recipe), in particolare sul dataset Recipe1M, ampiamente descritto nella sezione 2.1.

2.4.1 Modello

Gli autori, con questo progetto, propongono un modello semplificato, basato su encoder per testo e immagini ad alte prestazioni. In particolare, per la codifica delle ricette viene proposto un trasformatore gerarchico che opera sulle singole componenti, ovvero titolo, ingredienti e istruzioni. A differenza dei precedenti lavori che utilizzavano un LSTM, questo approccio permette di ottenere rappresentazioni

Figura 2.20: Esempi della procedura di generazione degli ingredienti. Partendo da sinistra sono riportati, l'immagine di input, gli ingredienti predetti, gli ingredienti predetti dal modello R_{I2LR} e gli ingredienti reali. Sono evidenziati in rosso gli ingredienti non presenti nella ricetta originale, mentre in blu quelli corretti.[3]

più forti per le ricette di input. Inoltre è stata introdotta una nuova funzione di perdita auto-supervisionata, che lavorando su coppie di singoli componenti delle ricette, permette di considerare sia i dati composti da immagine-ricetta sia solamente dalla ricetta. Questo aspetto ha permesso di utilizzare per l'addestramento del modello anche i dati di Recipe1M non associati a delle immagini, i quali non venivano considerati nei precedenti lavori, oppure usati solamente nella fase di pre-addestramento della rappresentazione del testo. Nella Fig. 2.21 è riportata la struttura del modello, nella quale la ricetta di input e l'immagine, se presente, attraverso gli opportuni encoder, vengono codificate in ϕ_{rec} e ϕ_{img} rispettivamente. Questi valori vengono poi incorporati in uno spazio comune attraverso le apposite funzioni di perdita L_{pair} e L_{rec}. Per tutte queste componenti vengono fornite delle descrizioni dettagliate nel seguito.

Encoder delle immagini

Lo scopo dell'encoder delle immagini è quello di trovare una funzione di mappatura che proietti l'immagine di input in uno spazio di rappresentazione condiviso con la ricetta, ϕ_{img}. Per fare ciò viene utilizzato il modello ResNet-50 [19], inizializzato con i pesi ricavati da un pre-addestramento sul dataset ImageNet [16]. In particolare, viene rimosso l'ultimo layer di classificazione e viene proiettato l'output del modello, nello spazio condiviso, tramite un layer lineare, ottenendo un risultato di dimensione 1024. Oltre alla soluzione sopra descritta, viene anche testato l'utilizzo di modello basato su ResNeXt [29], un'estensione di ResNet, che grazie all'introduzione di un nuovo modulo di aggregazione, basato su un inseme di filtri condivisi, consente una migliore capacità di generalizzazione e l'encoder Vision Transformer (ViT) [30] che applica i principi dei trasformatori, sviluppati in origine per il linguaggio testuale, alla codifica delle immagini.

Encoder delle ricette

Analogamente all'encoder delle immagini, quello delle ricette ha l'obiettivo di trovare una funzione di mappatura che proietti la ricetta in input in uno spazio di rappresentazione comune con l'immagine, ϕ_{rec}. Come accennato in precedenza, in altri lavori per la realizzazione di questa componente era utilizzato un LSTM, che veniva solitamente pre-addestrato o ottimizzato attraverso una funzione obiettivo per le coppie immagine-ricetta. In questo caso invece gli autori hanno scelto, grazie alle sue migliori performance nel NLP, di utilizzare il trasformatore [26] come componente di base. Nello specifico, ne vengono utilizzati tre distinti, che operano sulle tre componenti di una ricetta, ovvero, titolo, ingredienti e istruzioni. Lo scopo di questi elementi è quello di estrarre una codifica che contenga informazioni rilevanti per l'associazione tra l'immagine e la ricetta. Ogni elemento è composto da 2 layer di dimensione 512, ognuno con 4 attention heads. Nella parte sinistra della Fig. 2.22 è possibile visionare una rappresentazione dei trasformatori, $TR(.,\theta)$, dove θ rappresenta i parametri del modello, diversi per ognuna delle componenti. Siccome, gli ingredienti e le istruzioni di ogni ricetta non sono formati unicamente da una frase, come il titolo, ma sono composti da insiemi di frasi, per la loro codifica è stato utilizzato come encoder un trasformatore gerarchico, $HTR(.,\theta)$, ovvero una struttura formata da due livelli. Il primo livello è rappresentato da un primo trasformatore, $TR_{\text{L=1}}$, che elabora ogni frase che compone la lista di ingredienti o istruzioni di input, producendo una serie di codifiche a lunghezza fissa. Quest'ultime vengono concatenate e passate ad un secondo trasformatore, $TR_{\text{L=2}}$, che restituisce una rappresentazione per l'intera lista di dati in input. Uno schema di questa soluzione viene riportato nella parte destra della Fig. 2.22. Infine attraverso un layer lineare di dimensione 1024, le diverse codifiche del titolo, degli

ingredienti e delle istruzioni vengono concatenate e proiettate nello spazio condiviso con l'immagine, ottenendo cosi la rappresentazione finale della ricetta.

Funzione di perdita supervisionata per coppie di dati

Per l'addestramento del modello, viene definita la seguente funzione di perdita relativa alle coppie di dati, immagine-ricetta:

$$L_{\text{pair}} = L_{\text{bi}}(e_I^{n=i}, e_R^{n=i}, e_R^{n\neq i}, e_I^{n\neq i})$$

dove e_I e e_R rappresentano rispettivamente, la codifica dell'immagine e della ricetta, la notazione n=i indica che la coppia immagine-ricetta è presa dallo stesso elemento del dataset, mentre $n \neq i$ significa che la coppia è presa da due elementi diversi del dataset. La funzione di perdita L_{bi} viene calcolata per ogni elemento i di un batch, tramite una media delle perdite considerando tutti gli altri elementi j come negativi, ovvero non correlati con l'elemento i. Per maggiore chiarezza ne viene riportata la formulazione:

$$L_{\text{bi}}(a^{n=i}, b^{n=i}, b^{n\neq i}, a^{n\neq i}) = \frac{1}{B}\sum_{j=0}^{B} L'_{\text{bi}}(i,j)\delta(i,j)$$

Dove L'_{bi} rappresenta la bi-directional triplet loss function [31], una variante della triplet loss function, il cui scopo è posizionare in modo coerente gli elementi simili vicini e gli elementi diversi lontani. Di seguito sono riportate la funzione di perdita e le sue componenti:

$$L'_{\text{bi}}(i,j) = L_{\cos}(a^{n=i}, b^{n=i}, b^{n=j}) + L_{\cos}(b^{n=i}, a^{n=i}, a^{n=j})$$

$$L_{\cos}(a,p,n) = \max(0, c(a,n) - c(a,p) + m)$$

dove m rappresenta il margine, a rappresenta l'ancora, n e p gli elementi positivi e negativi, ovvero gli elementi correlati o non correlati rispettivamente con a e infine $cos(.)$ indica la funzione di similarità del coseno, già descritta in precedenza.

Funzione di perdita supervisionata per la sola ricetta

Per rimuovere il vincolo nell'utilizzo di soli dati composti sia da una ricetta che dalla relativa immagine, viene introdotta una funzione di perdita che prende in considerazione solamente gli elementi che compongo una ricetta. In particolare, questa funzione permette sia di avvicinare le componenti, come per esempio il titolo e gli ingredienti, se queste appartengono alla stessa ricetta, sia di allontanarle se appartengono a ricette differenti. Questa soluzione permette di produrre delle rappresentazioni delle ricette più forti e con una maggiore correlazione tra le diverse

componenti. Formalmente, viene introdotta una triplet loss function, descritta dalla seguente formulazione:

$$L'_{\text{rec}}(a,b) = L_{\text{bi}}(e_a^{n=i}, \hat{e}_{b\to a}^{n=i}, \hat{e}_{b\to a}^{n\neq i}, e_a^{n\neq i})$$

dove a e b rappresentano una qualsiasi componente della ricetta, mentre $e_{b\to a}$ indica la codifica di una delle componenti proiettata nello spazio di un'altra componente, attraverso l'utilizzo di un layer lineare, $g_{b\to a}(e_b)$, [32], [33]. Nella Fig. 2.23 sono riportate le sei possibili funzioni di proiezione. Questa perdita viene quindi calcolata per tutte le possibili combinazioni delle componenti prendendo poi il valore medio come risultato finale:

$$L_{\text{rec}} = \frac{1}{6}\sum_a \sum_b L'_{\text{rec}}(a,b)\delta(a,b)$$

Infine le due funzioni descritte nelle precedenti sezioni, 2.4.1 e 2.4.1, vengono combinate tramite una somma pesata dai parametri α e β, che assumono entrambi il valore 1.0 nel caso di coppie immagine-ricetta, mentre α viene impostato a 0.0 nel caso di dati non associati direttamente ad un'immagine.

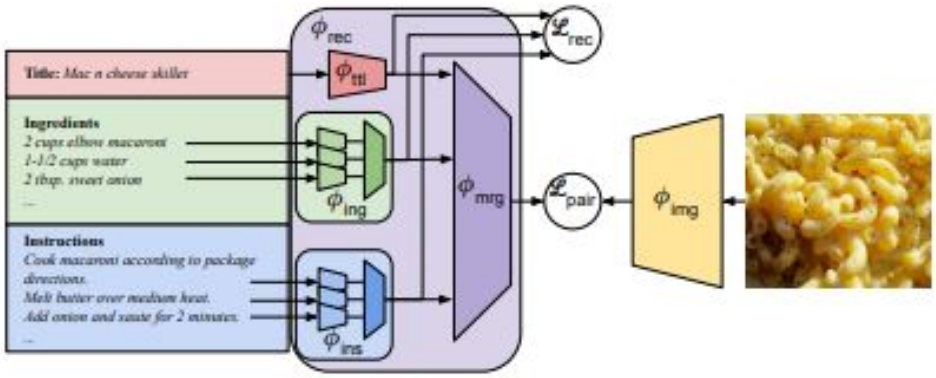


Figura 2.21: Struttura del modello.[4]

2.4.2 Metriche

Utilizzando il dataset Recipe1M per i diversi esperimenti, vengono scelte le stesse metriche dei lavori precedenti, ovvero la median rank (medR) ed il recall rate di K (R@K), dove K assume il valore di 1 (R@1), 5 (R@5) e 10 (R@10). Questi valori vengono calcolati su un insieme di N elementi, dove N assume valore di 1.000 e 10.000 e vengono ripetuti per 10 gruppi di campioni differenti scelti casualmente, prendendo poi come risultato finale una media dei valori ottenuti.

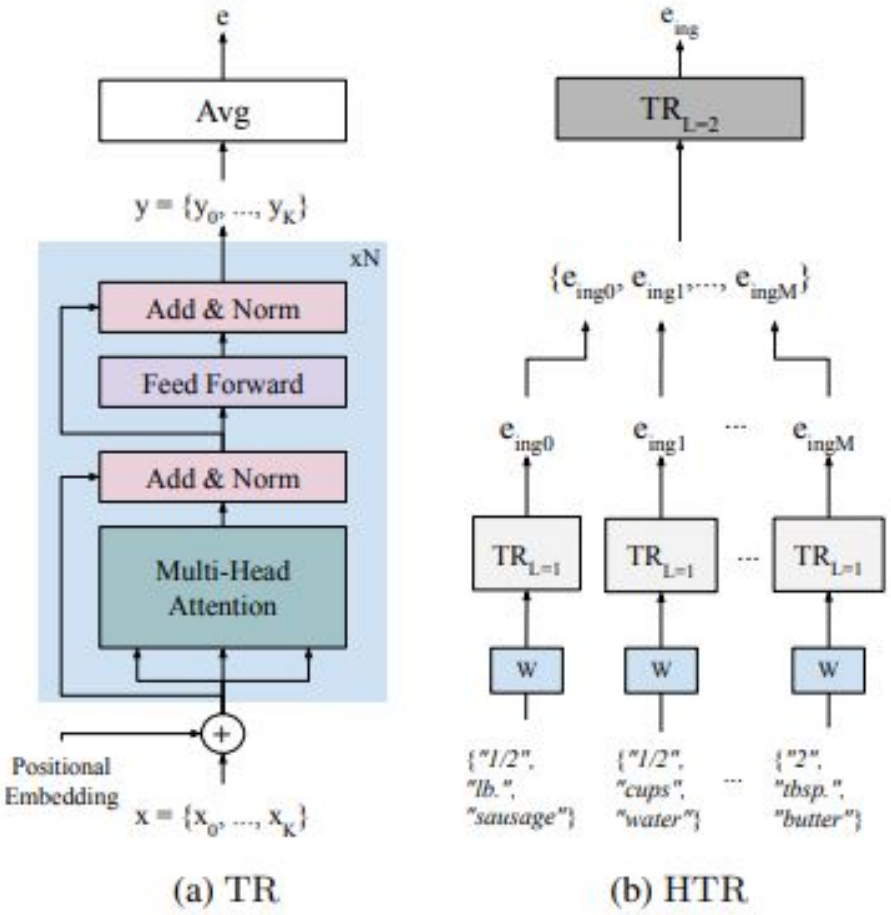


Figura 2.22: Rappresentazione dei trasformatori utilizzati. TR (a) indica la struttura del singolo trasformatore, mentre HTR (b) indica l'architettura del trasformatore gerarchico.[4]

2.4.3 Esperimenti e risultati

Gli esperimenti e l'addestramento del modello, come detto in precedenza, sono stati eseguiti sul dataset Recipe1M. In particolare sono state mantenute le divisioni originali, con 238.999 ricette per il training set, 51.119 per il validation set e 51.303 per il test set. Questi valori riguardano solamente le ricette associate con immagini, tuttavia, grazie all'introduzione di una funzione di perdita supervisionata per la sola ricetta, gli autori hanno considerato i restanti 482.231 dati che non erano associati direttamente ad un'immagine. Per l'implementazione e la realizzazione degli esperimenti è stato utilizzato il framework PyTorch [25]. Nello specifico, per

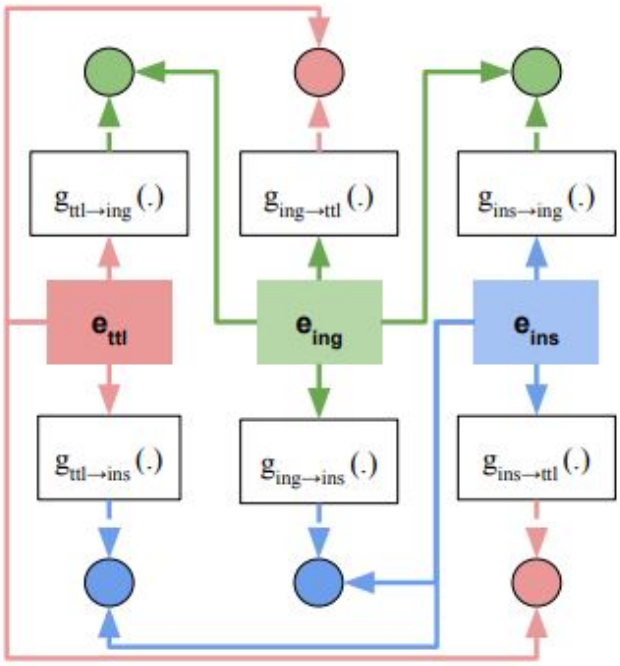


Figura 2.23: Funzione di perdita supervisionata per la sola ricetta. I cerchi colorati indicano le possibili combinazioni delle diverse componenti.[4]

l'addestramento del modello è stato scelto un ottimizzatore Adam [34] con un valore di learning rate iniziale pari a 10^{-4} per tutte le componenti, decrementato ogni 30 epoche di un fattore 0.1. Per le coppie di dati immagine-ricetta viene utilizzato un batch size di 128, mentre, visti i minori requisiti di memoria, per i dati formati dalle sole ricette il batch size viene incrementato a 256. Nella Fig. 2.25 sono riportati i risultati delle performance sul test set di Recipe1M, considerando 10.000 elementi, che mostrano il confronto tra le architetture che implementano un LSTM e le soluzioni proposte dagli autori. Dalla tabella è possibile notare come i trasformatori abbiano ottenuto risultati migliori, in particolare il modello che implementa il trasformatore gerarchico che ha ridotto il medR di 2.0 e incrementato l'R@1 di 4.6 punti. La tabella riportata nella Fig. 2.26 mostra il confronto tra diversi modelli dell'encoder dell'immagine, evidenziando come il ViT fornisca delle performance superiori rispetto a ResNet e sue varianti. Gli autori hanno anche eseguito degli esperimenti per la valutazione della rilevanza di ogni componente di una ricetta, mostrando come tutti gli elementi contribuiscono nella formazione di una rappresentazione più robusta. I risultati sono riportati nella Fig. 2.24. Infine, nella tabella della Fig. 2.27 viene riportato un confronto, tra i modelli proposti e alcuni lavori rilasciati negli anni precedenti ([1], [35], [36], [37], [38], [31], [39], [40]), da cui risultano chiare le migliori performance delle architetture proposte, sia per

il task di image-to-recipe sia per quello di recipe-to-image, rappresentando quindi lo state of the art per questi difficili processi. In conclusione nella tabella 2.1, per permettere una maggiore chiarezza nel confronto, sono riportati tutti i risultati dei diversi modelli analizzati.

	medR	**R1**	**R5**	**R10**
Ingredients only	8.2	19.1	42.8	54.3
Instructions only	15.0	12.6	32.2	43.3
Title only	35.5	6.0	18.7	28.1
Ingrs + Instrs	6.0	22.4	48.3	60.4
Title + Ingrs	6.0	22.1	47.7	59.8
Title + Instrs	10.5	15.9	38.4	50.2
Full Recipe	**5.0**	**24.4**	**51.4**	**63.4**

Figura 2.24: Performance del modello considerando diverse combinazioni delle componenti di una ricetta. Gli ingredienti risultano essere le componenti che forniscono la maggiore quantità di informazioni.[4]

	medR	**R1**	**R5**	**R10**
LSTM + avg	9.0	17.9	41.2	52.9
H-LSTM	7.0	19.8	44.8	57.2
Transformer + avg	7.0	20.2	45.2	57.3
H-Transformer	**5.0**	**24.4**	**51.4**	**63.4**

Figura 2.25: Performance delle architetture con trasformatori rispetto a quelle con LSTM.[4]

	medR	R1	R5	R10
DaC (ResNeXt-101) [13]	4.0	30.0	56.5	67.0
ResNet-50	4.0	27.9	56.4	68.1
ResNeXt-101	4.0	28.9	57.4	69.0
ViT	**3.0**	**33.5**	**62.2**	**72.9**

Figura 2.26: Confronto tra diversi modelli per la realizzazione dell'encoder dell'immagine.[4]

	1k								10k							
	image-to-recipe				recipe-to-image				image-to-recipe				recipe-to-image			
	medR	R1	R5	R10	medR	R1	R5	R10	medR	R1	R5	R10	medR	R1	R5	R10
Salvador et al. [37] ◦	5.2	24.0	51.0	65.0	5.1	25.0	52.0	65.0	41.9	-	-	-	39.2	-	-	-
Chen et al. [7]	4.6	25.6	53.7	66.9	4.6	25.7	53.9	67.1	39.8	7.2	19.2	27.6	38.1	7.0	19.4	27.8
Carvalho et al. [3] ◦	2.0	39.8	69.0	77.4	1.0	40.2	68.1	78.7	13.2	14.9	35.3	45.2	12.2	14.8	34.6	46.1
R2GAN [50]◦	2.0	39.1	71.0	81.7	2.0	40.6	72.6	83.3	13.9	13.5	33.5	44.9	12.6	14.2	35.0	46.8
MCEN [14]	2.0	48.2	75.8	83.6	1.9	48.4	76.1	83.7	7.2	20.3	43.3	54.4	6.6	21.4	44.3	55.2
ACME [42]◦	1.0	51.8	80.2	87.5	1.0	52.8	80.2	87.6	6.7	22.9	46.8	57.9	6.0	24.4	47.9	59.0
SCAN [43]	1.0	54.0	81.7	88.8	1.0	54.9	81.9	89.0	5.9	23.7	49.3	60.6	5.1	25.3	50.6	61.6
DaC [13]	-	-	-	-	-	-	-	-	5.9	24.4	49.4	60.5	-	-	-	
DaC [13]◦	1.0	55.9	82.4	88.7	-	-	-		5.0	26.5	51.8	62.6	-	-	-	-
Ours ($\mathcal{L}_{pair}$)	1.0	58.3	86.2	91.8	1.0	59.6	86.1	92.2	4.1	26.8	54.7	66.5	4.0	27.6	55.1	66.8
Ours ($\mathcal{L}_{pair} + \mathcal{L}_{rec}$)	1.0	59.1	86.9	92.3	1.0	59.1	87.0	92.7	4.0	27.3	55.4	67.3	4.0	27.8	55.6	67.3
Ours ($\mathcal{L}_{pair} + \mathcal{L}_{rec}$)◦	1.0	**60.0**	**87.6**	**92.9**	1.0	**60.3**	**87.6**	**93.2**	4.0	**27.9**	**56.4**	**68.1**	4.0	**28.3**	**56.5**	**68.1**

Figura 2.27: Confronto tra i modelli proposti e lavori rilasciati negli anni precedenti.[4]

Tabella 2.1: Tabella di confronto tra i diversi modelli analizzati

	Recipe1M						*Recipe1M+*			
	MedR	**R@1**	**R@5**	**R@10**	**IoU**	**F1**	**MedR**	**R@1**	**R@5**	**R@10**
image-to-recipe										
Recipe1M (2.2)	5.1	24.0%	52.0%	64.0%	/	/	13.6	15.0%	35.0%	46.0%
Recipe1M+ (2.2)	5.7	21.0%	49.0%	62.0%	/	/	8.6	17.0%	42.0%	54.0%
transformer (2.4)	1.0	60%	87.6%	92.9%	/	/	/	/	/	/
inverseCooking (2.3)	/	/	/	/	32.11%	48.61%	/	/	/	/
recipe-to-image										
Recipe1M (2.2)	4.8	27.0%	54.0%	65.0%	/	/	11.9	17.0%	38.0%	48.0%
Recipe1M+ (2.2)	4.6	26.0%	54.0%	66.0%	/	/	6.8	21.0%	46.0%	58.0%
transformer (2.4)	1.0	60.3%	87.6%	93.2%	/	/	/	/	/	/

Capitolo 3

Soluzioni proposte

In questo capitolo vengono descritti i diversi approcci e esperimenti realizzati, con particolare attenzione verso l'implementazione, i risultati e i confronti tra le diverse metodologie.

3.1 Approccio 0

In questa sezione in particolare vengono descritti i diversi test ed esperimenti effettuati per la verifica dell'efficacia del modello img2recipe [2], utilizzato partendo dal miglior checkpoint messo a disposizione dagli sviluppatori.

3.1.1 Implementazione

Inizialmente sono stati condotti degli studi per comprendere al meglio la struttura del dataset Recipe1M+, il funzionamento del modello e del codice messo a disposizione dagli autori. Grazie a questo processo è stato possibile realizzare un insieme di script python che hanno permesso di applicare il processo di image-to-recipe ad una singola immagine, funzionalità non introdotta nel codice originale, che prevedeva unicamente l'utilizzo di un intero test set per la valutazione delle ricette, che si è però rilevata molto utile per testare l'efficacia del modello. Gli script realizzati, in particolare, sono:

- *image2embedding*, che partendo da un'immagine di input restituisce la rappresentazione di essa nello spazio condiviso tra ricetta e immagine e la salva in un file pickle temporaneo (.pkl);
- *recipe2embedding*, che permette l'estrazione della rappresentazione nello spazio condiviso di tutte le ricette presenti nel dataset passato come parametro di input, salvando anch'esse in un file pickle;

- *recipe_loader*, una versione semplificata del data loader realizzato dagli autori, che permette l'estrazione dei dati riguardanti le ricette, tralasciando invece quelli relativi alle immagini ad esse associate;

- *rank*, script che effettua il confronto tra la rappresentazione dell'immagine, ricavata precedentemente, e le rappresentazioni delle ricette, producendo un vettore di similarità, nel quale ogni elemento indica la correlazione tra la singola ricetta e l'immagine;

- *image2recipe*, realizzato per automatizzare il processo di image-to-recipe, sia per una singola immagine, fornendo unicamente l'indirizzo di quest'ultima come input, sia per un gruppo di immagini per le quali dovrà essere fornito l'indirizzo della cartella nella quale sono contentute;

3.1.2 Esperimenti e Risultati

Tramite il codice descritto nella sezione precedente sono stati condotti diversi esperimenti per la valutazione delle performance del modello messo a disposizione dagli autori. Nello specifico esso è stato testato su singole immagini, ottenute tramite Google, prendendo in considerazione le 10 ricette con l'indice di similarità più elevato, confrontando per ognuna, il titolo e gli ingredienti. Tramite questi esperimenti è stato possibile verificare che il modello è in grado di associare con buona precisione immagini che raffigurano pietanze di largo consumo e diffusione in tutto il mondo, come per esempio sushi, hamburger o semplici patatine fritte, associandogli quindi ricette nelle quali il numero di ingredienti corretti risulta molto elevato. Tuttavia queste ottime performance non vengono riscontrate nel riconoscimento di piatti meno famosi e conosciuti, con i quali il modello produce delle associazioni che nella maggior parte dei casi non coincidono né per ingredienti né per titolo ai valori reali. Vista quindi la necessità di ottenere un indice di precisione elevato soprattutto per pietanze appartenenti alla cultura culinaria italiana, sono state raccolte un insieme di ricette, tramite uno strumento di data mining chiamato ParseHub [7], partendo da famosi siti di cucina italiana come, *GialloZafferano* [41], *Fatto in casa da Benedetta* [42] e *Misya* [43]. Per ogni ricetta sono stati raccolti, il titolo, immagine associata e gli ingredienti, caratterizzati da nome e quantità. Attraverso questi dati è stato possibile verificare le performance di img2recipe solamente su piatti italiani, mostrando come per un numero ridotto di ricette, come per esempio *Pizza*, *Lasagna*, *Pasta alla carbonara* o ancora *Pasta all'amatriciana*, la precisione sia elevata e permetta una corretta associazione sia per titolo che per ingredienti, mentre per la maggior parte dei piatti, nonostante la loro fama, le associazioni risultino del tutto errate, come nel caso dell'*Arancino*, della *Mozzarella in carrozza* e dei *Panzerotti*. Si può quindi concludere che il modello di img2recipe, nonostante sia stato addestrato su un numero estremamente

elevato di ricette non sia adatto, in questa versione almeno, a riconoscere con elevata precisione le pietanze tipiche della cucina italiana.

3.2 Approccio 1

3.2.1 Nuovo dataset

Considerando i risultati ottenuti negli esperimenti precedentemente descritti, nel tentativo di migliorare la precisione nel riconoscimento dei piatti tipici italiani, è stato realizzato un nuovo dataset. Il quale contiene quasi 5000 (4928) ricette tipiche della cultura culinaria italiana, raccolte attraverso lo strumento ParseHub, già utilizzato per la raccolta delle ricette nella sezione precedente, partendo da famosi siti di cucina come, *GialloZafferano* [41], *Cookaround* [44] e *Misya* [43]. Per ogni ricetta, sono stati collezionati, il titolo, la portata alla quale appartiene il piatto, l'URL della ricetta, l'URL dell'immagine associata, la lista degli ingredienti, composta da nome, quantità e unità di misura per ogni elemento, la lista delle istruzioni per la realizzazione della pietanza e infine un identificativo univoco. Nello specifico, per le immagini si è scelto di considerare l'URL invece che scaricarle direttamente in locale, a causa dell'elevato costo in termini di memoria che avrebbe rappresentato. Di conseguenza durante gli esperimenti, l'accesso alle diverse immagini avviene tramite il modulo *requests* [45] di python, che permette di effettuare richieste HTTP/1.1 in maniera estremamente semplice, specificando solamente l'indirizzo richiesto. Le istruzioni di preparazione invece sono state raccolte per permettere la corretta interazione con il modello img2recipe, il quale, come specificato nel capitolo precedente 2.2, le utilizza insieme all'elenco di ingredienti per ottenere la rappresentazione della ricetta. Nel caso degli esperimenti che coinvolgono il modello inverseCooking invece, questi dati non verranno utilizzati, prendendo in considerazione solamente il titolo e gli ingredienti. Il dataset è stato inoltre suddiviso in tre parti. La prima, contentene circa il 70% delle ricette destinate al training (3348), mentre il restante 30% è stato suddiviso in modo equo tra validation e test set, (780 e 784 rispettivamente). Di seguito vengono riportati una serie di grafici che riportano alcune statistiche relative a questa nuova collezione di dati. In particolare nella Fig. 3.1 è riportata, per le partizioni di training, validation e test, la distribuzione delle ricette in base alla portata di appartenenza. Il grafico mostra una prevalenza di ricette appartenenti ai *primi*, *secondi* e *dolci*, che risultano molto bilanciate tra loro, inoltre tutte le portate che presentavano un numero di ricette estremamente basso sono state raggruppate all'interno della categoria *altri*. Nel grafico riportato nella Fig. 3.2 invece è mostrata la distribuzione del numero di ingredienti che compongono le diverse ricette, dal quale è possibile notare che la maggior parte dei piatti è composto da un numero di ingredienti differenti compreso tra 4 e 15. Le Fig. 3.3 e 3.4 mostrano rispettivamente, la distribuzione del numero

di istruzioni per ogni ricetta e la distribuzione delle lunghezze delle istruzioni stesse, dalle quali è possibile notare come la gran parte delle ricette è descritta da un numero di istruzioni non superiore a 11 e che la lunghezza delle istruzioni stesse non supera le 75 parole nella maggior parte dei casi.

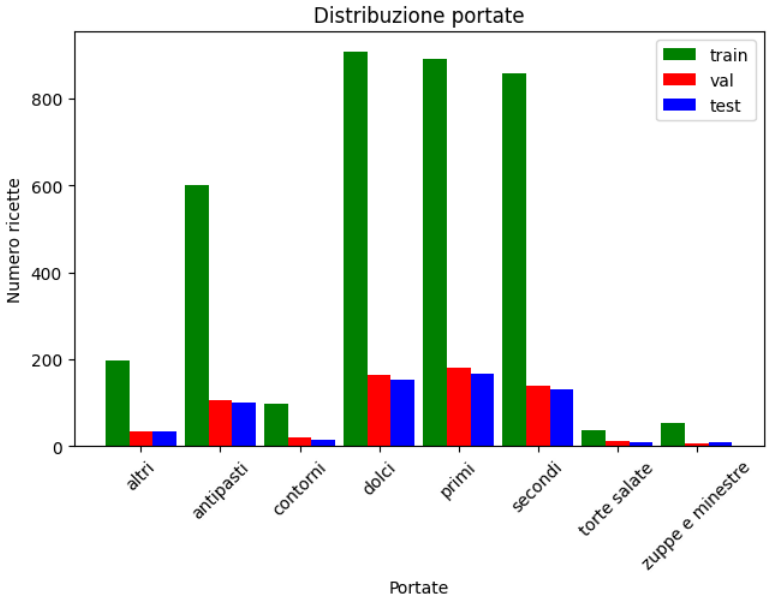


Figura 3.1: Distribuzione delle ricette secondo la portata di appartenenza per le partizioni di training, validation e test

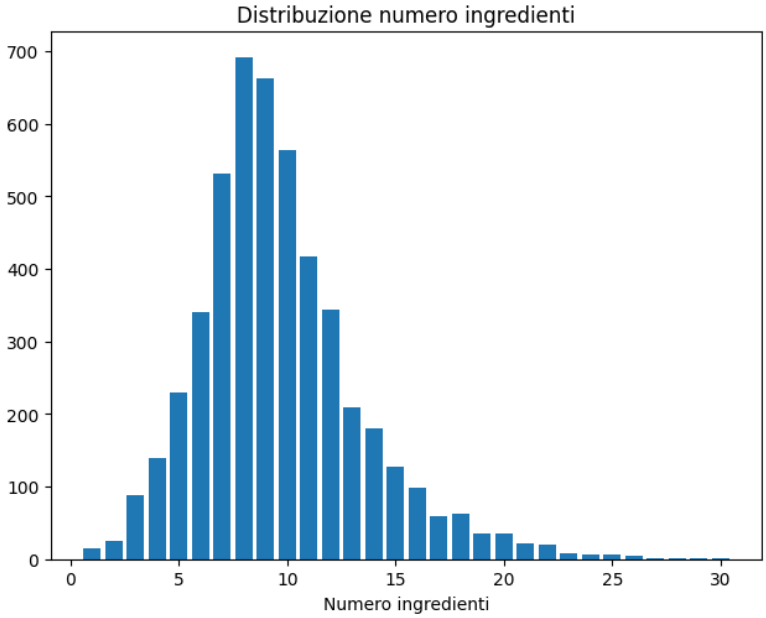


Figura 3.2: Distribuzione del numero di ingredienti differenti per ogni ricetta

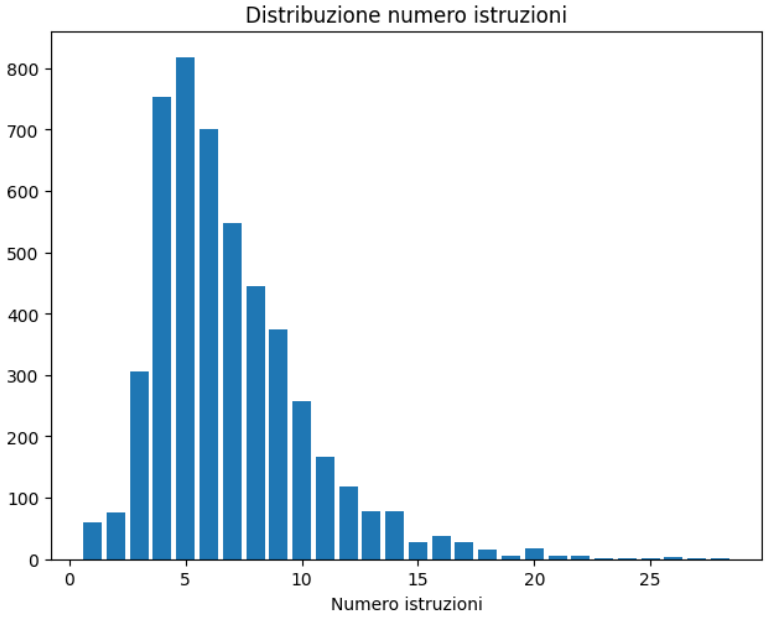


Figura 3.3: Distribuzione del numero di istruzioni per ogni ricetta

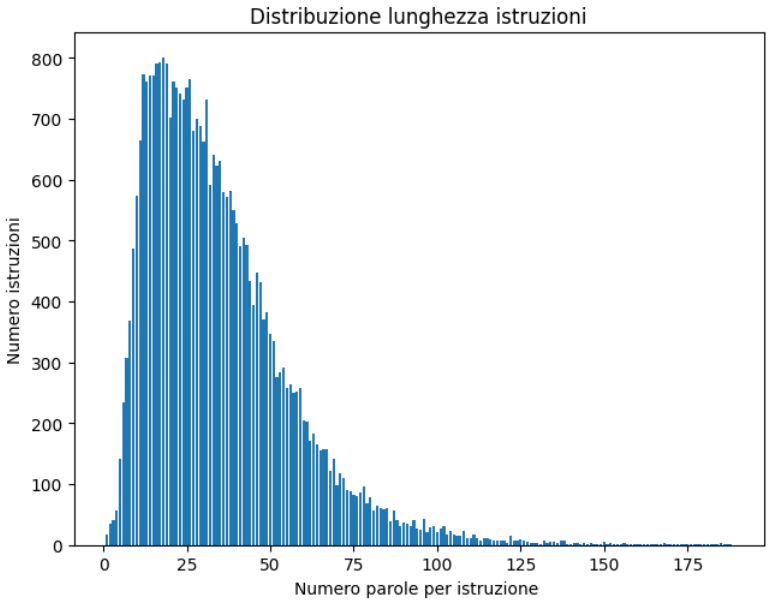


Figura 3.4: Distribuzione del numero di parole che compongono le diverse istruzioni di ogni ricetta

3.2.2 Esperimenti con img2recipe

Implementazione

Per la realizzazione degli esperimenti con il modello img2recipe utilizzando il dataset sopra descritto è stato necessario, in un primo momento, eseguire una serie di operazioni per adattare i nuovi dati alla struttura realizzata dagli autori. Le diverse fasi di questo pre-processamento vengono descritte nello specifico nel seguito:

- **Pulizia degli ingredienti**: in questa fase è stato necessario ripulire gli ingredienti presenti all'interno delle ricette raccolte, estraendo quindi solamente il nominativo e tralasciando le quantità e le unità di misura. Il nominativo inoltre è composto da una sola parola, di conseguenza ingredienti come *l'olio d'oliva*, saranno rappresentati come *olio_oliva*. Per maggiore chiarezza, partendo dal testo di ogni ingrediente, come per esempio, *un pizzico di sale*, si ottiene solamente *sale*. Il risultato di questa operazione viene salvato in un file, indicando per ogni ricetta l'identificativo univoco e una lista di tutti gli ingredienti ripuliti;

- **Creazione dizionario**: in questa fase invece, tramite alcuni script python realizzati dagli autori è stato possibile estrarre il testo dalle ricette utilizzate per l'addestramento del modello, impegnandolo poi come input per un modello

di rappresentazione word2vec, descritto precedentemente nella sezione 2.2.1, che ha permesso la creazione di un dizionario, contenuto nel file *vocab.bin*, che permette l'associazione tra le parole che compongono una ricetta e degli identificativi numerici univoci;

- **Estrazione delle categorie semantiche** Per l'implementazione del processo di regolarizzazione semantica, descritto nella sezione 2.2.1, vengono estratte delle categorie semantiche a partire dai titoli delle ricette. Questo viene effettuato tramite l'utilizzo di *bigrams* del modulo *nltk* di python, ([46], [47]), che permette di estrarre le coppie di parole che vengono associate con maggiore frequenza all'interno di un testo. In termini più semplici, a partire dai titoli delle ricette vengono estratte le coppie di parole che compaiono maggiormente, utilizzandole poi come label per il processo di regolarizzazione. Negli esperimenti condotti vengono anche aggiunte le categorie presenti nel dataset food-101 [14] (riportate nel dettaglio in appendice A), precedentemente tradotte in italiano, ottenendo un valore finale di 52 label, che permettono di classificare circa il 14% delle ricette, mentre al restante 86% viene assegnata la label *background*;

- **Codifica degli ingredienti** Attraverso il dizionario ottenuto precedentemente gli ingredienti di ogni ricetta vengono codificati in una lista di numeri interi, che rappresentano semplicemente i valori numerici univoci per ogni singolo ingrediente. Come dimensione massima di questa lista viene scelto come valore 25, in modo da poter rappresentare correttamente tutte le ricette raccolte;

- **Rappresentazione delle istruzioni** Per ottenere una rappresentazione numerica per le istruzioni di ogni ricetta è stato necessario utilizzare un modello per l'estrazione dei cosiddetti *Skip-Thought vectors*, presentati nel capitolo precedente 2.2.1. Questo modello [48] è stato addestrato sul testo estratto dalle ricette di tutte le partizioni e produce come output un insieme di vettori numerici che rappresentano le diverse istruzioni di una ricetta. In particolare il modello, se addestrato correttamente, è in grado di produrre per istruzioni e preparazioni simili tra loro dei valori numerici analoghi. Come dimensione massima di questo insieme di vettori è stato scelto come valore 25 , mentre come numero massimo di parole per ogni istruzione 125, per rappresentare al meglio ogni ricetta;

- **Creazione database lmdb** Infine grazie ad un apposito script tutti i dati elaborati nelle fasi precedenti vengono uniti in un unico database in formato lmdb, dove per ogni ricetta è presente, il titolo, i vettori degli ingredienti e delle istruzioni codificate, l'URL dell'immagine associata e l'identificativo univoco della ricetta;

Metriche

Per la valutazione delle performance del modello vengono impiegate le metriche, median rank (MedR) e recall rate (R@K), in particolare R@1, R@5 e R@10, descritte nel capitolo precedente 2.2.2. Per ottenere un corretto adattamento al dataset utilizzato, questi valori sono calcolati rispetto a tutti gli elementi del validation set, ovvero 780 e rispetto ad un sottoinsieme di 500 coppie immagine-ricetta. Le valutazioni vengono effettuate 10 volte, su sottoinsiemi scelti casualmente, come negli esperimenti condotti dagli autori, considerando poi i valori medi.

Risultati

Sono state condotte due tipologie di esperimenti differenti: nel primo, per il training del modello vengono utilizzati unicamente i dati del nuovo dataset delle ricette italiane; il secondo invece, è stato addestrato partendo dal miglior checkpoint fornito dagli autori, cercando quindi di sfruttare un pre-addestramento sul datatset Recipe1M, per ottenere migliori performance. Nello specifico, per entrambi i modelli sono stati realizzati dei grafici per la valutazione delle prestazioni, che vengono riportati nel seguito. Si sottolinea il fatto che il numero di punti utilizzati nei grafici per il validation set risulta molto inferiore a quello del training set, questo perché durante l'addestramento la valutazione sul validation set non è stata effettuata ogni singola epoca, ma ogni 10 e inoltre la durata del secondo esperimento risulta inferiore alla prima, a causa delle peggiori performance riscontrate. Nella Fig. 3.5 e 3.6 viene mostrato rispettivamente l'andamento della funzione di perdita coseno, *(cosine loss)*, sul training set e sul validation set. Analizzando l'andamento di questi dati è possibile notare come durante l'avanzamento delle diverse epoche i due valori decrementino, mostrando la capacità del modello di produrre rappresentazioni sempre più efficaci per la corretta associazione tra immagine e ricetta. Tuttavia risulta evidente come nel validation set, dopo l'epoca 100, questa discesa si arresti, producendo dei valori che si aggirano sempre intorno a 0.63, mentre nel trainig set il valore decresce costantemente. Questo fenomeno potrebbe essere attribuito ad un problema di overfitting, descritto con maggiore dettaglio nella sezione successiva 3.2.4. Le Fig 3.7 e 3.8 riportano invece l'andamento della *image loss* e della *recipe loss*, rispettivamente, utilizzate per la regolarizzazione semantica del modello, nelle quali è possibile riscontrare lo stesso fenomeno evidenziato precedentemente. La Fig. 3.9 mostra invece i valori di median rank (medR), ottenuti con l'avanzare dell'esperimento, calcolati rispetto a tutto il validation set e ad un sottoinsieme di esso formato da 500 elementi. Dal grafico risulta evidente come la valutazione su un numero inferiore di elementi produca delle prestazioni superiori, in quanto il modello ha meno probabilità di associare l'immagine alla ricetta errata. Infine, la Fig. 3.10 riporta i valori di recall rate (R@K), calcolati anch'essi sull'intero set di validazione e su un sottoinsieme di esso, composto da 500 elementi. Questi

valori mostrano la crescente capacità del modello di associare correttamente la ricetta all'immagine di input, posizionandola nel 10% dei casi nella prima posizione e nel 45% nelle prime 10. I dati riportati fino a questo momento si riferiscono all'esperimento che coinvolge il modello addestrato unicamente su ricette italiane, mentre le Fig. 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, mostrano le prestazioni ottenute partendo dal checkpoint fornito dagli autori [2]. Da questi grafici risultano chiare le migliori performance ottenute nell'esperimento precedente, evidenziando come, il pre-addestramento sul dataset Recipe1M, al contrario di quanto si possa pensare, non fornisce un aiuto nel riconoscimento delle nuove ricette, ma bensì, rende più difficile per il modello creare delle rappresentazioni efficaci per la corretta associazione immagine-ricetta. Questo effetto può essere attribuito ad un problema di cambiamento di dominio, comunemente denominato *domain shift*, che come nel caso precedente, viene descritto con maggiore dettaglio nella sezione seguente 3.2.4. Infine nel tabella 3.1 vengono riportati i migliori risultati ottenuti dai due esperimenti, enfatizzando ulteriormente le migliori prestazioni ottenute dal primo esperimento descritto, aggiungendo inoltre le performance ottenute sul test set, che risultano essere molto simili a quelle riscontrate sul validation set.

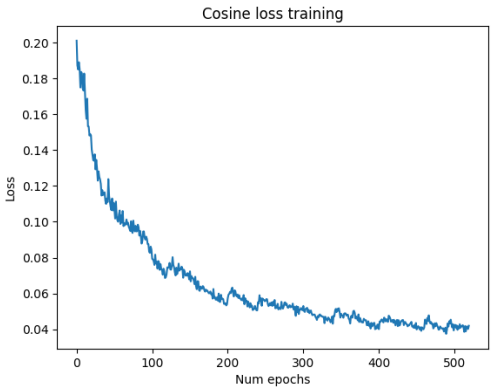


Figura 3.5: Andamento della cosine loss, sul training set, durante l'addestramento del modello unicamente su ricette italiane

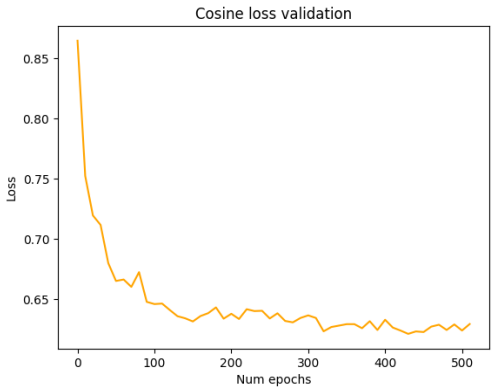


Figura 3.6: Andamento della cosine loss, sul validation set, durante l'addestramento del modello unicamente su ricette italiane

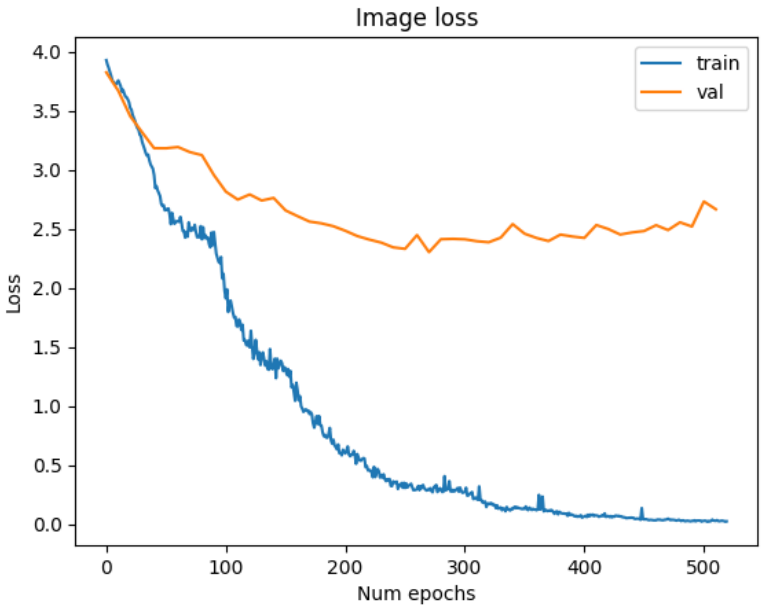


Figura 3.7: Andamento della image loss, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane

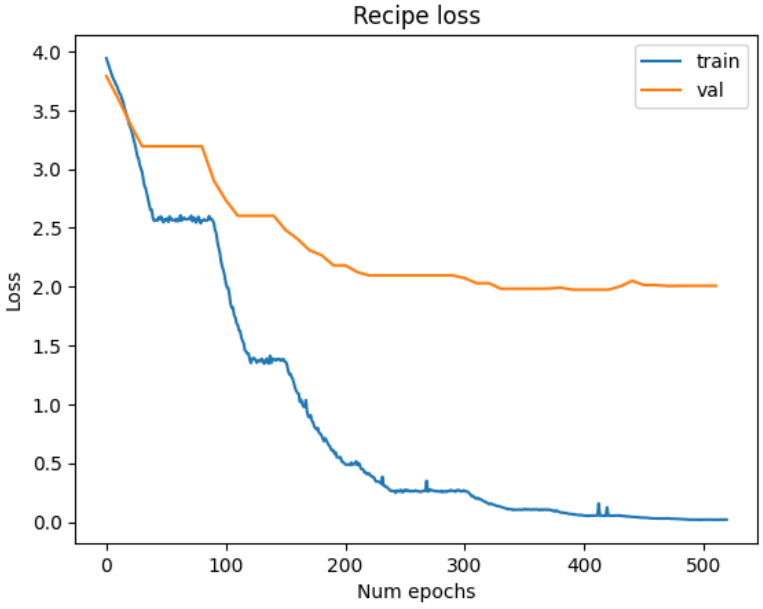


Figura 3.8: Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane

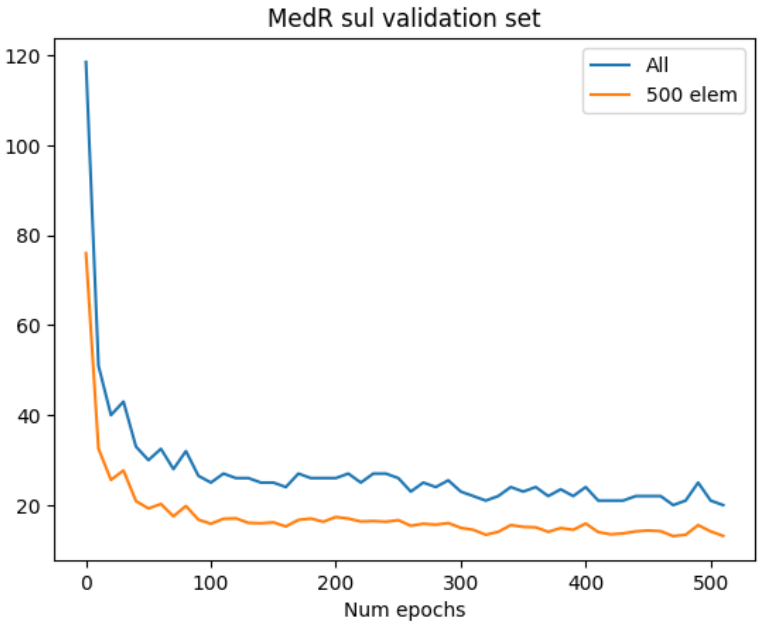


Figura 3.9: Andamento della MedR, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane

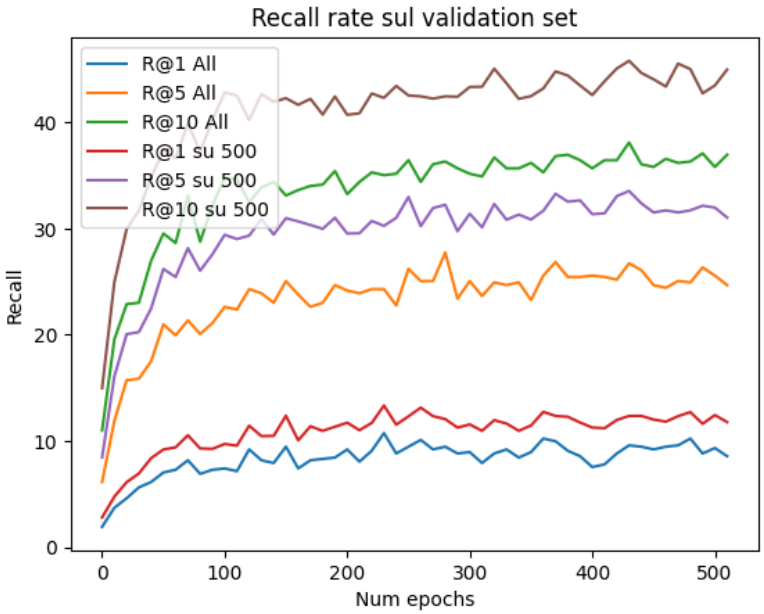


Figura 3.10: Andamento del recall rate, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette

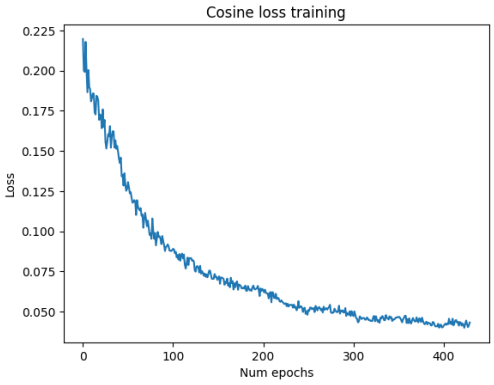


Figura 3.11: Andamento della cosine loss, sul training set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori

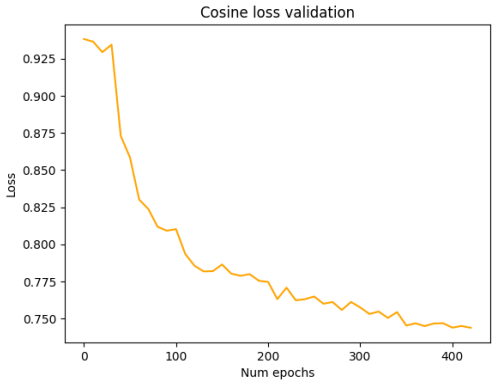


Figura 3.12: Andamento della cosine loss, sul validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori

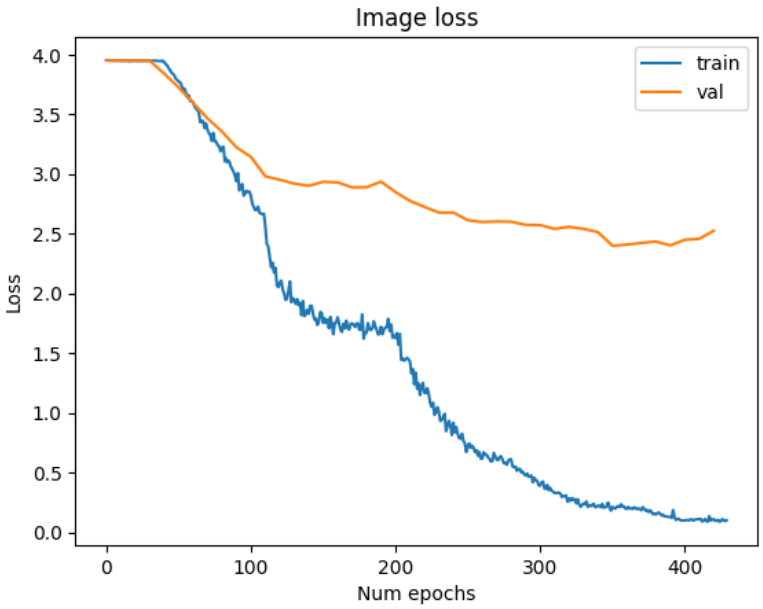


Figura 3.13: Andamento della image loss, sul training e validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori

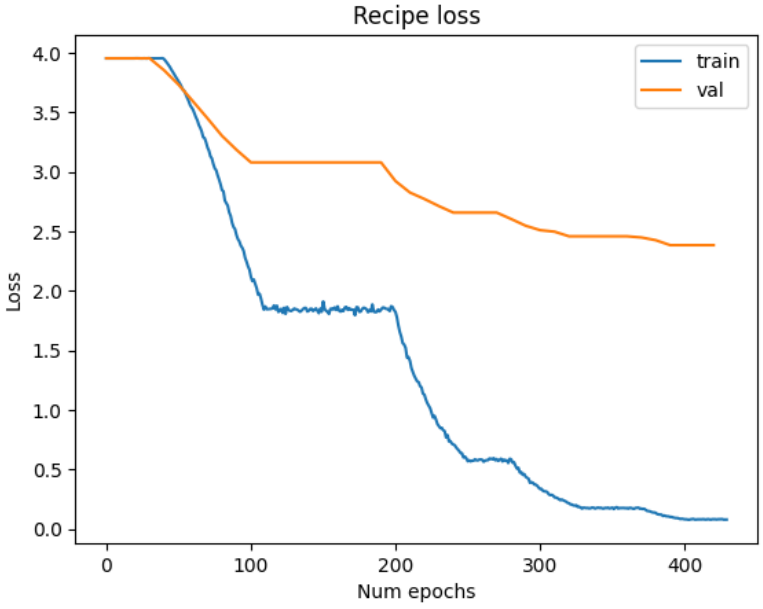


Figura 3.14: Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori

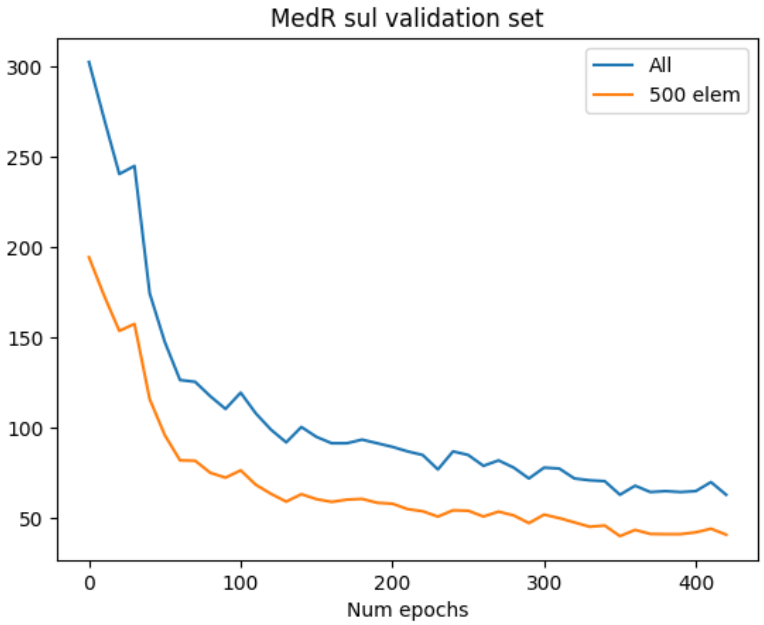


Figura 3.15: Andamento della MedR, sul training e validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori

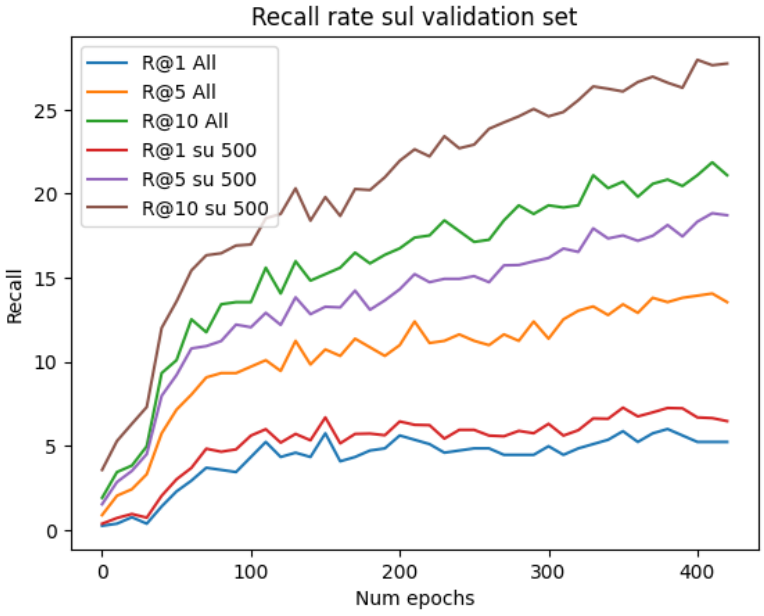


Figura 3.16: Andamento del recall rate, sul training e validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette

Tabella 3.1: Risultati degli esperimenti sul modello img2recipe

	From 0			**From checkpoint**		
	Train	**Val**	**Test**	**Train**	**Val**	**Test**
Cosine Loss	0.0390	0.6292	0.6326	0.0372	0.7435	0.7482
Image Loss	0.0246	2.6647	2.3579	0.0942	2.5242	2.1938
Recipe Loss	0.0204	2.0079	1.7866	0.0793	2.3829	2.1045
MedR (Tot)	/	20.0	23.0	/	62.5	67.0
R@1 (Tot)	/	8.5%	9.6%	/	5.2%%	5.5%
R@5 (Tot)	/	24.6%	25%	/	13.5%	15.0%
R@10 (Tot)	/	36.8%	36.6%	/	21.1%	22.2%
MedR (500 elem)	/	13.15	15.0	/	40.6	42.5
R@1 (500 elem)	/	11.7%	12.5%	/	6.5%	7.2%
R@5 (500 elem)	/	30.9%	30%	/	18.7%	19.5%
R@10 (500 elem)	/	44.9%	42.9%	/	27.9%	26.8%

3.2.3 Esperimenti con inverseCooking

Implementazione

Come per gli esperimenti descritti in precedenza, anche per la verifica delle prestazioni di inverseCooking è stata necessaria una prima fase di studio della struttura del modello e del codice fornito dagli autori, seguita da una serie di operazioni di pre-processamento dei dati, per garantire un corretto adattamento al modello. Di seguito sono descritte con maggiore dettaglio le operazioni appena citate:

- **Pulizia degli ingredienti**: come nel caso degli esperimenti precedenti su img2recipe, è stato necessario ripulire gli ingredienti mantenendo unicamente il nome e tralasciando la quantità e l'unità di misura. Questo processo è analogo a quello descritto nella sezione precedente;

- **Creazione del dizionario degli ingredienti**: tramite un opportuno script fornito dagli autori è stato possibile creare un dizionario che contenesse tutti gli ingredienti presenti nelle ricette fornite come input, applicando inoltre una clusterizzazione ed eliminazione dei plurali dalle parole estratte, riducendo così il numero totale di ingredienti considerati. Per maggiore chiarezza, tra i molti ingredienti forniti si trovano *fettine_di_manzo*, *filetto_di_manzo* e *carne_di_manzo* che vengono considerate tutte semplicemente come *manzo*;

- **Creazione del dizionario delle parole**: attraverso lo script descritto precedentemente viene inoltre realizzato un dizionario che comprende tutte le parole che compongono le ricette presenti nel dataset di input, considerando

quindi, titolo, ingredienti e istruzioni. Per l'estrazione delle singole parole viene utilizzato il package *tokenize* del modulo *nltk* di python [49];

- **Creazione del database**: Uno volta eseguite le operazioni precedenti è stato possibile creare un database che contenesse, per ogni ricetta, un titolo, una lista di ingredienti e una di istruzioni ripulite, l'immagine associata e una lista delle singole parole che compongono la ricetta;

- **Modifica del data loader**: infine, siccome le immagini non sono state salvate direttamente in locale, ma vengono acquisite tramite richieste HTTP/1.1 sfruttando il modulo *requests* di python sono stati apportati dei cambiamenti al data loader fornito dagli autori, per permettere un corretto caricamento delle immagini durante l'addestramento del modello

Metriche

Per la valutazione delle performance del modello vengono impiegate le metriche, Intersection over Union (IoU), cardinality prediction error ed F1, descritte nel capitolo precedente 2.3.2. In particolare, visto il minore interesse nella valutazione delle istruzioni generate per la preparazione dei piatti, rispetto che a quelle degli ingredienti, gli esperimenti si concentrano unicamente sulla verifica di quest'ultimi.

Risultati

Come per il caso precedente sono state condotte due tipologie di esperimenti differenti: uno in cui il training del modello sfrutta il miglior checkpoint fornito dagli autori come pre-addestramento, per ottenere una maggiore precisione; un secondo, addestrato utilizzando unicamente i dati del nuovo dataset delle ricette italiane. Nello specifico, per entrambi i modelli sono stati realizzati dei grafici per la valutazione delle prestazioni, che vengono riportati nel seguito. Si sottolinea che il numero di punti utilizzati nei grafici per il validation set è equivalente a quelli del training set, questo perché durante l'addestramento la valutazione sul validation set viene effettuata ogni singola epoca. Inoltre, viste le performance inferiori del modello sottoposto a pre-addestramento, la durata dell'esperimento di quest'ultimo risulta inferiore, come per il modello img2recipe. Nella Fig. 3.17 viene mostrato l'andamento della funzione di perdita totale (loss), che nel caso del trainng set decrementa in maniera costante, mentre per il validation set, dopo una riduzione nelle prime epoche, si riscontra un leggero aumento, per poi stabilizzarsi su valori intorno al 75. Questa grande differenza è probabilmente dovuta ad un problema di overfitting, riscontrato anche durante gli esperimenti su img2recipe, che viene approfondito maggiormente nella sezione successiva (3.2.4). La Fig. 3.18 riporta invece l'avanzamento della funzione di perdita riferita agli ingredienti (ingredient

loss), ovvero una funzione che regola l'andamento dell'addestramento in base alla precisione ottenuta sulla generazione degli ingredienti. Il grafico risulta molto simile a quello precedente, questo perché la funzione di perdita totale è calcolata come somma pesata di diverse loss, tra cui, l'*ingredient loss*, la *recipe loss*, la *eos loss* e la *cardinality penality*, descritte con maggiore dettaglio nel capitolo precedente 2.3.1, per le quali vengono utilizzati i rispettivi pesi di 1000.0, 0, 1.0 e 1.0. Per la *recipe loss* viene utilizzato un peso pari a 0, in quanto non si è interessati alla valutazione delle istruzioni generate e di conseguenza si vuole evitare che queste possano influenzare il training del modello. Nelle Fig. 3.19 e 3.20 vengono invece mostrati i risultati ottenuti in termini di intersection over union (IoU) e cardinality prediction error, rispettivamente. Per entrambi nuovamente è possibile riscontrare una grande differenza tra training e validation set, da attribuire come detto precedentemente al problema di overfitting. Osservando l'Iou sul training set è possibile notare come nei picchi più alti ottenga valori superiori al 60%, risultato ottimo nel caso si riuscisse a ridurre la distanza con il validation set. Infine, la Fig. 3.21 riporta i valori della metrica F1 sul validation set, evidenziando come i valori massimi siano ottenuti nelle prime epoche, in corrispondenza del decremento iniziale riscontrato negli altri grafici. I dati riportati fino a questo momento si riferiscono all'esperimento che coinvolge il modello addestrato unicamente su ricette italiane, mentre le Fig. 3.22, 3.23, 3.24, 3.25, 3.26, mostrano le prestazioni ottenute partendo dal checkpoint fornito dagli autori [3]. Da questi grafici è possibile notare come la loss e l' ingredient loss ottengano valori migliori rispetto a quelli dell'esperimento precedente, tuttavia l'Iou, il cardinality prediction error e l'F1 ottengono prestazioni decisamente inferiori. Questo effetto può essere attribuito ad un problema di cambiamento di dominio, comunemente denominato *domain shift*, riscontrato anche negli esperimenti su img2recipe e trattato con maggior approfondimento nella sezione successiva 3.2.4. Inoltre, da questi grafici è possibile osservare una riduzione della differenza tra training e validation set, che caratterizzava l'esperimento precedente.

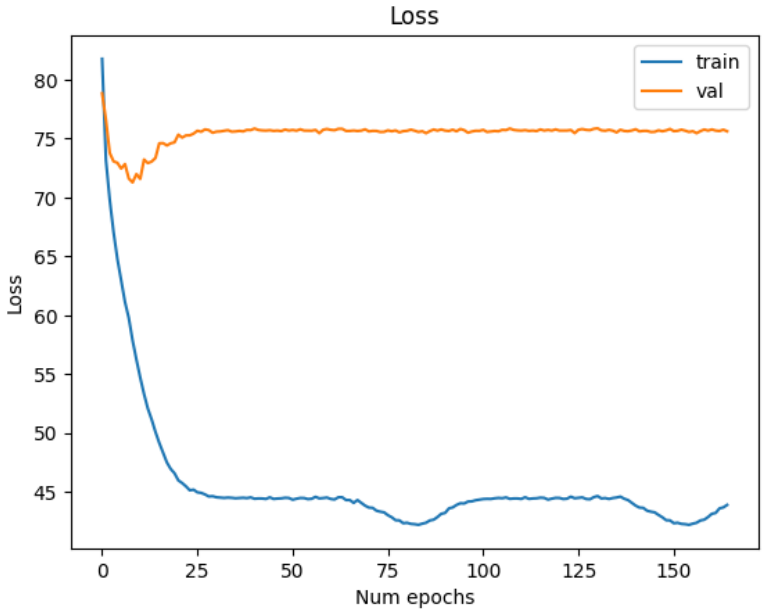


Figura 3.17: Andamento della Loss, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane

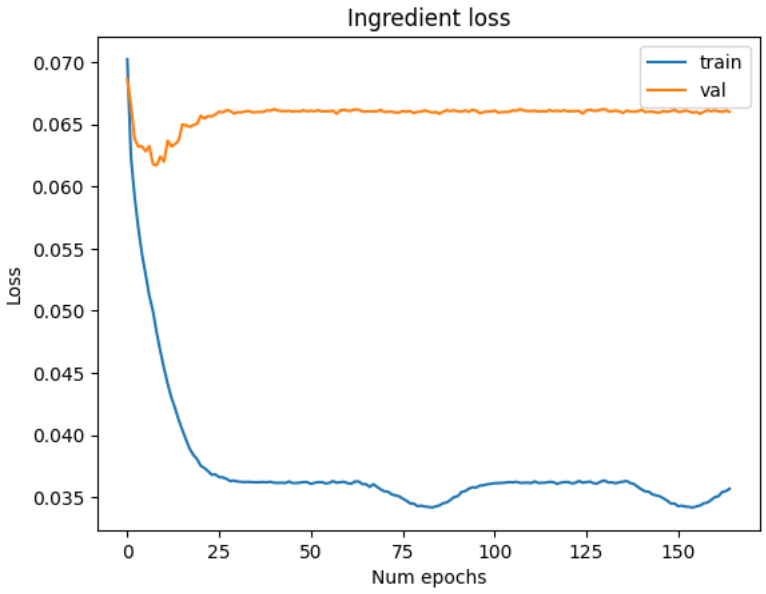


Figura 3.18: Andamento della Ingredient loss, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane

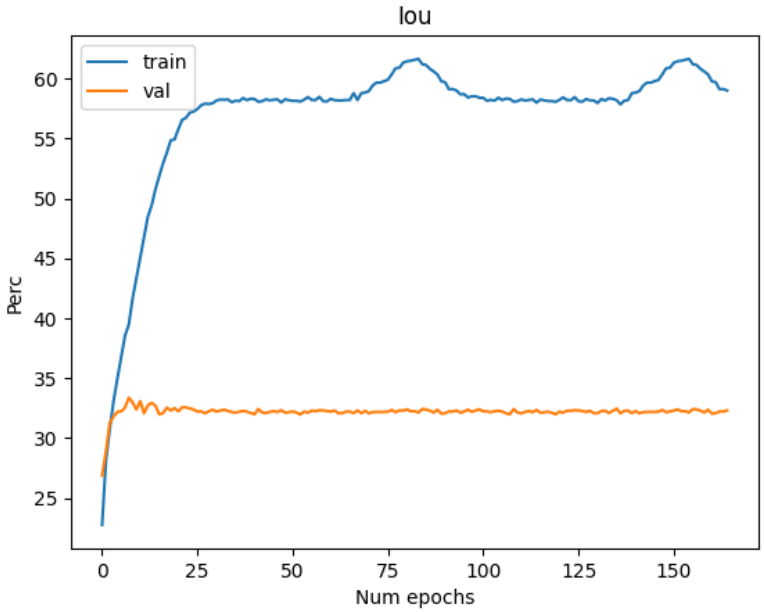


Figura 3.19: Andamento della Iou, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane

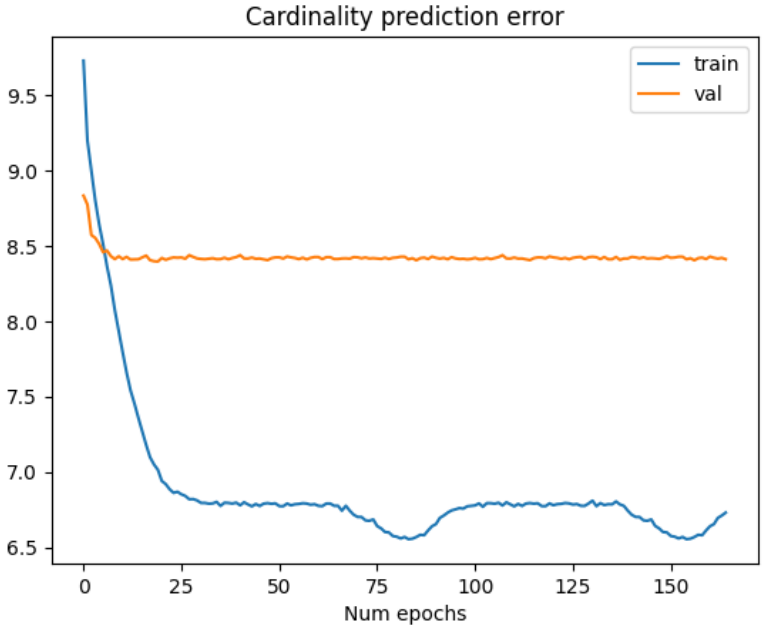


Figura 3.20: Andamento della Cardinality prediction error, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane

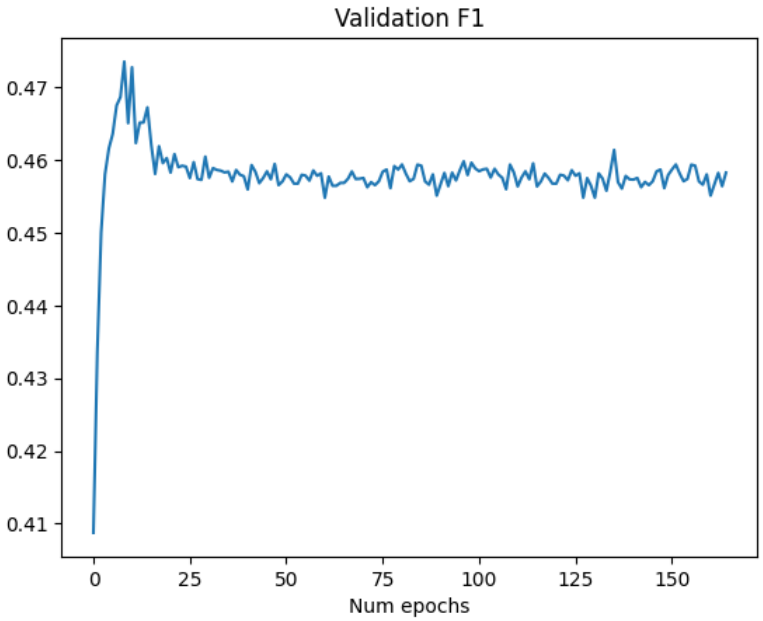


Figura 3.21: Andamento della F1, sul validation set, durante l'addestramento del modello unicamente su ricette italiane

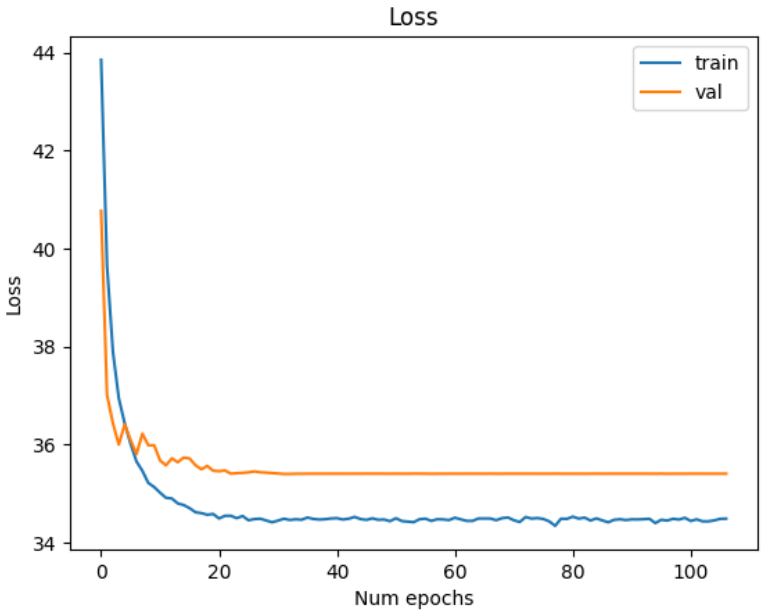


Figura 3.22: Andamento della Loss, sul training e validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori

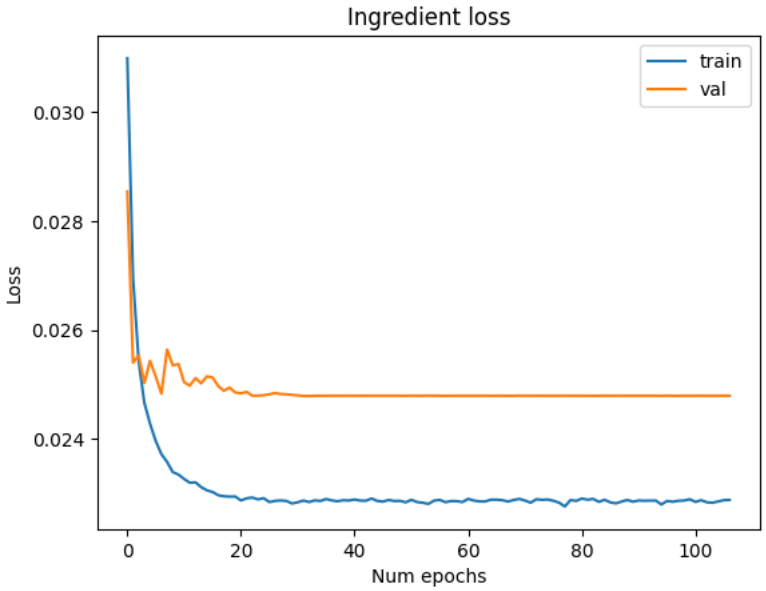


Figura 3.23: Andamento della Ingredient loss, sul training e validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori

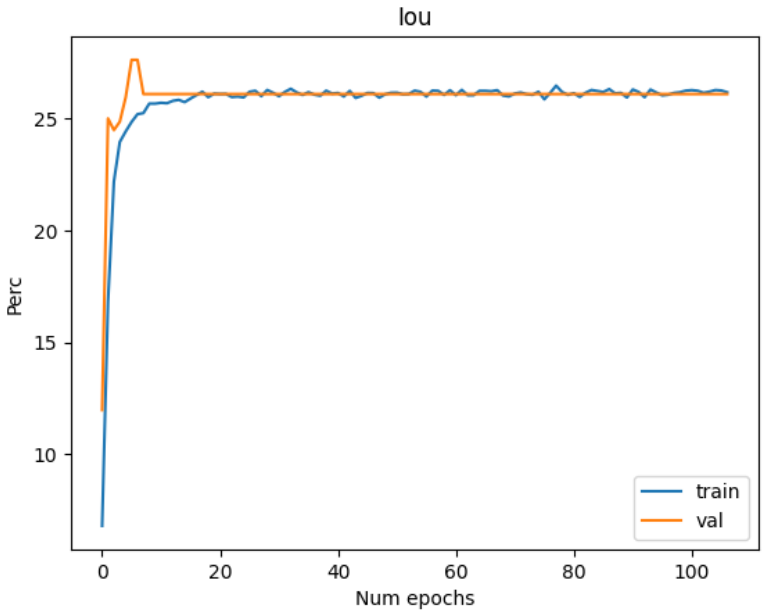


Figura 3.24: Andamento della Iou, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane

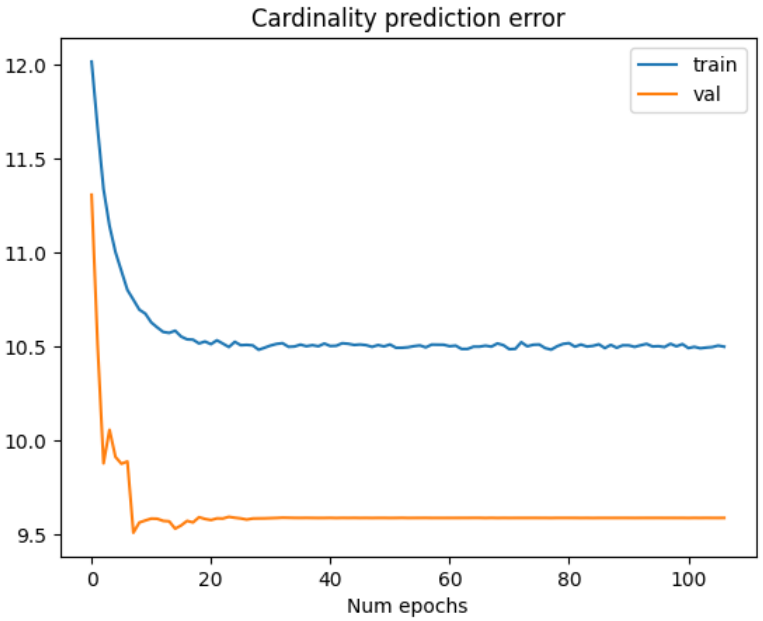


Figura 3.25: Andamento della Cardinality prediction error, sul training e validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori

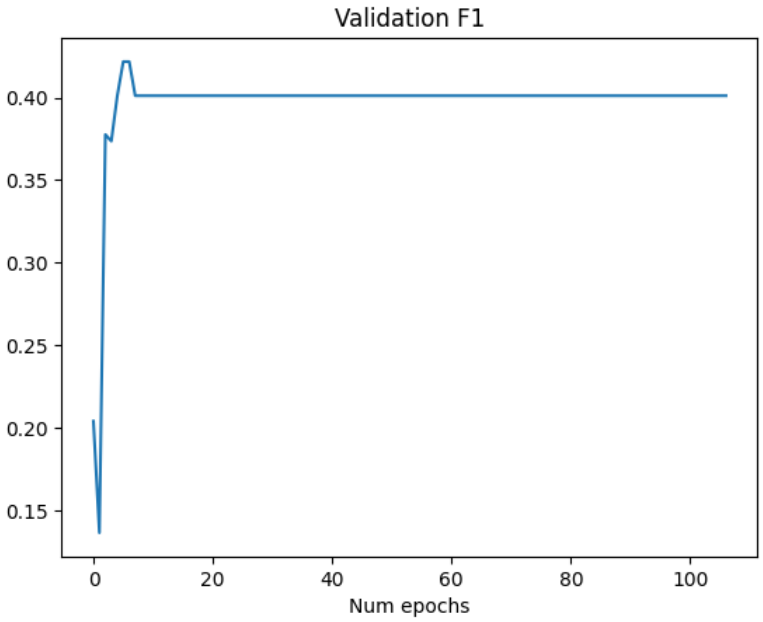


Figura 3.26: Andamento della F1, sul validation set, durante l'addestramento del modello partendo dal checkpoint fornito dagli autori

3.2.4 Principali problemi

Analizzando i risultati prodotti dai precedenti esperimenti sono state evidenziate due problematiche principali: la prima consiste in una notevole differenza di performance riscontrata tra il training e il validation set. In particolare dai grafici si evince che in seguito ad un certo numero di epoche di addestramento le prestazioni sul training set sono in continuo miglioramento, mentre sul set di validazione si arrestano e si stabilizzano. Questo fenomeno viene chiamato sovradattamento (overfitting), che rappresenta un fenomeno molto comune all'interno di modelli e algoritmi di apprendimento automatico (machine learning), caratterizzati da un numero estremamente elevato di parametri. Questi modelli solitamente vengono addestrati utilizzando un insieme di dati già noti, detto training set. Per poter allenare in maniera efficace un modello, l'algoritmo di apprendimento non solo tenta di imparare la distribuzione dei dati di questo insieme, ma è anche in grado di adattarsi a dati nuovi. Tuttavia, nel caso in cui l'addestramento sia stato eseguito per un periodo di tempo troppo lungo o con un numero limitato di esempi, il modello potrebbe adattarsi troppo bene alle caratteristiche e alla distribuzione dei dati di training perdendo la capacità di adattarsi a dati mai visti; la seconda problematica riguarda invece le peggiori performance ottenute con i modelli che sono stati sottoposti ad un pre-addestramento, su ricette diverse da quelle utilizzate

nel training set, rispetto ai modelli addestrati unicamente sulle ricette italiane raccolte. Questo fenomeno può essere ricondotto ad un problema conosciuto col nominativo di cambiamento di dominio (domain shift), che consiste sostanzialmente in un cambiamento nella distribuzione dei dati utilizzati per l'apprendimento di un modello, il quale non riesce ad adattarsi in maniera efficace, con conseguente peggioramento delle performance. Per maggiore chiarezza, essendo le ricette utilizzate per il pre-addestramento interamente in lingua inglese, il modello si adatta ad un determinato tipo di distribuzione, che viene completamente modificata durante il training del modello sulle ricette italiane, essendo queste rappresentate interamente in lingua italiana. Nelle sezioni seguenti si tenta di mitigare queste problematiche attraverso nuovi approcci e modifiche alle strutture originali dei modelli e del dataset.

3.3 Approccio 2

3.3.1 Idea e Implementazione

Analizzando i risultati ottenuti negli esperimenti precedenti risulta evidente la difficoltà nell'ottimizzazione e nell'ottenimento di una buona precisione per il processo di image-to-recipe, evidenziando quanto questo task risulti complicato. Fino ad ora i modelli utilizzati, presentavano delle strutture molto complesse, caratterizzate da un numero di parametri estremamente elevato, fattore che complica ulteriormente il processo di addestramento. Per questo motivo, con questo approccio si è deciso di tentare di semplificare la struttura di un modello, in modo da facilitare il più possibile il processo di training. In particolare, si è deciso di operare sull'architettura di img2recipe, andando ad escludere la componente che teneva conto delle istruzioni per la realizzazione della rappresentazione di una ricetta. Questa scelta risulta coerente con l'obiettivo di questo lavoro di tesi, in quanto per il calcolo dei valori nutrizionali di un determinato piatto sono necessari unicamente gli ingredienti che lo compongono, mentre le istruzioni per la realizzazione non forniscono alcuna informazione aggiuntiva. La struttura del modello viene dunque semplificata, rimuovendo l'encoder delle istruzioni e facendo in modo che la rappresentazione di una ricetta sia determinata solamente dal prodotto dell'encoder degli ingredienti. Per maggiore chiarezza, è sufficiente rimuovere la componente evidenziata con colore *blu* all'interno della Fig. 2.7. Le altre componenti dell'architettura non subiscono variazioni, di conseguenza una volta ottenuta la rappresentazione della ricetta, quest'ultima è proiettata in uno spazio condiviso con l'immagine nel tentativo di far apprendere al modello la corretta associazione tra questi due elementi. Una spiegazione più dettagliata di questo processo viene fornita all'interno del capitolo precedente (2.2.1).

3.3.2 Metriche ed Esperimenti

Per la realizzazione di questi esperimenti è stato necessario apportare delle modifiche al codice prodotto per le versioni precedenti, semplificando per esempio il data loader, che per ogni ricetta restituisce i dati relativi agli ingredienti che la compongono e all'immagine ad essa associata, tralasciando i vettori di rappresentazione delle istruzioni di preparazione. Per questa tipologia di approccio gli esperimenti si focalizzano su un modello addestrato unicamente sui dati delle ricette italiane, in quanto a causa della modifica dell'architettura, il checkpoint messo a disposizione dagli autori non risulta compatibile. Infine, per la valutazione delle performance si è scelto di utilizzare le stesse metriche descritte nella sezione 3.2.2, ovvero MedR, R@1, R@5 e R@10, calcolate sull'intero validation set e su un sottoinsieme di 500 elementi.

3.3.3 Risultati

Per la valutazione delle prestazioni di questa nuova versione del modello img2recipe sono stati realizzati dei grafici che vengono riportati nel seguito. Quest'ultimi sono gli stessi che sono stati realizzati per gli esperimenti precedenti sullo stesso modello, per permettere una maggiore chiarezza nel confronto. Anche in questo caso, si sottolinea il fatto che il numero di punti utilizzati nei grafici per il validation set risulta molto inferiore a quello del training set, questo perché durante l'addestramento la valutazione sul validation set non è stata effettuata ogni singola epoca, ma ogni 10 e inoltre la durata di questo esperimento risulta inferiore a quelli precedenti, 300 epoche, invece che 500, a causa delle prestazioni inferiori riscontrate. Nella Fig. 3.27 e 3.28 viene mostrato rispettivamente l'andamento della funzione di perdita coseno, *(cosine loss)*, sul training set e sul validation set. Analizzando l'andamento di questi dati, è possibile notare il progressivo decremento dei due valori, come riportato anche negli esperimenti precedenti, tuttavia in questo caso si ottengono prestazioni leggermente superiori sul validation set, raggiungendo un valore minimo di 0.609 , rispetto a 0.62 del caso precedente. Risulta evidente inoltre, come i valori, sul validation set, dopo alcune epoche si stabilizzino intorno a 0.62, mentre sul training set decrementino in maniera progressiva. Questo andamento evidenzia la presenza dello stesso problema di overfitting riscontrato negli esperimenti precedenti, mostrando che la semplificazione del modello non risulta sufficiente per evitare il sovradattamento sui dati di addestramento. Le Fig 3.29 e 3.30 riportano invece l'andamento della *image loss* e della *recipe loss*, rispettivamente, utilizzate per la regolarizzazione semantica del modello, nelle quali è possibile riscontrare lo stesso fenomeno evidenziato precedentemente. La Fig. 3.31 mostra invece i valori di median rank (medR), ottenuti con l'avanzare dell'esperimento, calcolati rispetto a tutto il validation set e ad un sottoinsieme di esso formato da 500 elementi, mentre la Fig. 3.32 riporta i valori di recall rate

(R@K). Osservando questi valori è possibile notare come le prestazioni risultino nettamente inferiori, sia sull'intero validation set, sia su un sottoinsieme di 500 elementi (47.0 e 29.75 rispettivamente), rispetto a quelle mostrate negli esperimenti precedenti (20.0 e 13.5), dimostrando che escludere le istruzioni di preparazione e considerare unicamente gli ingredienti produce una rappresentazione delle ricette meno efficace. Infine, la tabella 3.2 riporta i migliori risultati ottenuti durante l'esperimento, mostrando anche le performance sul test set.

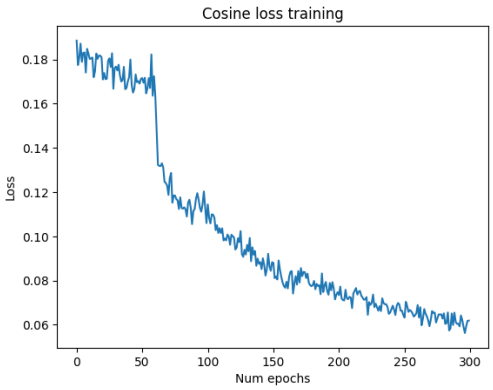


Figura 3.27: Andamento della cosine loss, sul training set, durante l'addestramento del modello, unicamente su ricette italiane, considerando solamente gli ingredienti che lo compongono ed escludendo le istruzioni

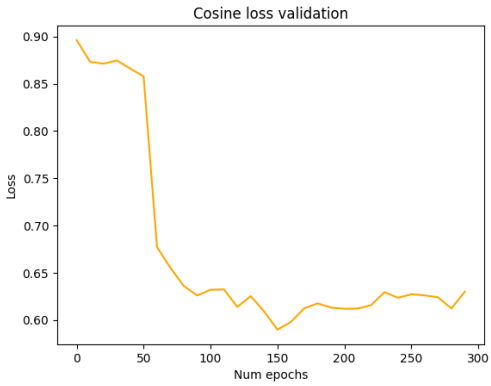


Figura 3.28: Andamento della cosine loss, sul validation set, durante l'addestramento del modello unicamente su ricette italiane, considerando solamente gli ingredienti che lo compongono ed escludendo le istruzioni

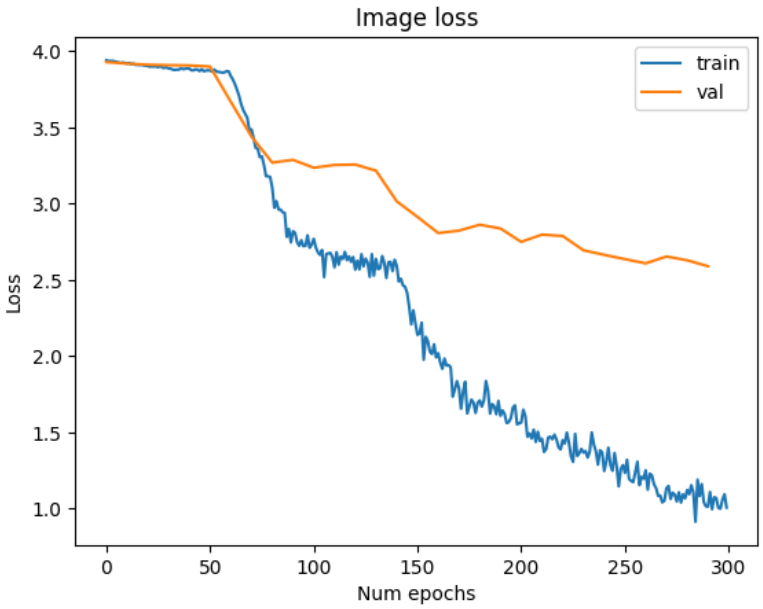


Figura 3.29: Andamento della image loss, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane, considerando solamente gli ingredienti che lo compongono ed escludendo le istruzioni

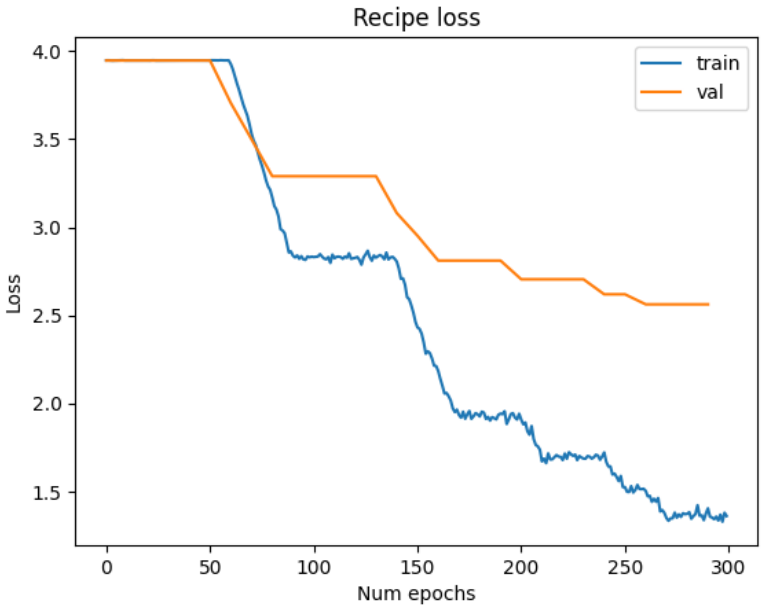


Figura 3.30: Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane, considerando solamente gli ingredienti che lo compongono ed escludendo le istruzioni

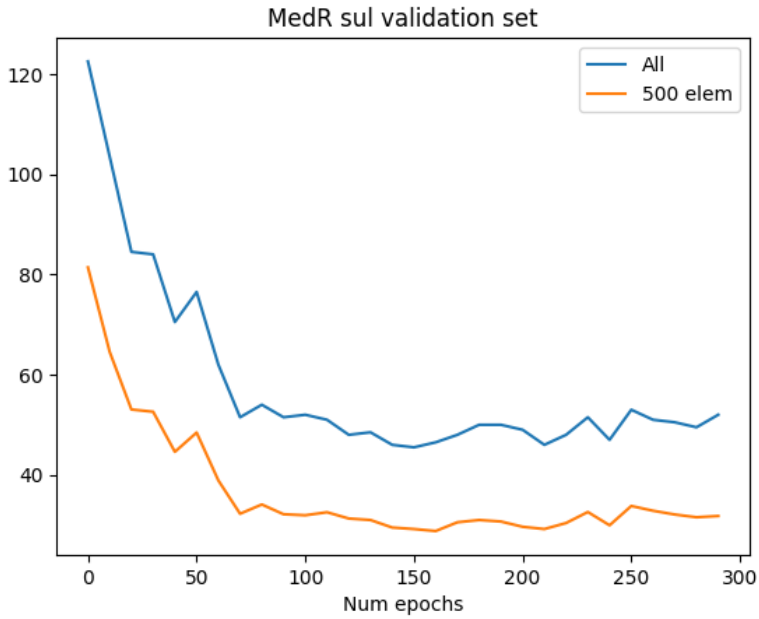


Figura 3.31: Andamento della MedR, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane, considerando solamente gli ingredienti che lo compongono ed escludendo le istruzioni

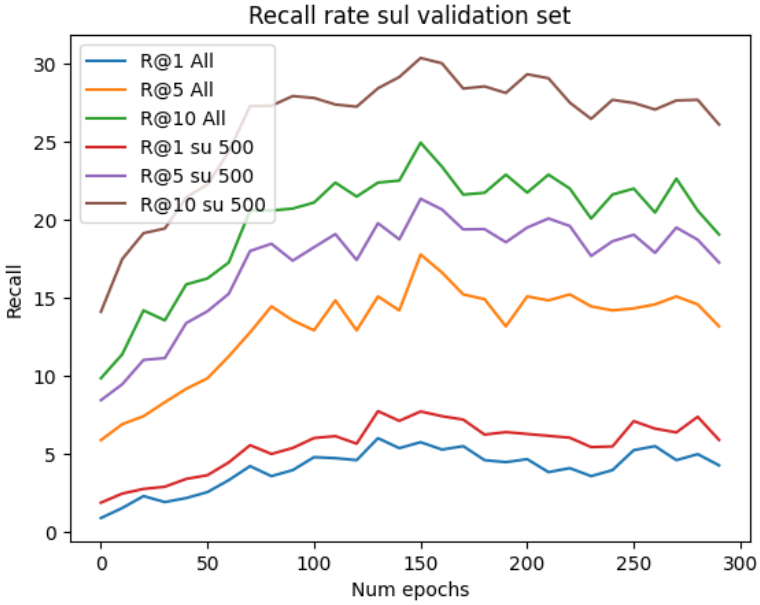


Figura 3.32: Andamento del recall rate, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane, considerando solamente gli ingredienti che lo compongono ed escludendo le istruzioni. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette

Tabella 3.2: Risultati degli esperimenti sul modello img2recipe escludendo le istruzioni dalla composizione delle ricette

	From 0		
	Train	**Val**	**Test**
Cosine Loss	0.0802	0.6089	0.6232
Image Loss	2.1664	3.0167	2.9817
Recipe Loss	2.4691	3.0810	2.8437
MedR (Tot)	/	47.0	45.0
R@1 (Tot)	/	5.4%	4.4%
R@5 (Tot)	/	14.6%	14.7%
R@10 (Tot)	/	22.6%	24.3%
MedR (500 elem)	/	29.75	28.7
R@1 (500 elem)	/	7.0%	6.18%
R@5 (500 elem)	/	18.9%	20.6%
R@10 (500 elem)	/	28.5%	28.8%

3.4 Approccio 3

3.4.1 Idea e implementazione

Come evidenziato nelle sezioni precedenti, uno dei principali problemi del modello img2recipe, riscontrato durante l'addestramento partendo dal checkpoint fornito dagli autori, è la differenza tra la distribuzione dei dati di pre-training e la distribuzione di quelli utilizzati per il training effettivo. Questa differenza si traduce in una maggiore difficoltà da parte del modello nell'adattarsi ai nuovi dati, ottenendo così prestazioni inferiori a quelle di un modello addestrato unicamente su ricette italiane. Nel tentativo di ridurre questo effetto di cambiamento di dominio (domain shift) è stata effettuata la traduzione, in lingua inglese, di tutte le ricette italiane raccolte. Questo processo è stato eseguito attraverso la libreria python *googletrans* [50] che ha permesso la traduzione di tutte le componenti delle diverse ricette, ovvero, titolo, ingredienti e istruzioni per la preparazione. Una volta terminato questo processo sono state eseguite le operazioni di preparazione dei dati, descritte nelle sezioni precedenti (3.2.2), apportando qualche leggera modifica. In particolare, per la *pulizia degli ingredienti*, è stato tradotto in lingua inglese il file già ricavato precedentemente per le ricette in lingua italiana, senza eseguire nuovamente tutti i passaggi; per la *creazione del dizionario* invece, si è preferito utilizzare quello fornito dagli autori insieme al checkpoint; infine per la *rappresentazione delle istruzioni* oltre al testo estratto dalle ricette italiane appena tradotte, è stato anche utilizzato una parte di testo preso delle ricette utilizzate dagli autori per l'addestramento del loro modello. Tutte queste modifiche sono state realizzate con l'obiettivo di

mitigare il più possibile la distanza tra i due domini e migliorare le performance ottenute.

3.4.2 Metriche

Anche per questo esperimento per la valutazione delle performance si è scelto di utilizzare le metriche impiegate e descritte nella sezione 3.2.2, ovvero MedR, R@1, R@5 e R@10, calcolate sull'intero validation set e su un sottoinsieme di 500 elementi.

3.4.3 Risultati

Per permettere una maggiore chiarezza nel confronto con i risultati precedenti, per questo esperimento sono stati realizzati gli stessi grafici riportati nelle sezioni precedenti. Anche in questo caso, si sottolinea il fatto che il numero di punti utilizzati nei grafici per il validation set risulta molto inferiore a quello del training set, questo perché durante l'addestramento la valutazione sul validation set non è stata effettuata ogni singola epoca, ma bensì ogni 10. Nelle Fig. 3.33 e 3.34 in particolare, viene riportato l'andamento della funzione di perdita coseno (cosine loss) rispettivamente sul training e sul validation set. Confrontando questi valori con quelli ottenuti dal modello precedente, partendo anch'esso dal checkpoint fornito dagli autori, ma addestrato sulle ricette italiane non tradotte in lingua inglese 3.2.2, evidenziamo come questo modello, in particolare sul validation set, ottenga risultati decisamente migliori, raggiungendo un valore minimo intorno agli 0.6, dove precedentemente si era raggiunto solamente un valore di 0.75. Queste prime osservazioni, lasciano intuire che le modifiche apportate al dataset delle ricette siano servite per ridurre la distanza tra la distribuzione dei dati di pre-training e quelli di addestramento, mitigando in questo modo il problema di cambiamento di dominio (domain shift) riscontrato in precedenza. Tuttavia, come per gli altri esperimenti, è possibile riscontrare che i valori, sul validation set, dopo alcune epoche si stabilizzino, mentre sul training set decrementano in maniera progressiva, evidenziando la presenza di un problema di overfitting. Le Fig 3.35 e 3.36 riportano invece l'andamento della *image loss* e della *recipe loss*, rispettivamente, utilizzate per la regolarizzazione semantica del modello, nelle quali è possibile constatare lo stesso fenomeno evidenziato precedentemente. La Fig. 3.37 invece, mostra i valori di median rank (medR), ottenuti con l'avanzare dell'esperimento, calcolati rispetto a tutto il validation set e ad un sottoinsieme di esso formato da 500 elementi. Anche in questo caso, rispetto all'esperimento citato in precedenza, le performance sono decisamente migliori, raggiungendo valori di circa 23.0 e 13.65 sull'intero validation set e su un sottoinsieme di esso, formato da 500 elementi, rispetto ai precedenti 62.5 e 40.6. Questo andamento è confermato anche nei risultati, ottenuti per la

metrica del recall rate, riportati nella Fig. 3.38, dalla quale si nota chiaramente un miglioramento di diversi punti percentuali, (5% per R@1 e 15% per il R@10). Inoltre, la tabella 3.3 riporta i migliori risultati ottenuti durante l'esperimento, mostrando anche le performance sul test set. In conclusione, la traduzione delle ricette italiane in lingua inglese ha aiutato notevolmente il modello nell'ottenimento di performance migliori, che risultano comparabili anche con quelle ottenute dal modello addestrato unicamente sulle ricette italiane.

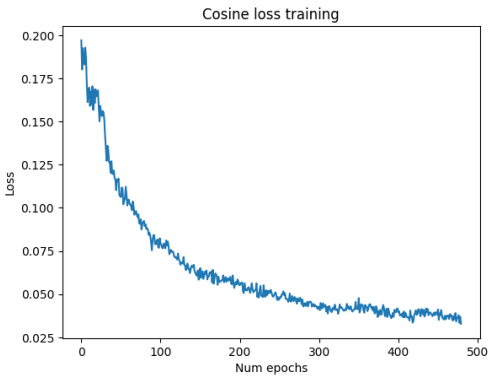


Figura 3.33: Andamento della cosine loss, sul training set, durante l'addestramento del modello sulle ricette italiane tradotte in lingua inglese, partendo dal checkpoint fornito dagli autori

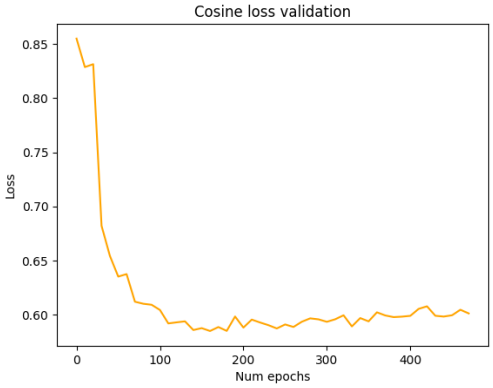


Figura 3.34: Andamento della cosine loss, sul validation set, durante l'addestramento del modello sulle ricette italiane tradotte in lingua inglese, partendo dal checkpoint fornito dagli autori

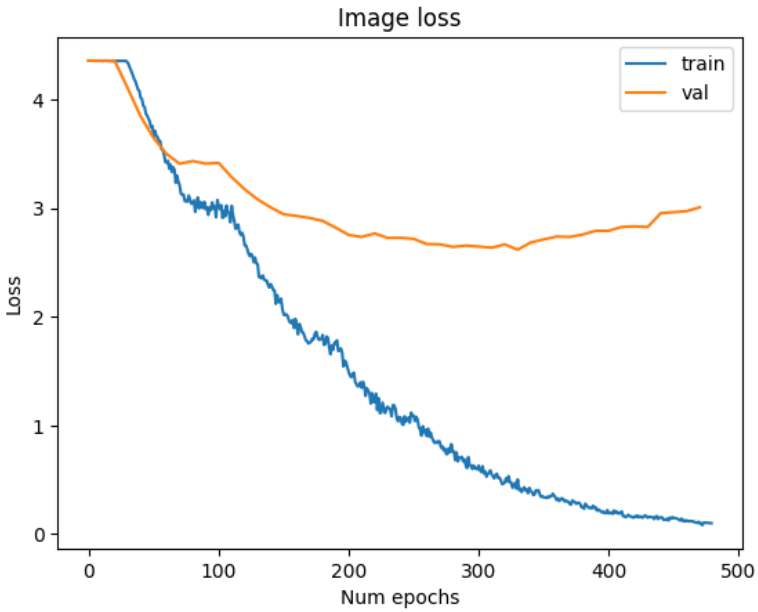


Figura 3.35: Andamento della image loss, sul training e validation set, durante l'addestramento del modello sulle ricette italiane tradotte in lingua inglese, partendo dal checkpoint fornito dagli autori

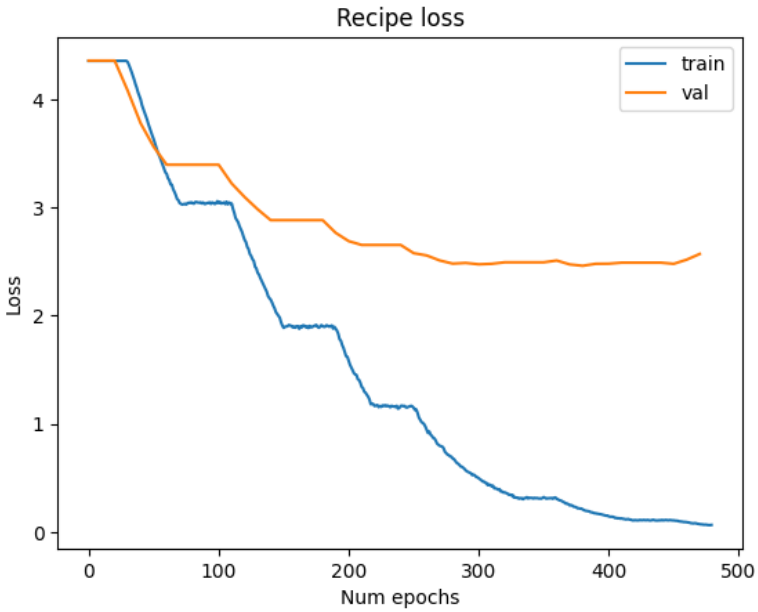


Figura 3.36: Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello sulle ricette italiane tradotte in lingua inglese, partendo dal checkpoint fornito dagli autori

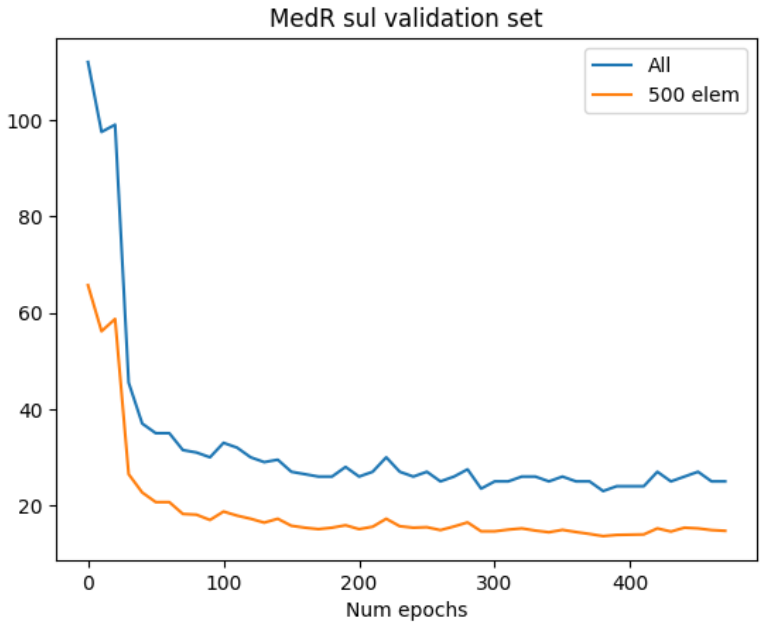


Figura 3.37: Andamento della MedR, sul training e validation set, durante l'addestramento del modello sulle ricette italiane tradotte in lingua inglese, partendo dal checkpoint fornito dagli autori

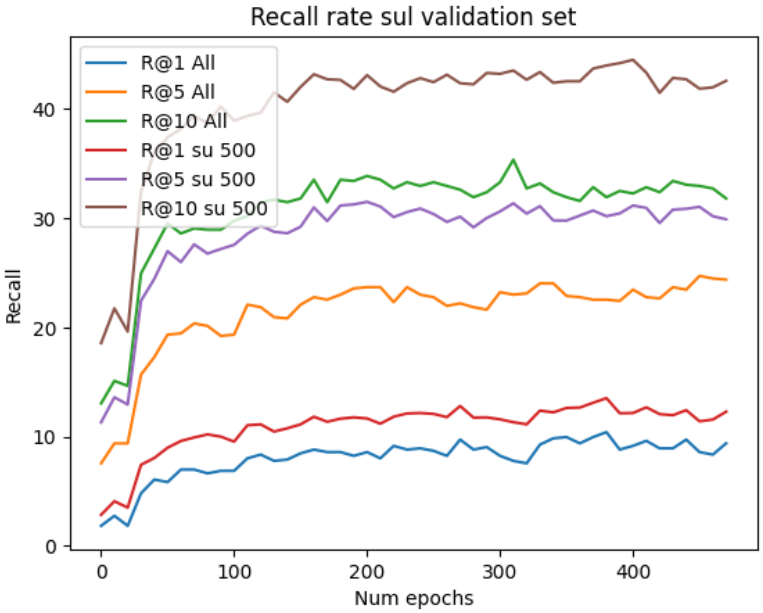


Figura 3.38: Andamento del recall rate, sul training e validation set, durante l'addestramento del modello sulle ricette italiane tradotte in lingua inglese, partendo dal checkpoint fornito dagli autori. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette

Tabella 3.3: Risultati degli esperimenti sul modello img2recipe sulle ricette italiane tradotte in lingua inglese, partendo dal checkpoint fornito dagli autori

	From check		
	Train	**Val**	**Test**
Cosine Loss	0.0344	0.5976	0.6102
Image Loss	0.2119	2.7559	2.4784
Recipe Loss	0.1787	2.4615	2.0272
MedR (Tot)	/	23.0	30.0
R@1 (Tot)	/	10.4%	9.8%
R@5 (Tot)	/	22.5%	23.5%
R@10 (Tot)	/	31.9%	32%
MedR (500 elem)	/	13.65	18.0
R@1 (500 elem)	/	13.5%	13%
R@5 (500 elem)	/	30.2%	30%
R@10 (500 elem)	/	43.9%	39.9%

3.5 Approccio 4

3.5.1 Nuova versione del dataset

Nel tentativo di mitigare le principali problematiche descritte negli approcci precedenti, è stata prodotta una nuova versione del dataset contenente le ricette culinarie italiane. Attraverso la stessa metodologia e gli stessi siti utilizzati per la realizzazione della prima versione 3.2.1, sono state raccolte un elevato numero di nuove ricette, che hanno permesso di raddoppiare la dimensione originale del database. In particolare, da un numero di piatti di circa 5000 si è passati a più di 10000 (10112), mantenendo però lo stesso rapporto di partizionamento presente nella versione precedente. Di conseguenza il 70% dei dati è stato destinato all'addestramento del modello (6897), mentre il restante 30% è stato suddiviso equamente tra validation e test set, (1592 e 1611 rispettivamente). Inoltre come ulteriore miglioramento apportato a questa versione, prendendo spunto dall'idea utilizzata dagli autori del dataset recipe1M+ [2], si è scelto di aumentare il numero di immagini associate alla singola ricetta. Questa decisione ha l'obiettivo di aumentare la capacità di generalizzazione del modello, ovvero la precisione con cui il modello è in grado di riconoscere un determinato piatto anche se questo viene rappresentato in maniere differenti. Questa capacità risulta estremante importate per il tipo di applicativo che intende realizzare questo lavoro di tesi, basti pensare alle moltissime variazioni che possono verificarsi in termini di composizione, illuminazione dell'immagine e angolazione dello scatto, nel momento in cui viene effettuata la foto della pietanza

che viene consumata. Nello specifico si è scelto di collezionare un numero massimo di 7 immagini differenti per ogni ricetta, che sono state raccolte tramite uno strumento di scraping chiamato, Google Image Scraper [8], che tramite il titolo del piatto esegue una ricerca su Google immagini e permette di salvare i primi risultati trovati. Come nel caso della precedente versione, si è scelto di non scaricare le diverse immagini in locale, a causa dell'eccessiva quantità di memoria richiesta, ma di accedervi tramite il modulo *requests* di python, sfruttando gli URL salvati. Il dataset realizzato mantiene la stessa struttura descritta nella sezione precedente 3.2.1, tuttavia non presenta più il singolo URL dell'immagine associata, ma una lista di massimo 7 elementi. Di seguito vengono riportati una serie di grafici che riportano alcune statistiche relative a questa collezione di dati, mostrando anche le analogie e le differenze rispetto alla prima versione. Nella Fig. 3.39 è riportata la distribuzione delle ricette in base alla portata di appartenenza, considerando le partizioni di training, validation e test. Il grafico mostra, come nella sua precedente versione 3.1, una prevalenza di ricette appartenenti alle portate dei *primi*, *secondi* e *dolci*, che risultano tuttavia meno bilanciate tra loro, a causa di una maggiore presenza di ricette riguardanti i *dolci*. Inoltre è possibile notare una minore presenza in percentuale di ricette appartenenti agli *antipasti*, in quanto le ricette aggiunte per questa portata sono circa 400, rispetto alle portate citate prima, per cui questo aumento si aggira attorno ai 700/900. Come nel dataset precedente anche in questo caso le portate che presentavano un numero di ricette estremamente basso sono state raggruppate all'interno della categoria *altri*. Nel grafico riportato nella Fig. 3.40 invece è mostrata la distribuzione del numero di ingredienti che compongono le diverse ricette. La maggior parte dei piatti risulta composta da un numero di ingredienti compreso tra 4 e 15, con una maggiore concentrazione introno ai valori di 8, 9 e 10, come anche nella versione precedente, mostrata nella Fig. 3.2. Come unica differenza è possibile notare che sono presenti, anche se in numero estremamente ridotto, delle ricette che sono composte da più di 30 ingredienti diversi. Infine, le Fig. 3.41 e 3.42 mostrano rispettivamente, la distribuzione del numero di istruzioni per ogni ricetta e la distribuzione delle lunghezze delle istruzioni stesse. Dal primo grafico è possibile notare una maggiore concentrazione di ricette composte da 6/7 istruzioni differenti, mentre nella collezione precedente questo picco risulta intorno ai valori 4/5, di conseguenza abbiamo un aumento nel numero di istruzioni elaborate per il singolo piatto. Nel secondo grafico invece è possibile notare come la lunghezza delle singole istruzioni si aggiri intorno a valori inferiori alle 25 parole, a differenza del grafico 3.3 in cui invece, in media, si riscontra una lunghezza maggiore.

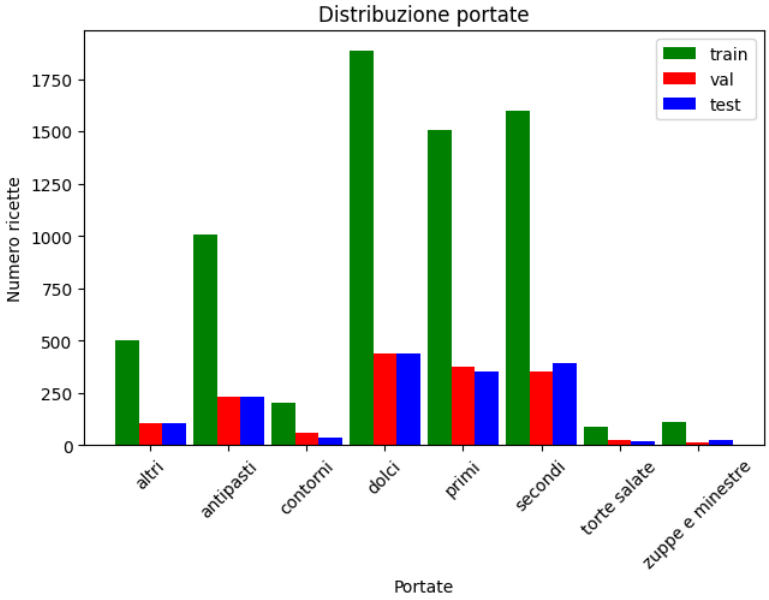


Figura 3.39: Nuova versione del database: distribuzione delle ricette secondo la portata di appartenenza per le partizioni di training, validation e test

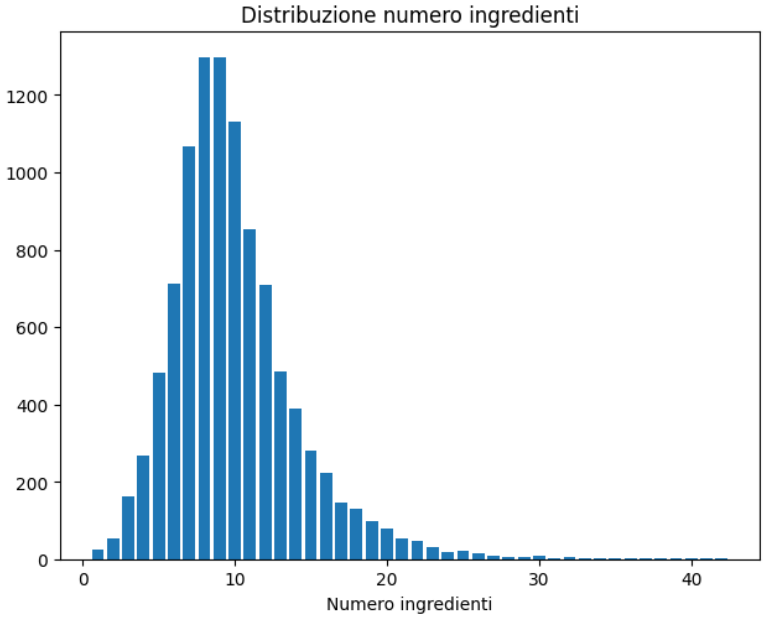


Figura 3.40: Nuova versione del database: distribuzione del numero di ingredienti differenti per ogni ricetta

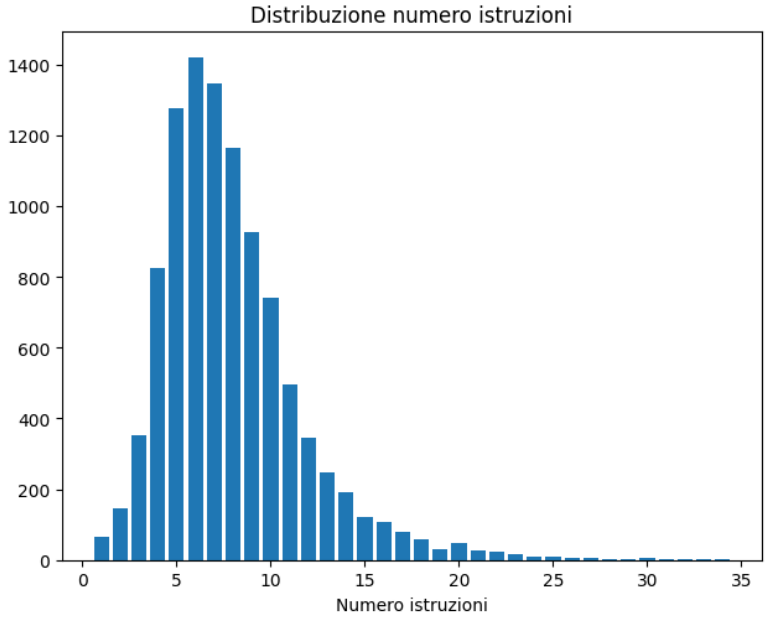


Figura 3.41: Nuova versione del database: distribuzione del numero di istruzioni per ogni ricetta

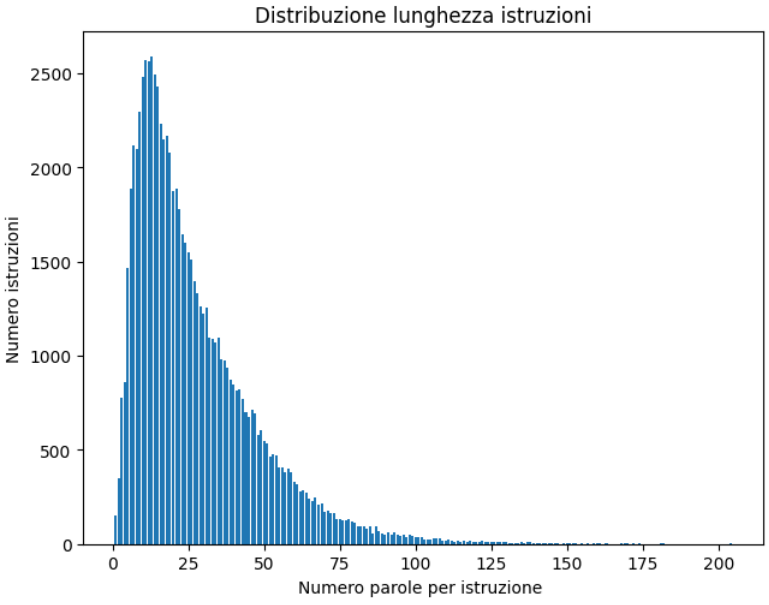


Figura 3.42: Nuova versione del database: distribuzione del numero di parole che compongono le diverse istruzioni di ogni ricetta

3.5.2 Esperimenti con img2recipe

Implementazione

Per la realizzazione dei nuovi esperimenti con il modello img2recipe, utilizzando la versione aggiornata del dataset sopra descritta, è stato necessario apportare delle modifiche alla struttura del modello e in particolare al data loader utilizzato, che fino a quel momento era in grado di gestire una singola immagine per ogni ricetta. In particolare, è stato introdotto, per ogni accesso ad una singola ricetta, un ordinamento casuale della lista di immagini, garantendo in questo modo una maggiore diversità nell'associazione ricetta-immagine. Inoltre nel caso in cui l'URL di un elemento non risulti disponibile o il modulo *requests* di python riscontri un errore, l'immagine non viene considerata, passando alla successiva. Le fasi di pre-processamento sono equivalenti a quelle descritte negli esperimenti precedenti 3.2.2. Viste le migliori performance riscontrate dal modello addestrato unicamente su ricette italiane, rispetto che a quello che prende in considerazione il miglior checkpoint fornito dagli autori, viene considerato solamente il primo per gli esperimenti su questa nuova versione del dataset.

Metriche

Per la valutazione delle performance del modello sono impiegate le stesse metriche degli esperimenti precedenti ovvero, median rank (MedR) e recall rate (R@K), in particolare R@1, R@5 e R@10. In questo caso però, per ottenere un corretto adattamento del dataset, questi valori sono calcolati rispetto a tutti gli elementi del validation set, ovvero 1592, un sottoinsieme di 1000 coppie ricetta-immagine, per permettere dei confronti con i risultati ottenuti dagli autori del modello ed un sottoinsieme di 500 elementi, per un confronto con la precedente versione del database. Inoltre, le valutazioni vengono effettuate, come in precedenza, 10 volte, su sottoinsiemi scelti casualmente, considerando poi i valori medi.

Risultati

Come per gli esperimenti precedenti anche in questo caso sono stati prodotti dei grafici per rappresentare le performance del modello e facilitare il confronto con i risultati precedenti. Anche in questo caso, si sottolinea il fatto che il numero di punti utilizzati nei grafici per il validation set risulta molto inferiore a quello del training set, questo perché durante l'addestramento la valutazione sul validation set non è stata effettuata ogni singola epoca, ma ogni 10 e inoltre la durata di questo esperimento risulta inferiore a quelli precedenti, 300 epoche, invece che 500, a causa degli elevati tempi di addestramento del modello. Nella Fig. 3.43 e 3.44 viene mostrato rispettivamente l'andamento della funzione di perdita coseno, *(cosine loss)*,

sul training set e sul validation set. Come negli altri casi osserviamo un andamento decrescente dei due valori, che però, nel caso del training set, risulta molto più lento rispetto a quello osservato nei risultati della prima versione del dataset, dove all'epoca 300 il valore era intorno agli 0.05, mentre in questo caso risulta circa il doppio (0.1). Questo fenomeno può essere attribuito alla maggiore complessità del modello, che associa fino a 7 immagini differenti ad una singola ricetta, rendendo più difficile per il modello adattarsi. Considerando il validation set invece, è possibile notare come nelle prime epoche la riduzione sia molto lieve, mentre dopo la 100-esima epoca decresca molto, stabilizzandosi intorno allo 0.52. Rispetto agli esperimenti precedenti quindi, abbiamo una riduzione dell'effetto di overfitting, che in precedenza non permetteva di ottenere valori inferiori allo 0.6, mostrando come l'aumento della dimensione del dataset e del numero di immagini per ricetta abbiano fornito al modello una maggiore capacità di generalizzazione. Tuttavia questo fenomeno resta comunque molto presente, limitando le prestazioni del modello su validation e test set. Le Fig 3.45 e 3.46 riportano l'andamento della *image loss* e della *recipe loss*, rispettivamente, utilizzate per la regolarizzazione semantica del modello, nelle quali è possibile riscontrare lo stesso andamento evidenziato precedentemente. La Fig. 3.47 mostra invece i valori di median rank (medR), ottenuti con l'avanzare dell'esperimento, calcolati rispetto a tutto il validation set, ad un primo sottoinsieme formato da 1000 elementi e ad un secondo formato da 500 elementi, mentre la Fig. 3.48 riporta le performace in termini di recall rate (R@K). Confrontando i risultati riportati con quelli degli esperimenti precedenti è possibile notare delle performance molto simili, addirittura per R@1 e R@5 leggermente inferiori. Questo fenomeno può essere attribuito, come già specificato prima, alla maggiore difficoltà di adattamento da parte del modello che, associando più di un'immagine alla singola ricetta, rende più complesso l'ottenimento di performance elevate. In particolare, l'associazione di 7 immagini per ogni singola ricetta potrebbe risultare eccessivo per il modello, una possibile soluzione quindi prevede di ridurre questo numero a 3/4, per garantire comunque una riduzione del fenomeno di overfitting, ma senza impattare troppo sulle performance. Inoltre il tempo per cui il modello è stato addestrato è inferiore, di conseguenza, è ragionevole supporre che, con un training più lungo le performance di questo modello superino quelle precedenti. Infine, nella tabella 3.4 sono riportati i migliori risultati ottenuti durante gli esperimenti, evidenziando anche come le performance sul test set siano molto simili a quelle del validation set.

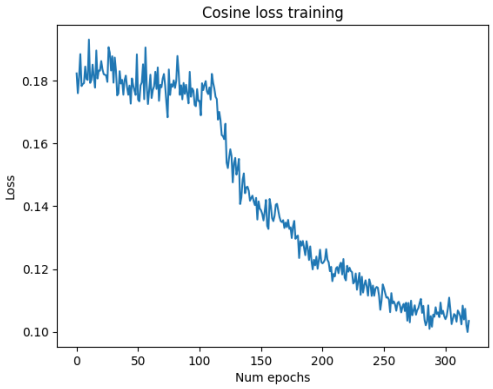


Figura 3.43: Andamento della cosine loss, sul training set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane

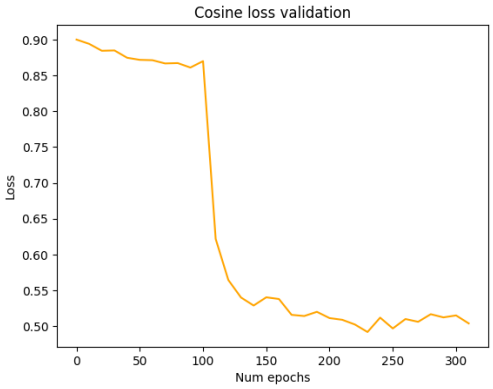


Figura 3.44: Andamento della cosine loss, sul validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane

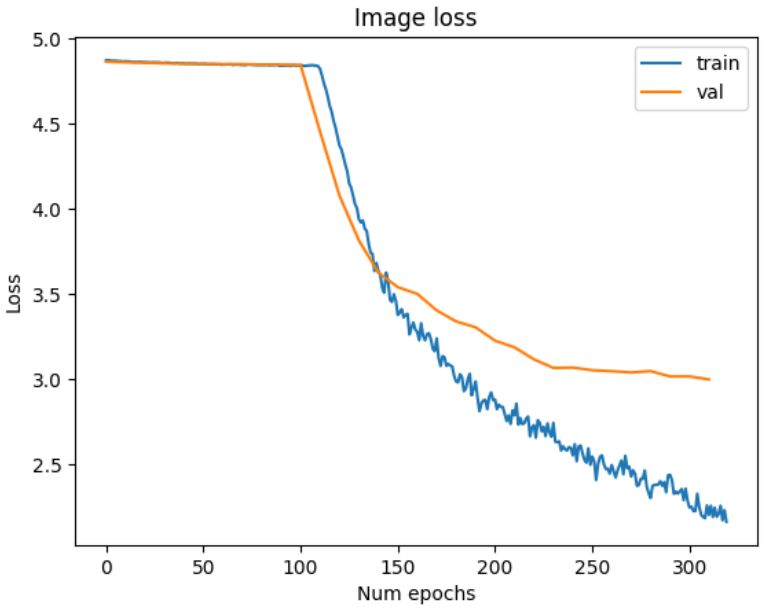


Figura 3.45: Andamento della image loss, sul training e validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane

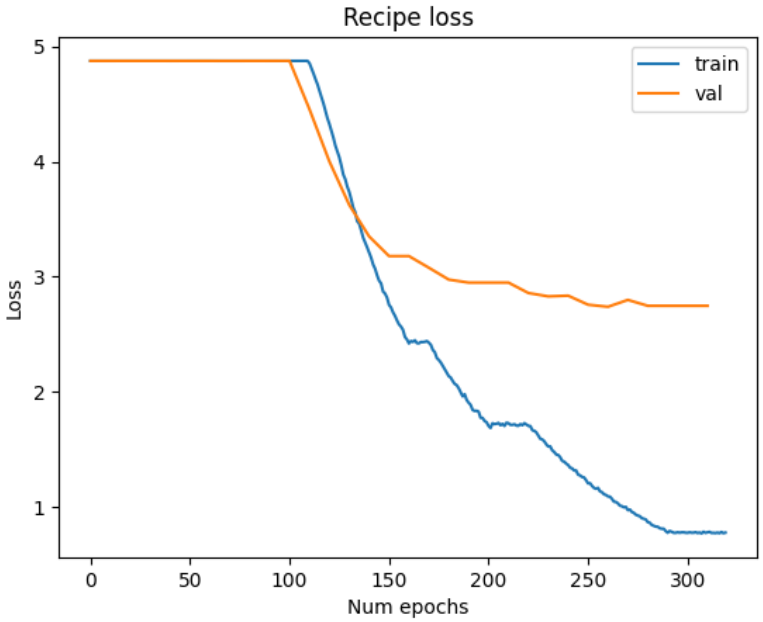


Figura 3.46: Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane

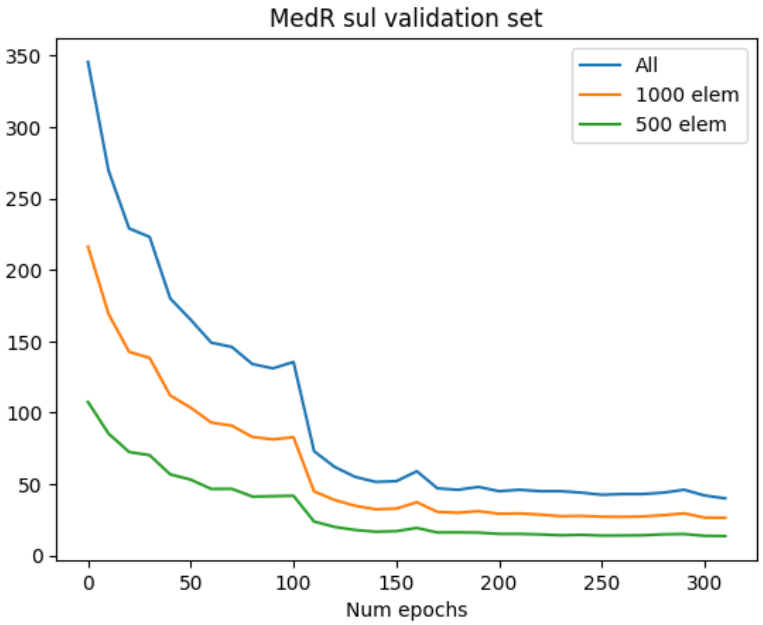


Figura 3.47: Andamento della MedR, sul training e validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane

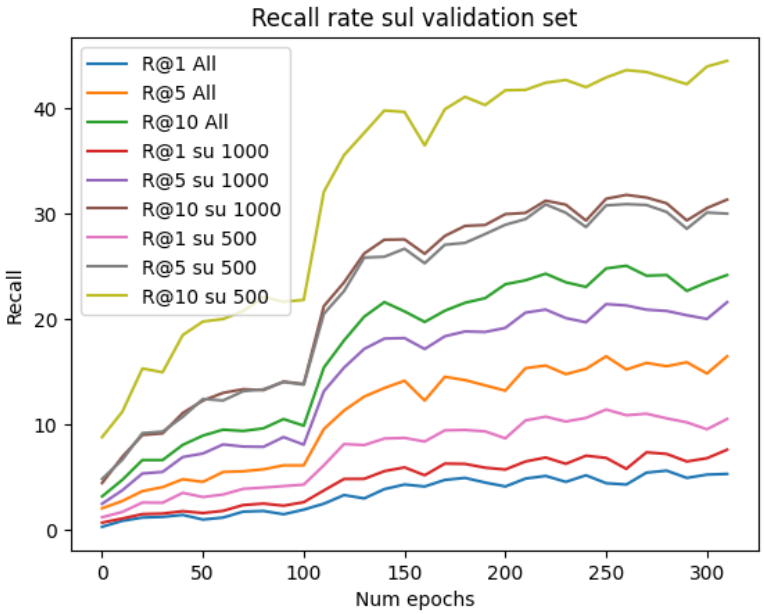


Figura 3.48: Andamento del recall rate, sul training e validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette

Tabella 3.4: Risultati degli esperimenti sul modello img2recipe addestrato sulla nuova versione del dataset delle ricette italiane

	From 0		
	Train	**Val**	**Test**
Cosine Loss	0.0976	0.5038	0.4985
Image Loss	2.0629	3.0004	2.6763
Recipe Loss	0.7668	2.7483	2.4538
MedR (Tot)	/	40.0	43.0
R@1 (Tot)	/	5.3%	5.5%
R@5 (Tot)	/	16.5%	17.1%
R@10 (Tot)	/	24.2%	25.8%
MedR (1000 elem)	/	26.4	27.2
R@1 (1000 elem)	/	7.6%	7.8%
R@5 (1000 elem)	/	21.6%	22.4%
R@10 (1000 elem)	/	31.4%	32.9%
MedR (500 elem)	/	13.55	13.8
R@1 (500 elem)	/	10.5%	12.8%
R@5 (500 elem)	/	30.1%	32.5%
R@10 (500 elem)	/	44.5%	44.3%

3.5.3 Esperimenti con inverseCooking

Implementazione

Come nel caso dei nuovi esperimenti con il modello img2recipe, anche per inverse-Cooking, sono state apportate delle leggere modifiche al data loader per permettere l'accesso a più immagini per una singola ricetta. Inoltre, le fasi di pre-processamento sono equivalenti a quelle descritte nella sezione precedente 3.2.3. Anche in questo caso, come per il modello img2recipe, viste le migliori performance riscontrate dal modello addestrato unicamente su ricette italiane, rispetto che a quello che prende in considerazione il miglior checkpoint fornito dagli autori, viene considerato solamente il primo per gli esperimenti su questa nuova versione del dataset.

Metriche

Per la valutazione delle performance del modello vengono impiegate le stesse metriche utilizzate negli esperimenti precedenti, ovvero Intersection over Union (IoU), cardinality prediction error ed F1. Anche in questo caso l'attenzione è incentrata principalmente sulla precisione nella generazione degli ingredienti, tralasciando quindi la valutazione dalle istruzioni per la preparazione delle ricette.

Risultati

Per questi esperimenti, sono stati realizzati dei grafici per la valutazione delle prestazioni del modello e per facilitare il confronto con i risultati ottenuti nella versione precedente del dataset. Si sottolinea, anche in questo caso, che il numero di punti utilizzati nei grafici per il validation set è equivalente a quelli del training set, questo perché durante l'addestramento la valutazione sul validation set viene effettuata ogni singola epoca. Inoltre, a causa dell'elevato tempo di training richiesto, la durata dell'esperimento risulta inferiore a quelli effettuati per la prima versione del database. Nella Fig. 3.49 viene mostrato l'andamento della funzione di perdita totale (loss), che nel caso del trainng set decrementa in maniera costante nelle prime epoche, per poi assumere un andamento ondulatorio con valori minimi intorno al 50, mentre per il validation set, dopo una riduzione nelle prime epoche, si stabilizza su valori poco superiori al 60, con un andamento molto simile a quello osservato negli esperimenti precedenti. In particolare però, abbiamo che la loss sul validation set, raggiunge valori decisamente inferiori rispetto a quelli ottenuti in precedenza, superiori al 70, mentre sul training set si può notare l'effetto opposto. Questo fenomeno può essere attribuito al numero maggiore di immagini per singola ricetta, che, da un lato permettono al modello di acquisire una maggiore capacità di generalizzazione e dall'altro rendono più complesso l'adattamento al training set. Inoltre, è possibile osservare una diminuzione nella distanza tra i valori del training e del validation set, suggerendo quindi una riduzione del problema di overfitting, seppur ancora molto presente, che caratterizzava gli esperimenti precedenti. La Fig. 3.50 riporta invece l'avanzamento della funzione di perdita riferita agli ingredienti (ingredient loss), ovvero una funzione che regola l'andamento dell'addestramento in base alla precisione ottenuta sulla generazione degli ingredienti. Il grafico risulta molto simile a quello precedente, questo perché la funzione di perdita totale è calcolata come somma pesata di diverse loss, tra cui, l'*ingredient loss*, la *recipe loss*, la *eos loss* e la *cardinality penality*, descritte con maggiore dettaglio nel capitolo precedente 2.3.1, per le quali vengono utilizzati i rispettivi pesi di 1000.0, 0, 1.0 e 1.0. Per la *recipe loss* viene utilizzato un peso pari a 0, in quanto non si è interessati alla valutazione delle istruzioni generate e di conseguenza si vuole evitare che queste possano influenzare il training del modello. Nelle Fig. 3.51 e 3.52 vengono invece mostrati i risultati ottenuti in termini di intersection over Union (IoU) e cardinality prediction error, rispettivamente. Per entrambi è possibile osservare delle prestazioni inferiori rispetto a quelle ottenute negli esperimenti precedenti, nonostante, come detto in precedenza sia presente una riduzione nella differenza tra training e validation set. Nello specifico la Iou sul training set raggiunge un valore massimo di 42.5%, in confronto al 60% ottenuto precedentemente e anche per il cardinality prediction error il valore minimo raggiunto è 8.4, in confronto al 6.6 precedentemente ottenuto. Infine, la Fig. 3.53 riporta i valori della metrica

F1 sul validation set, mostrando una crescita lineare nelle prime epoche, per poi stabilizzarsi intorno al valore 0.46.

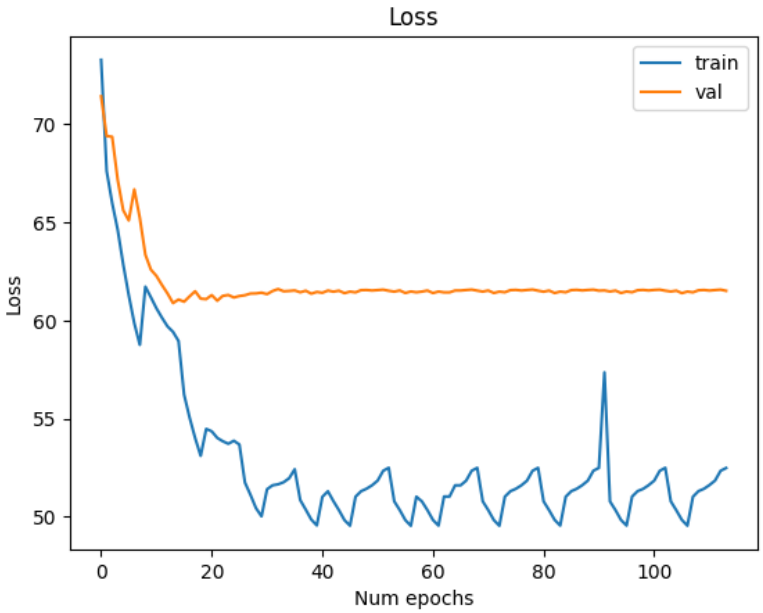


Figura 3.49: Andamento della Loss, sul training e validation set, durante l'addestramento del modello unicamente su ricette italiane

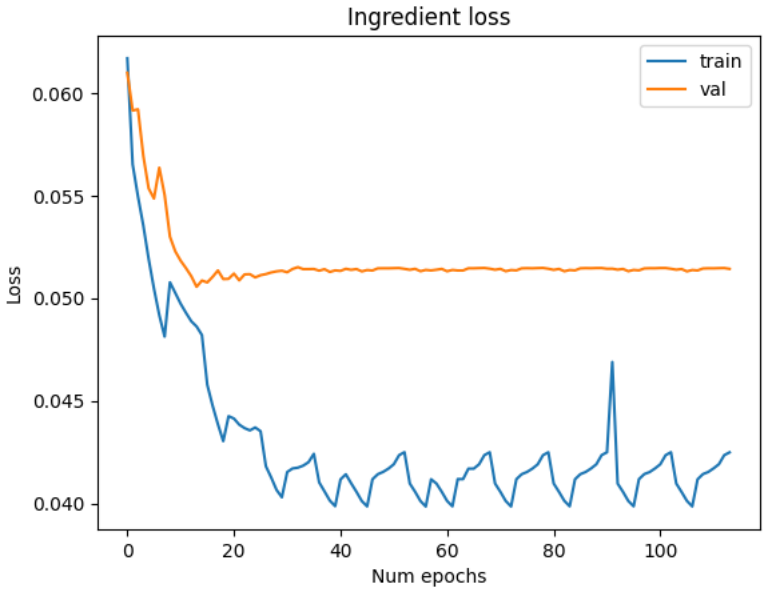


Figura 3.50: Andamento della Ingredient loss, sul training e validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane

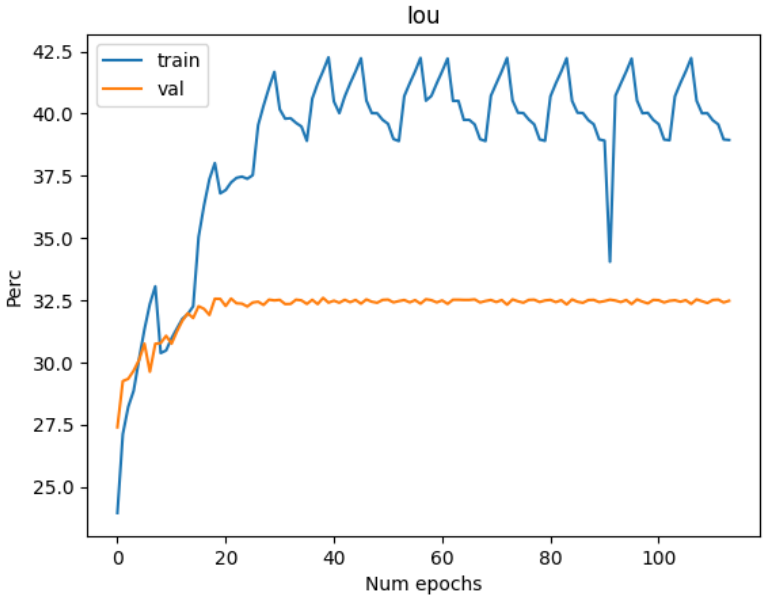


Figura 3.51: Andamento della Iou, sul training e validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane

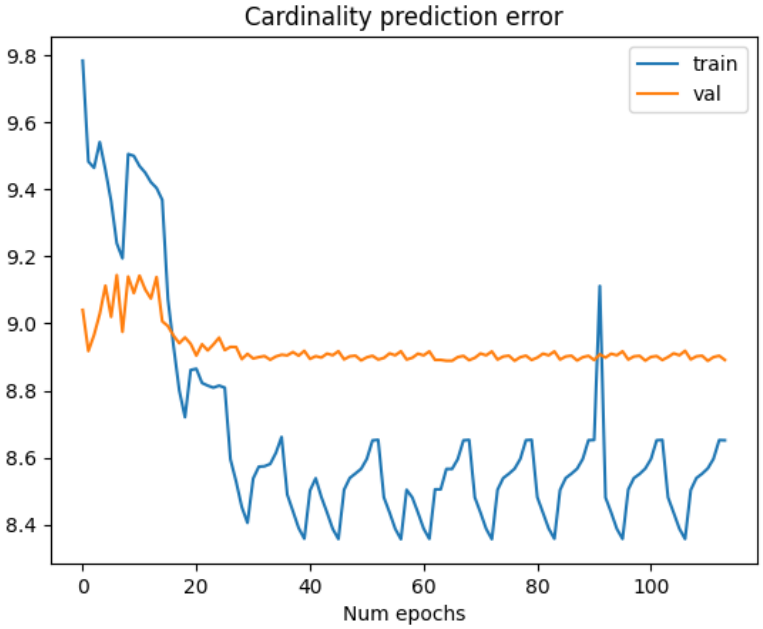


Figura 3.52: Andamento della Cardinality prediction error, sul training e validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane

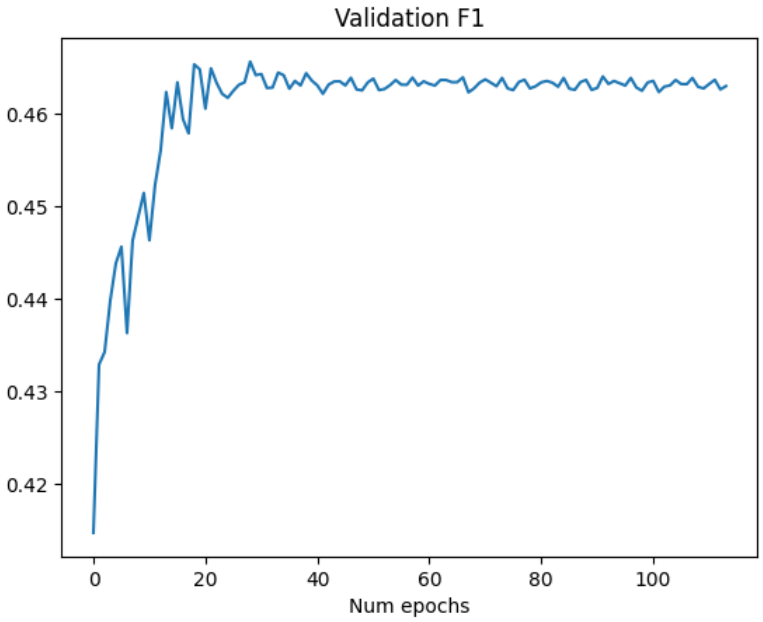


Figura 3.53: Andamento della F1, sul validation set, durante l'addestramento del modello sulla nuova versione del dataset delle ricette italiane

3.6 Approccio 5

3.6.1 Implementazione

Come ultimo approccio si è scelto di utilizzare un modello, rilasciato nel 2021, che attraverso l'utilizzo di tecniche innovative, ovvero i trasformatori, rappresenta lo state of the art per il task di associazione tra immagine e ricetta (image-to-recipe). Uno dei principali vantaggi di questa architettura risiede nella sua capacità di poter elaborare, oltre alle coppie immagine-ricetta, anche le singole componenti delle di quest'ultima, senza che queste siano associate a delle immagini, garantendo al modello la possibilità, sia di avvicinare le componenti, come per esempio il titolo e gli ingredienti, se queste appartengono alla stessa ricetta, sia di allontanarle in caso contrario. Per sfruttare al meglio questa capacità si è deciso di utilizzare tutte le ricette del dataset Recipe1M, considerandole come dati senza l'associazione diretta con un'immagine. Per evitare l'insorgere di un problema di cambiamento di dominio (domain shift), verificatosi in alcuni degli esperimenti precedenti, e per poter sfruttare al meglio i dati di recipe1M, si è scelto di ricorrere alla traduzione, in lingua inglese, di tutte le ricette italiane raccolte, attraverso la libreria python *googletrans* [50]. Una volta terminata questa operazione, i dati sono stati pre-processati, attraverso un opportuno script, messo a disposizione dagli autori, che

ha permesso di partizionare sia i dati associati con delle immagini, sia le sole ricette, e di creare un vocabolario, composto dalle parole estratte dai dati, che verrà utilizzato in seguito durante l'addestramento del modello. Ci si aspetta da questa nuova architettura un notevole aumento delle prestazioni rispetto agli esperimenti precedenti.

3.6.2 Esperimenti e metriche

Per poter valutare correttamente le capacità di questo nuovo modello sono stati eseguiti diversi esperimenti:

- Il primo, identificato come $Model_{15\_}rec_{v1}$, eseguito sulla prima versione del dataset delle ricette italiane, mantenendo l'architettura proposta dagli autori invariata, lasciando quindi a 15 il numero massimo di parole che compongono gli ingredienti e le istruzioni di preparazione delle diverse ricette;
- Il secondo, identificato come $Model_{100\_}rec_{v1}$, eseguito sempre sulla prima versione del dataset delle ricette italiane, modificando però in parte il modello proposto dagli autori, impostando a 25 in numero massimo di ingredienti e istruzioni considerate per singola ricetta e a 100 il numero massimo di parole che compongono gli ingredienti e le istruzioni di preparazione delle diverse ricette;
- il terzo, identificato come $Model_{125\_}rec_{v2}$, eseguito sulla seconda versione del dataset delle ricette italiane, aumentando a 125 il numero massimo di parole che compongono gli ingredienti e le istruzioni di preparazione delle diverse ricette;
- l'ultimo invece, identificato come $Model_{R50\_}rec_{v2}$, è stato eseguito sulla seconda versione del dataset delle ricette italiane, partendo però dal miglior checkpoint fornito dagli autori;

Per la valutazione di questi esperimenti sono state impiegate le stesse metriche degli esperimenti precedenti ovvero, median rank (MedR) e recall rate (R@K), in particolare R@1, R@5 e R@10. In questo caso però, lavorando su due versioni differenti del dataset delle ricette italiane, per $Model_{15\_}rec_{v1}$ e $Model_{100\_}rec_{v1}$ i valori sono calcolati, rispetto a tutti gli elementi del validation set, ovvero 784 e ad sottoinsieme di 500 elementi; mentre, per $Model_{125\_}rec_{v2}$ e $Model_{R50\_}rec_{v2}$, le metriche sono calcolate rispetto a tutti gli elementi del validation set, ovvero 1592, un sottoinsieme di 1000 coppie ricetta-immagine ed un sottoinsieme di 500 elementi, per permettere un confronto con gli esperimenti citati in precedenza e quelli delle precedenti sezioni. Le valutazioni vengono effettuate 10 volte, su sottoinsiemi scelti casualmente, considerando poi i valori medi.

3.6.3 Risultati

Per la valutazione delle performance di questo nuovo modello e per permettere un confronto con gli esperimenti precedenti, sono stati realizzati una serie di grafici, che riportano i dati più importanti raccolti durante l'addestramento dei diversi modelli descritti in precedenza. Viene sottolineato che, a differenza degli esperimenti svolti finora, il numero di punti utilizzati nei grafici per il validation set è uguale a quello per il training set, questo perché, durante l'addestramento, la valutazione sul validation set viene effettuata ogni singola epoca. La Fig. 3.54 mostra i valori della *loss*, sia sul training, che sul validation set. Osservando l'andamento di entrambe le curve è possibile notare come si verifichi un decremento costante sul training set, raggiungendo valori prossimi allo 0.0, mentre sul validation set, dopo un decremento iniziale, i valori si consolidano intorno a 0.15. Questo effetto può essere attribuito ad un problema di overfitting, ampiamente descritto in precedenza. Tuttavia, la capacità di questo modello di utilizzare dati, anche se non associati direttamente con delle immagini, ha permesso di ridurre notevolmente il forte impatto che questo fenomeno presentava negli esperimenti precedenti. Le Fig. 3.55 e 3.56 riportano invece l'andamento della *recipe loss* e della *paired loss*, rispettivamente. I grafici risultano molto simili a quello descritto in precedenza, questo perché la loss non è altro che una somma pesata delle due funzioni citate, utilizzando 1.0 come peso per entrambe. Le Fig. 3.57 e 3.58 mostrano l'andamento della *medR* rispettivamente su training e validation set. In entrambi viene riportato il valore della metrica rispetto all'intero validation set e rispetto ad un sottoinsieme di 500 elementi. Si può notare che, nel caso del training set, in poche epoche il modello riesce a raggiungere un valore di 1.0 rimanendo poi stabile su esso, mentre nel caso del validation set, si osserva un miglioramento progressivo dei valori, ma con prestazioni decisamente inferiori rispetto al training set. Anche in questo caso quindi è possibile osservare il fenomeno di overfitting. Infine, nelle Fig. 3.59 e 3.60, sono presentati i valori della metrica di *recall rate*, per il training set e validation set rispettivamente. Anche in questo caso valgono le osservazioni fatte per la metrica precedente. Tuttavia è necessario sottolineare come, già da questo primo esperimento si ottengano delle performance decisamente superiori rispetto a quelle analizzate finora. I dati riportati fino a questo momento fanno riferimento all'esperimento $Model_{15}\_rec_{v1}$. I grafici delle Fig. 3.61, 3.62, 3.63, 3.64, 3.65, 3.66 e 3.67 si riferiscono invece all'esperimento denominato $Model_{100}\_rec_{v1}$, il quale, raggiunge delle performance migliori rispetto al precedente, mostrando come modificare a 100 il numero massimo di parole che compongono gli ingredienti e le istruzioni di preparazione delle diverse ricette, aiuti il modello a distinguere con maggiore precisione le ricette e di conseguenza, ad associarle alle relative immagini con maggiore precisione. Questi primi due esperimenti utilizzano la prima versione del dataset delle ricette italiane, mentre quelli descritti nel seguito

si riferiscono alla seconda versione dei dati. In particolare, le Fig. 3.68, 3.69, 3.70, 3.71, 3.72, 3.73 e 3.74, mostrano i risultati relativi al $Model_{125}\_rec_{v2}$, le cui performance, per un sottoinsieme di 500 elementi del validation set sono molto simili a quelle del $Model_{100}\_rec_{v1}$, evidenziando come il modello riesce ad associare in maniera efficace le immagini e le ricette, nonostante durante l'addestramento vengano impiegate diverse immagini per ogni singola ricetta. Inoltre, anche in questo caso i valori ottenuti, rispetto agli approcci precedenti, sono decisamente superiori. L'ultimo esperimento considerato è il $Model_{R50}\_rec_{v2}$, i cui grafici sono riportati nelle Fig. 3.75, 3.76, 3.77, 3.78, 3.79, 3.80 e 3.81. Analizzando i dati è possibile notare come questo modello ottenga performance decisamente inferiori rispetto a quelle evidenziate nei casi precedenti. Questi risultati sono attribuibili ad un problema di cambiamento di dominio (domain shift), ampiamente descritto in precedenza, che non permette al modello di adattarsi efficacemente alla nuova distribuzione di dati. Per concludere, le migliori performance ottenute dai diversi esperimenti sono state riportate in due tabelle: la prima (3.5) riguardante la prima versione del database delle ricette italiane; la seconda (3.6) focalizzata sulla seconda versione dei dati. Inoltre, per riassumere e confrontare tutti i risultati ottenuti nei diversi approcci, sono state realizzate le tabelle 3.7 e 3.8.

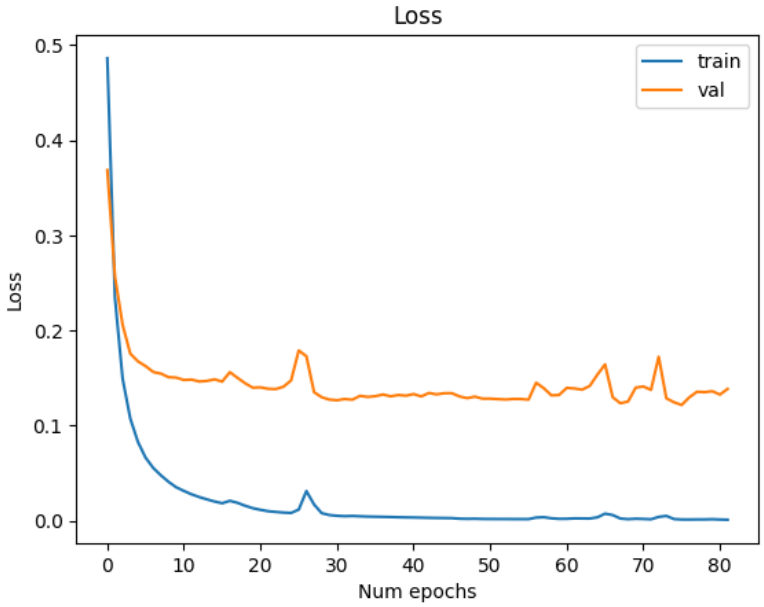


Figura 3.54: Andamento della loss, sul training e validation set, durante l'addestramento del modello $Model_{15}\_rec_{v1}$

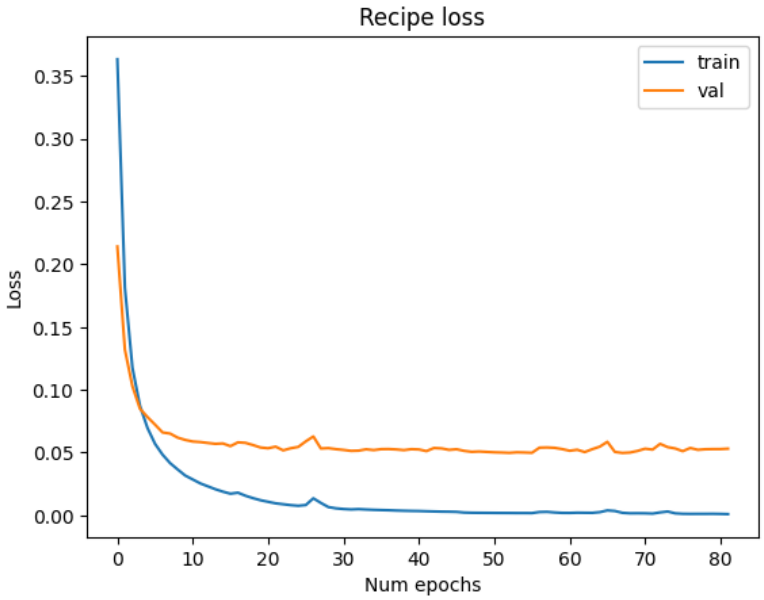


Figura 3.55: Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello $Model_{15\_}rec_{v1}$

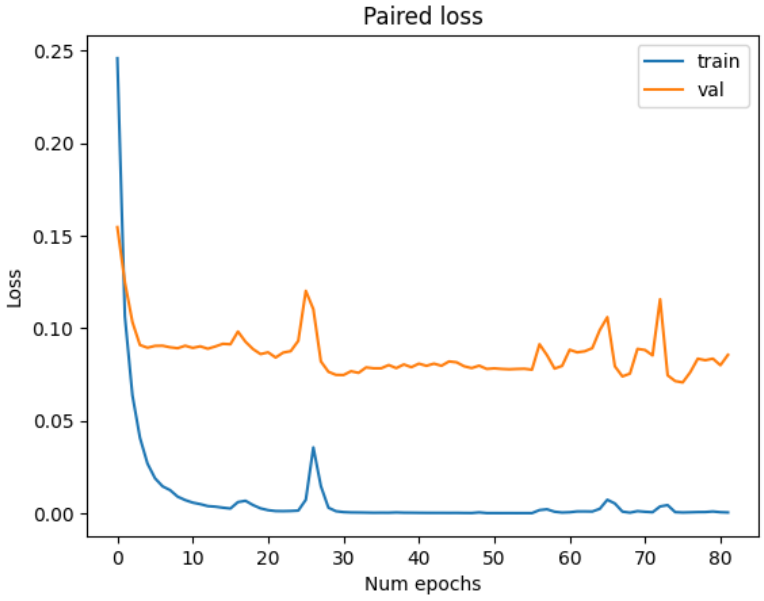


Figura 3.56: Andamento della paired loss, sul training e validation set, durante l'addestramento del modello $Model_{15\_}rec_{v1}$

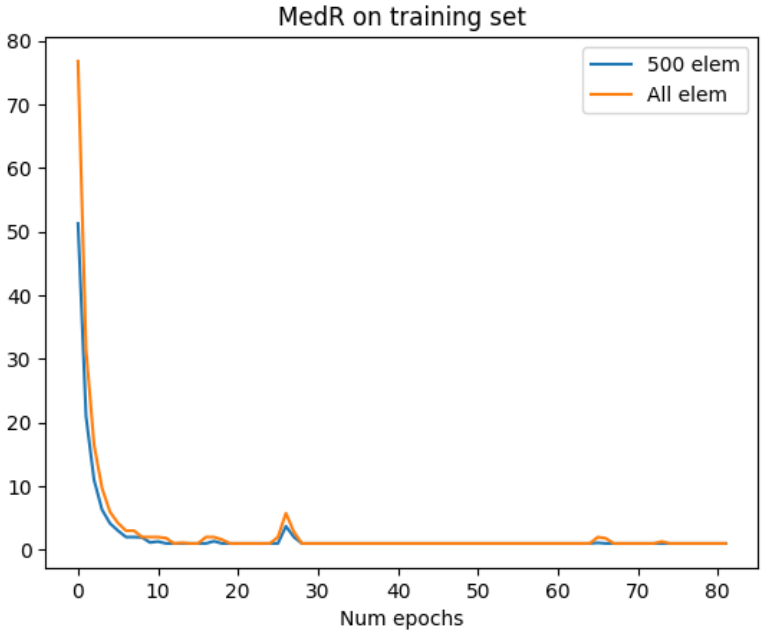


Figura 3.57: Andamento della medR, sul training set, durante l'addestramento del modello $Model_{15\_}rec_{v1}$

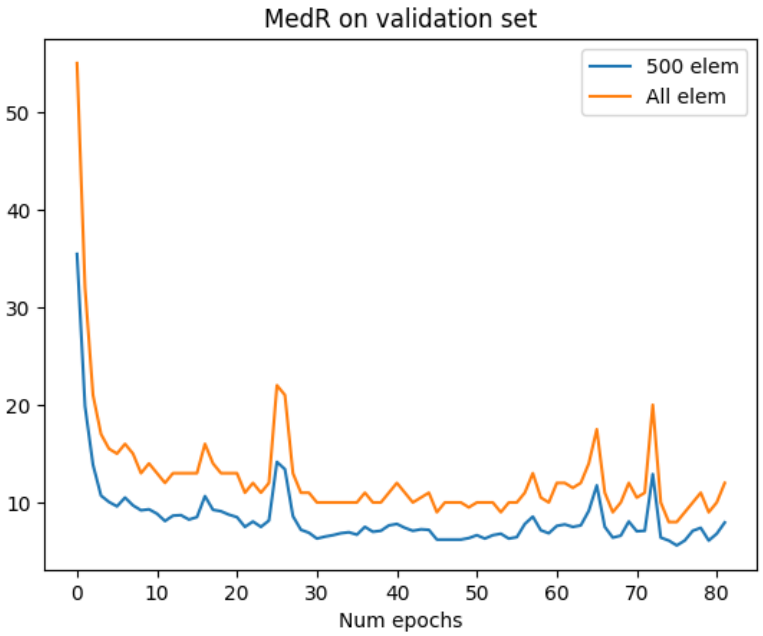


Figura 3.58: Andamento della medR, sul validation set, durante l'addestramento del modello $Model_{15}\_rec_{v1}$

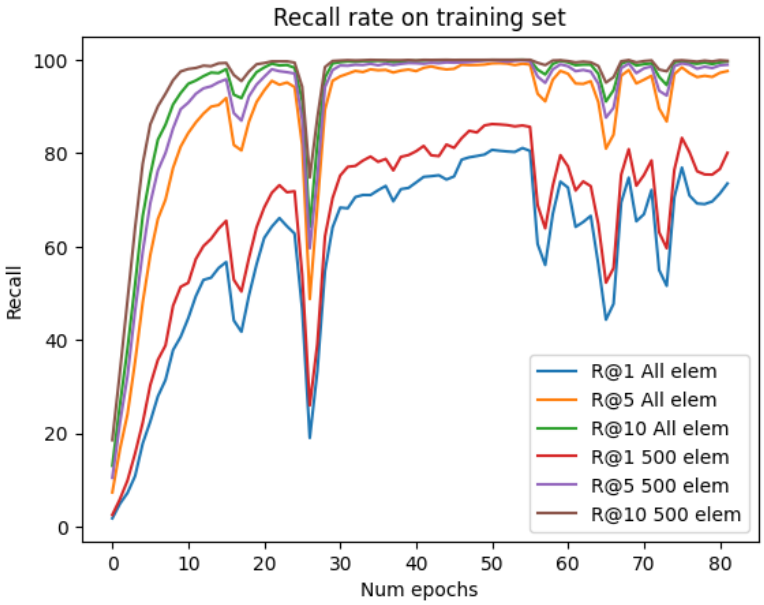


Figura 3.59: Andamento del recall rate, sul training set, durante l'addestramento del modello $Model_{15\_}rec_{v1}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette

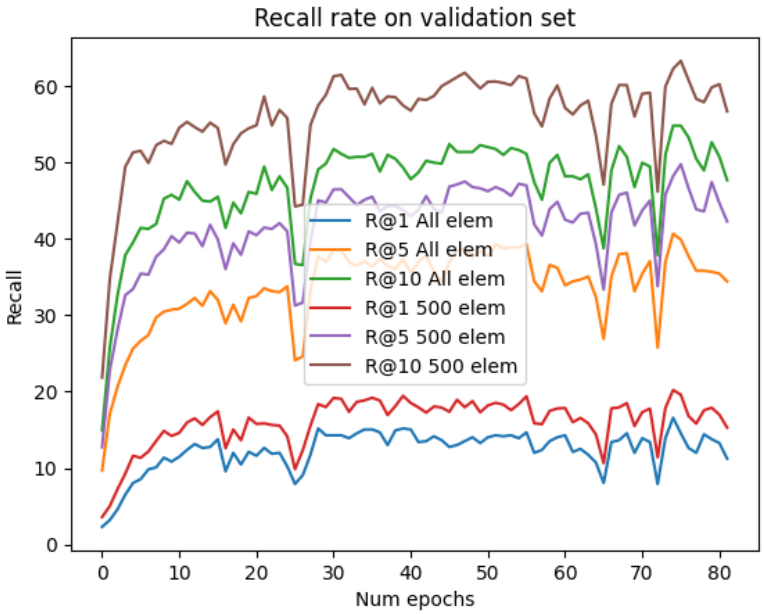


Figura 3.60: Andamento del recall rate, sul validation set, durante l'addestramento del modello $Model_{15}\_rec_{v1}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette

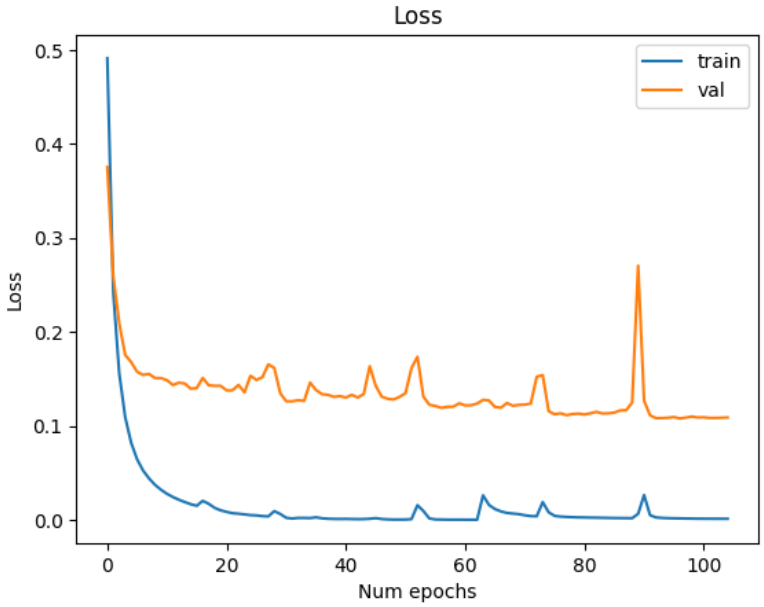


Figura 3.61: Andamento della loss, sul training e validation set, durante l'addestramento del modello $Model_{100\_rec_{v1}}$

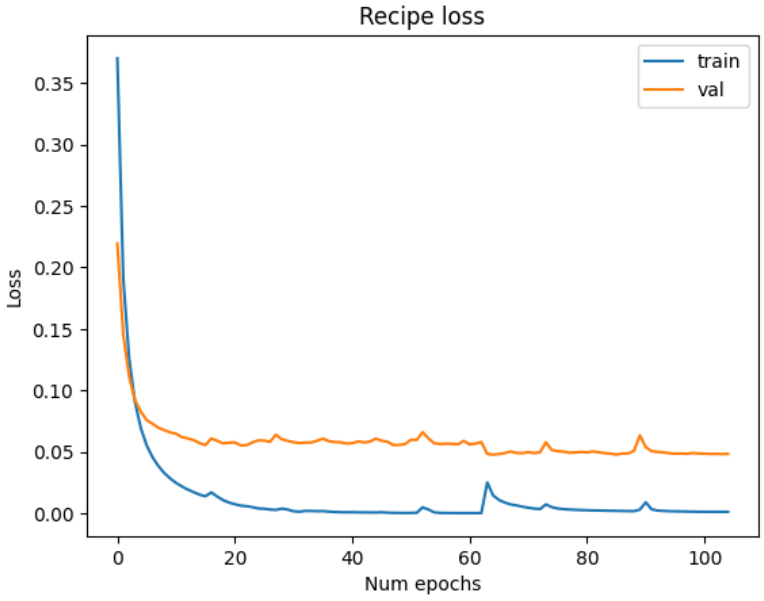


Figura 3.62: Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello $Model_{100\_rec_{v1}}$

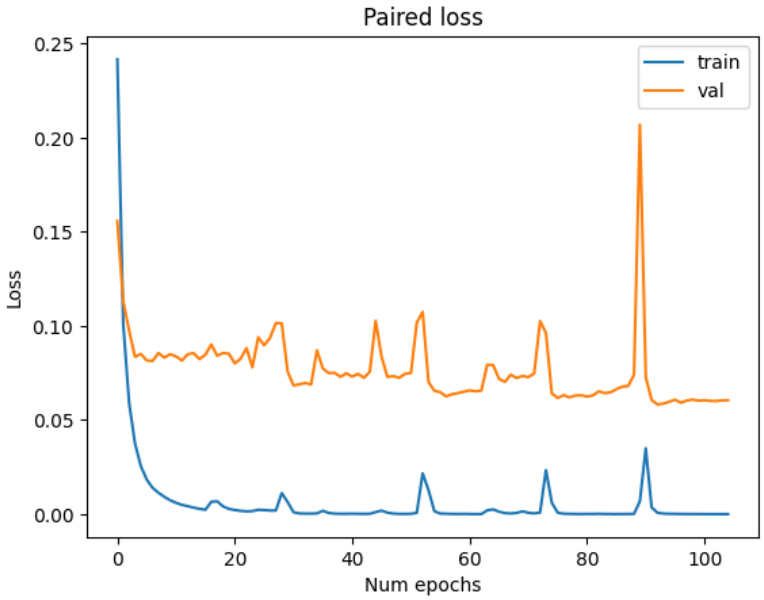


Figura 3.63: Andamento della paired loss, sul training e validation set, durante l'addestramento del modello $Model_{100\_rec_{v1}}$

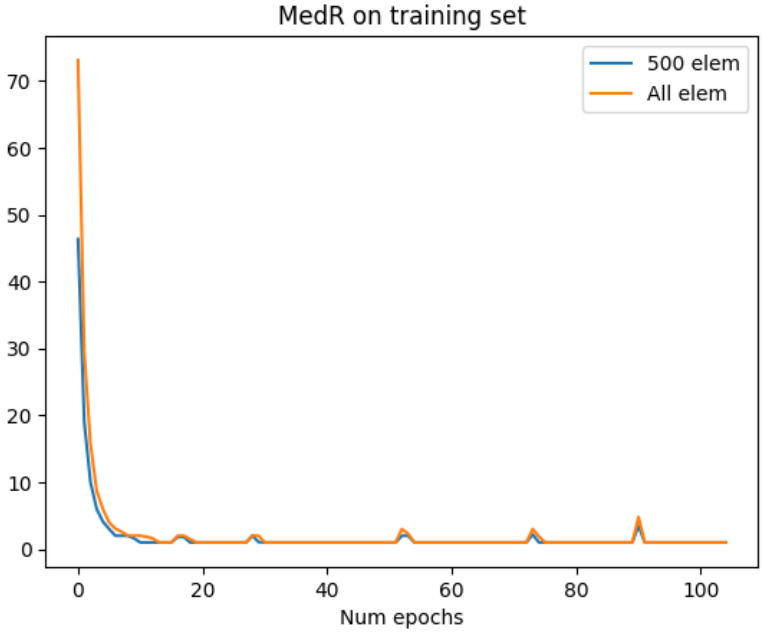


Figura 3.64: Andamento della medR, sul training set, durante l'addestramento del modello $Model_{100\_rec_{v1}}$

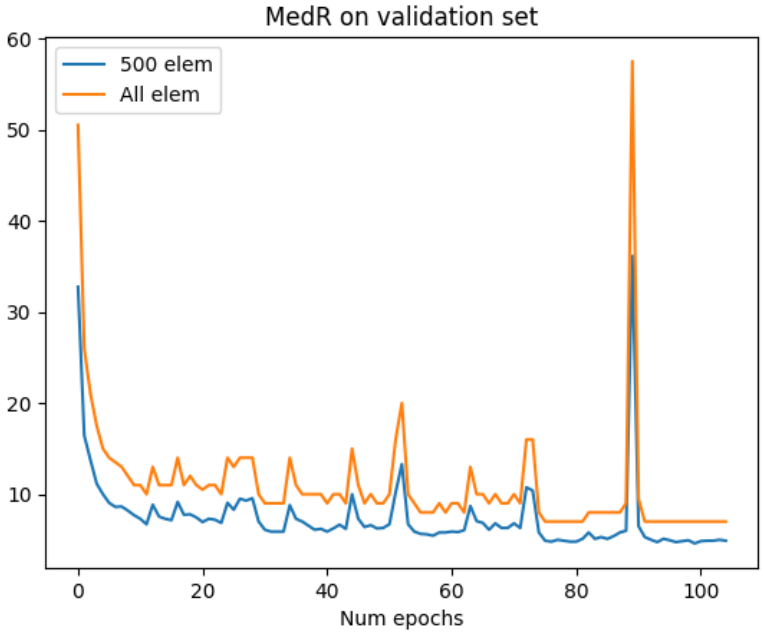


Figura 3.65: Andamento della medR, sul validation set, durante l'addestramento del modello $Model_{100\_rec_{v1}}$

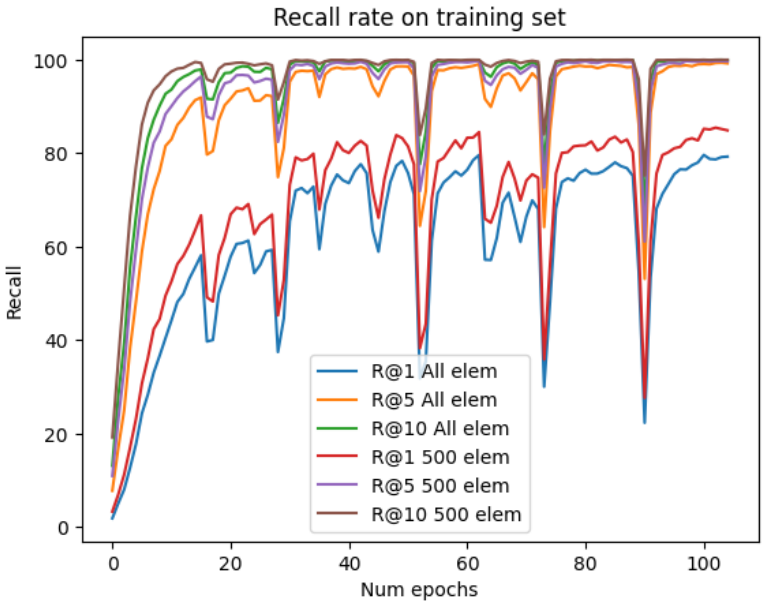


Figura 3.66: Andamento del recall rate, sul training set, durante l'addestramento del modello $Model_{100\_rec_{v1}}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette

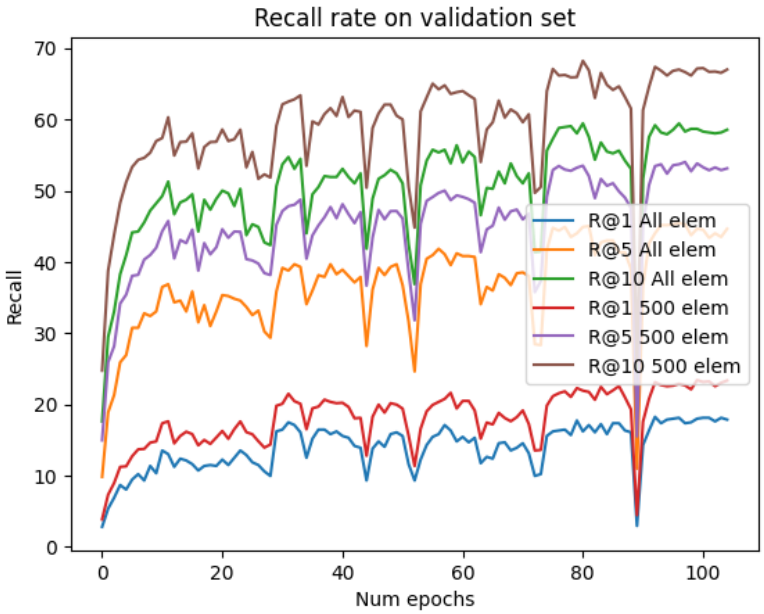


Figura 3.67: Andamento del recall rate, sul validation set, durante l'addestramento del modello $Model_{100\_rec_{v1}}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette

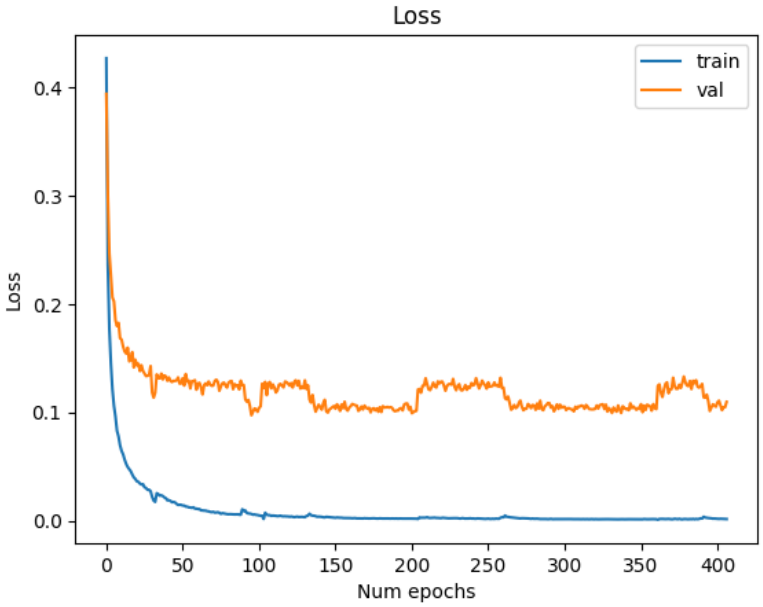


Figura 3.68: Andamento della loss, sul training e validation set, durante l'addestramento del modello $Model_{125\_}rec_{v2}$

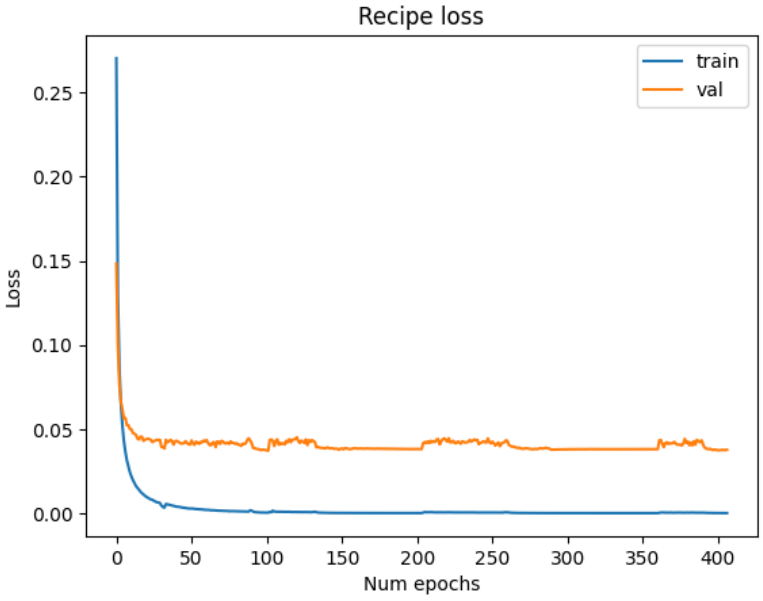


Figura 3.69: Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello $Model_{125\_}rec_{v2}$

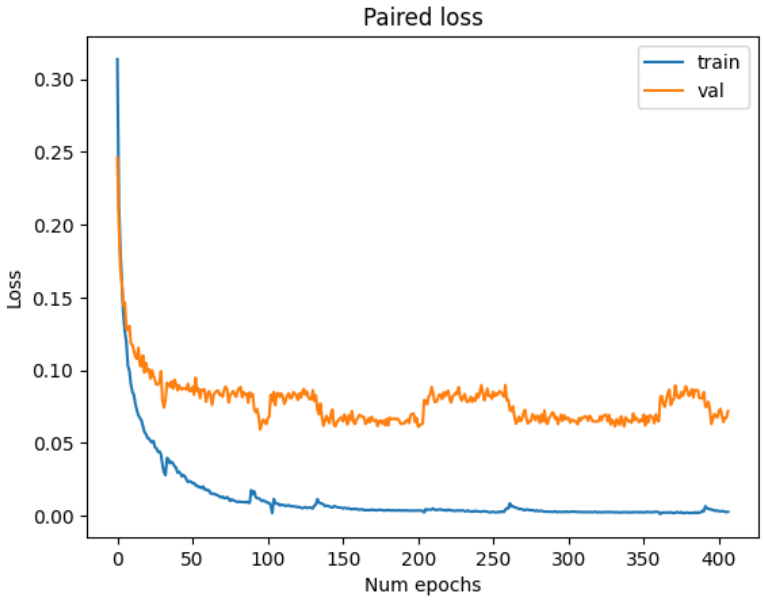


Figura 3.70: Andamento della paired loss, sul training e validation set, durante l'addestramento del modello $Model_{125\_}rec_{v2}$

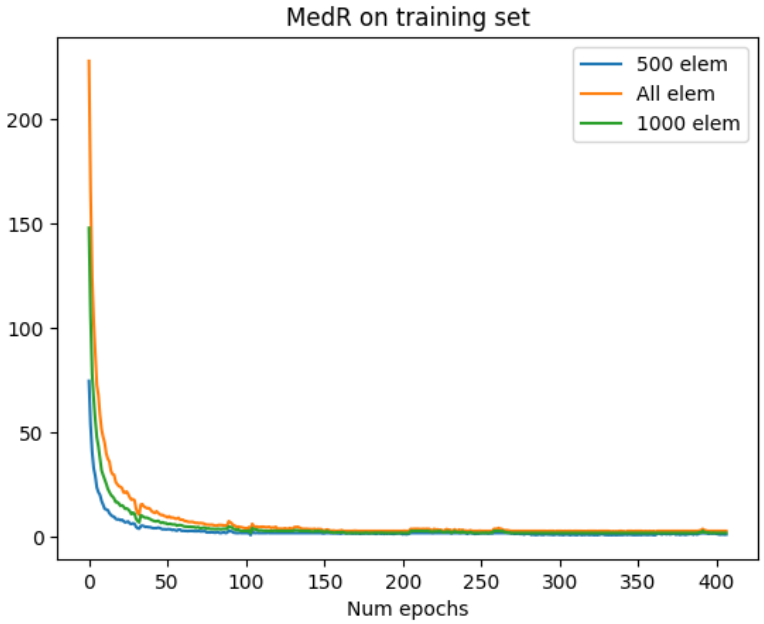


Figura 3.71: Andamento della medR, sul training set, durante l'addestramento del modello $Model_{125\_}rec_{v2}$

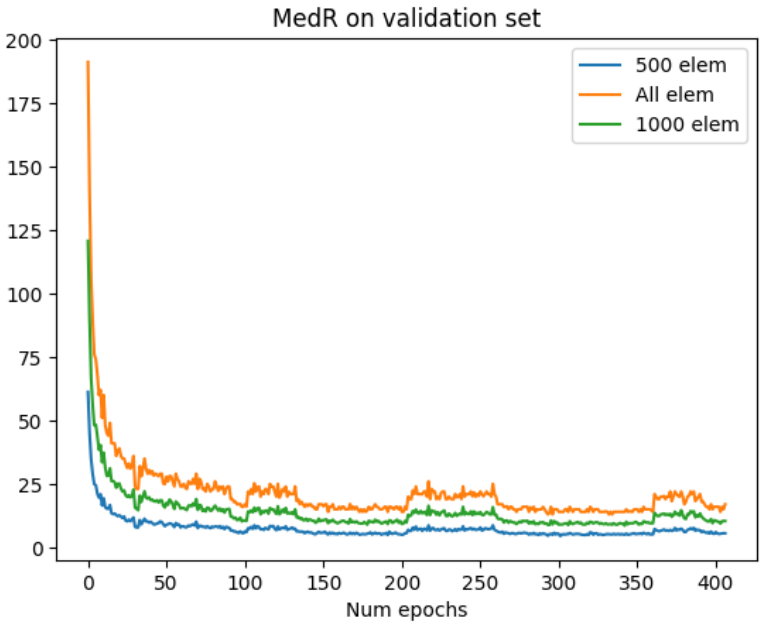


Figura 3.72: Andamento della medR, sul validation set, durante l'addestramento del modello $Model_{125}\_rec_{v2}$

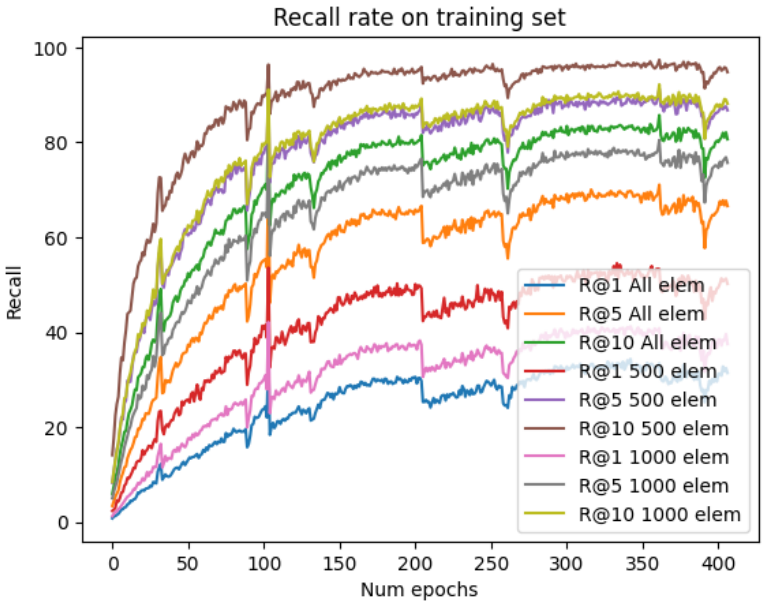


Figura 3.73: Andamento del recall rate, sul training set, durante l'addestramento del modello $Model_{125\_rec_{v2}}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette

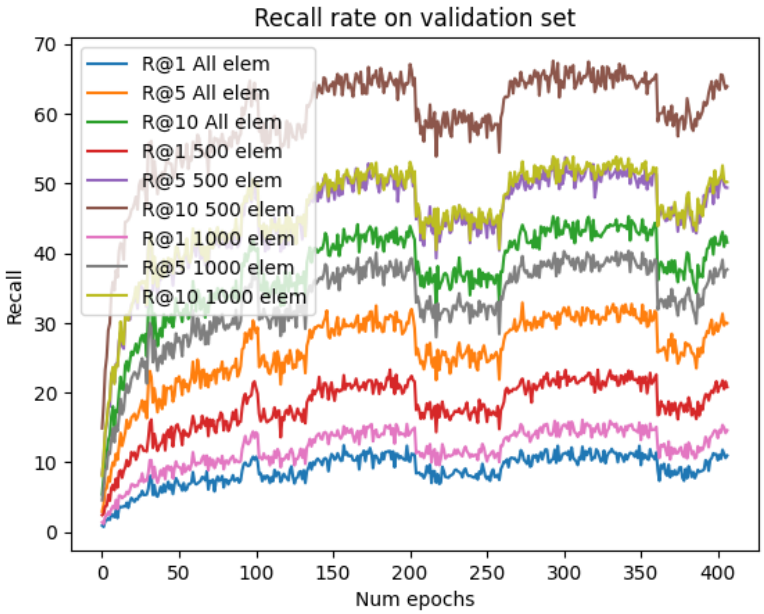


Figura 3.74: Andamento del recall rate, sul validation set, durante l'addestramento del modello $Model_{125\_rec_{v2}}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette

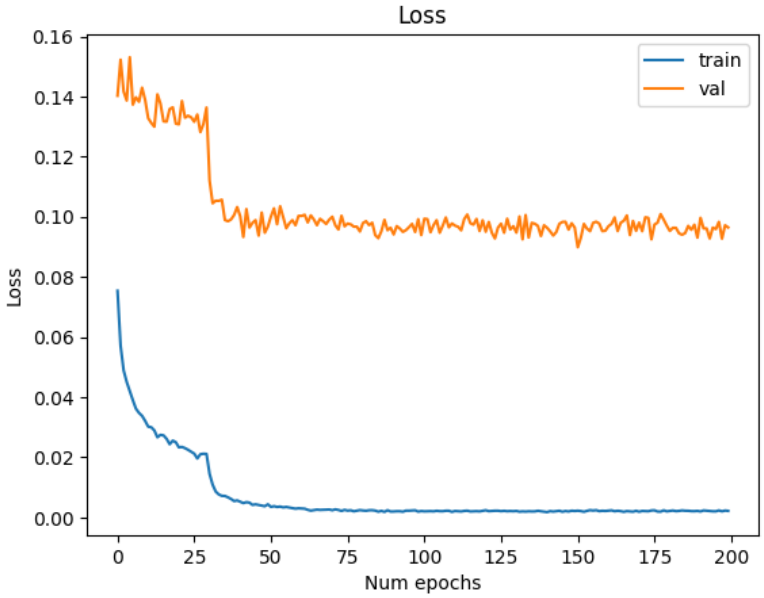


Figura 3.75: Andamento della loss, sul training e validation set, durante l'addestramento del modello $Model_{R50\_rec_{v2}}$

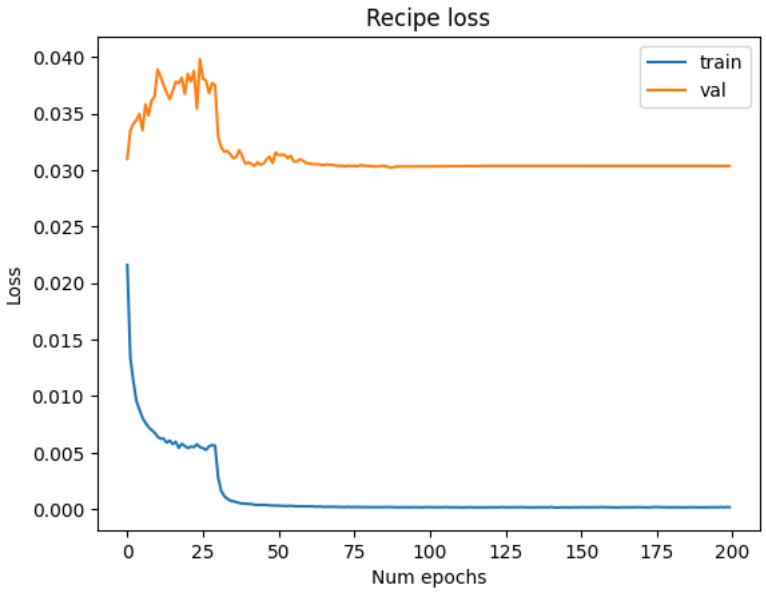


Figura 3.76: Andamento della recipe loss, sul training e validation set, durante l'addestramento del modello $Model_{R50\_rec_{v2}}$

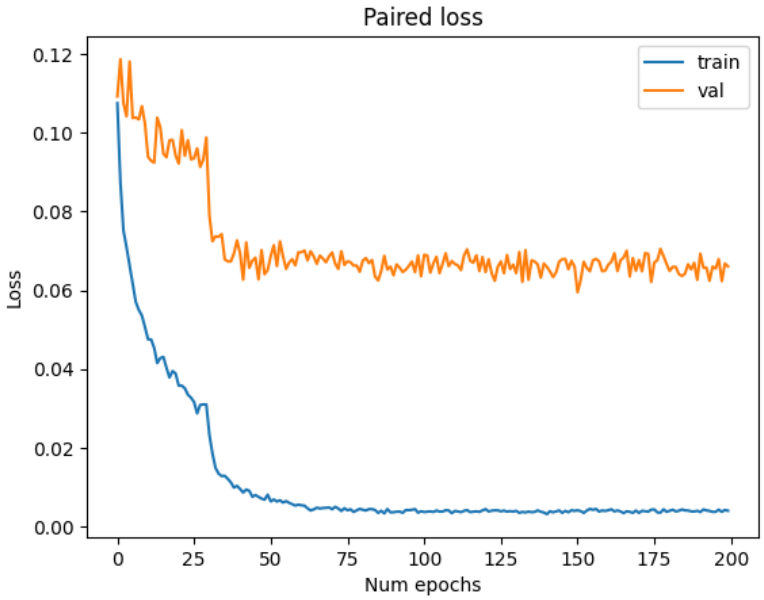


Figura 3.77: Andamento della paired loss, sul training e validation set, durante l'addestramento del modello $Model_{R50\_rec_{v2}}$

Figura 3.78: Andamento della medR, sul training set, durante l'addestramento del modello $Model_{R50\_rec_{v2}}$

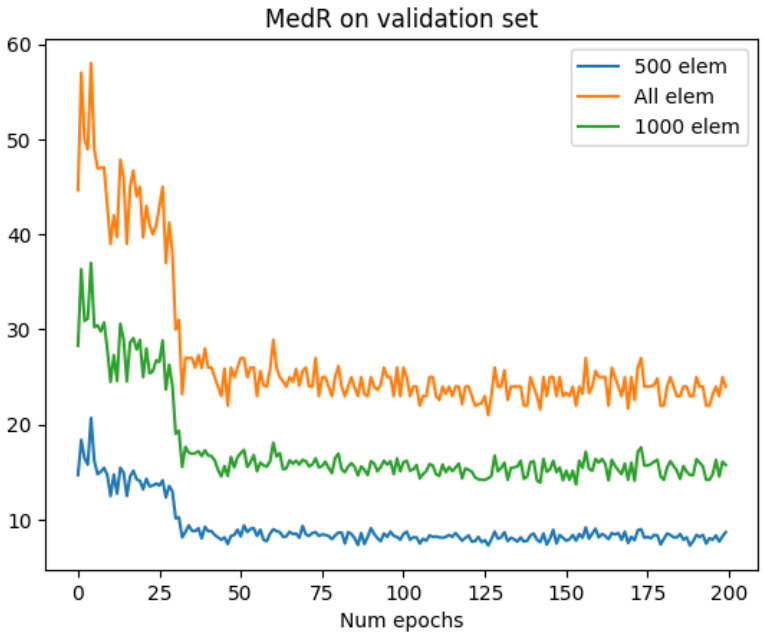


Figura 3.79: Andamento della medR, sul validation set, durante l'addestramento del modello $Model_{R50\_rec_{v2}}$

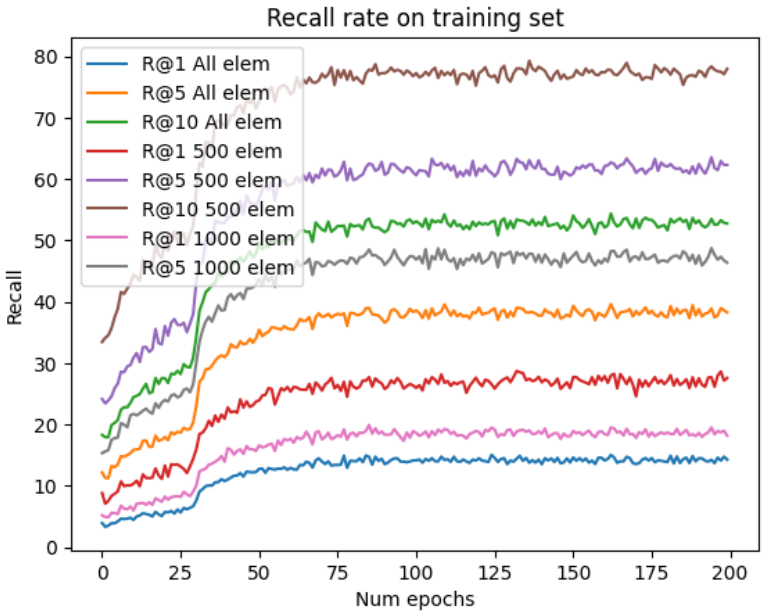


Figura 3.80: Andamento del recall rate, sul training set, durante l'addestramento del modello $Model_{R50\_rec_{v2}}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette

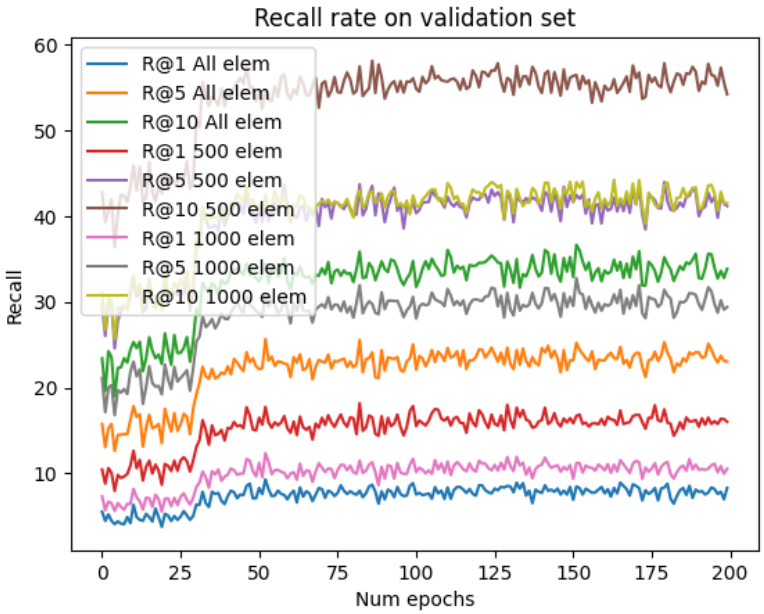


Figura 3.81: Andamento del recall rate, sul validation set, durante l'addestramento del modello $Model_{R50\_rec_{v2}}$. Questa metrica è stata calcolata su R@1, R@5 e R@10, ovvero sulla prima, sulle prime cinque e sulle prime dieci posizioni della classifica di similarità tra un'immagine di input e le ricette

Tabella 3.5: Risultati degli esperimenti $Model_{15\_rec_{v1}}$ e $Model_{100\_rec_{v1}}$

	$Model_{100\_rec_{v1}}$			$Model_{15\_rec_{v1}}$		
	Train	**Val**	**Test**	**Train**	**Val**	**Test**
Loss	0.0004	0.0792	0.0857	0.0006	0.1245	0.1287
Recipe Loss	0.0	0.0381	0.0369	0.0	0.0533	0.0497
Paired Loss	0.0004	0.0411	0.0487	0.0006	0.0713	0.00751
MedR (Tot)	1.0	7.0	8.0	1.0	8.0	9.0
R@1 (Tot)	69.0%	18.2%	14.9%	74.4%	16.5%	13.3%
R@5 (Tot)	96.8%	44.0%	41.9 %	99.0%	40.5%	39.7%
R@10 (Tot)	99.6%	59.2%	57.2%	99.5%	54.8%	52.6%
MedR (500 elem)	1.0	4.9	5.1	1.0	5.9	6.3
R@1 (500 elem)	77.1%	22.7%	19.4%	79.9%	20.2%	17.2%
R@5 (500 elem)	99.0%	53.2%	36.9%	98.1%	48.8%	46.7%
R@10 (500 elem)	99.8%	67.5%	66.9%	99.8%	63.1%	59.9%

Tabella 3.6: Risultati degli esperimenti $Model_{125\_}recs_{v2}$ e $Model_{R50\_}rec_{v2}$

	$Model_{125\_}rec_{v2}$			$Model_{R50\_}rec_{v2}$		
	Train	**Val**	**Test**	**Train**	**Val**	**Test**
Loss	0.0046	0.0813	0.0813	0.0210	0.0998	0.1078
Recipe Loss	0.0	0.0302	0.0286	0.0002	0.0295	0.0292
Paired Loss	0.0046	0.0511	0.0527	0.0208	0.070	0.0786
MedR (Tot)	4.0	14.0	16.0	14.55	26.0	26.6
R@1 (Tot)	23.0%	12.3%	10.9%	10.6%	7.6%	6.4%
R@5 (Tot)	56.2%	30.5%	30.5%	30.3%	22.7%	20.8%
R@10 (Tot)	72.2%	42.3%	41.4%	43.0%	33.1%	32.2%
MedR (1000 elem)	3.0	9.4	10.1	9.25	16.4	17.3
R@1 (1000 elem)	29.9%	15.5%	14.4%	14.2%	10.0%	8.8%
R@5 (1000 elem)	66.2%	37.9%	36.7%	38.9%	28.4%	27.6%
R@10 (1000 elem)	81.1%	52.6%	50.8%	53.1%	41.2%	39.7%
MedR (500 elem)	2.0	5.2	5.5	4.9	8.6	9.0
R@1 (500 elem)	41.9%	22%	21.6%	21.8%	14.9%	14.1%
R@5 (500 elem)	80.2%	51.5%	49.9%	52.5%	39.8%	39.6%
R@10 (500 elem)	91.8%	65.9%	64.9%	67.8%	54.2%	53.2%

Tabella 3.7: Tabella riassuntiva esperimenti sulla prima versione del dataset delle ricette italiane

	All elements				*500 elements*			
	MedR	**R@1**	**R@5**	**R@10**	**MedR**	**R@1**	**R@5**	**R@10**
Img2recipe From 0								
Val	20.0	8.5%	24.6%	36.8%	13.15	11.7%	30.9%	44.9%
Test	23.0	9.6%	25.0%	36.6%	15.0	12.5%	30.0%	42.9%
Img2recipe From check								
Val	62.5	5.2%	13.5%	21.1%	40.6	6.5%	18.7%	27.9%
Test	67.0	5.5%	15.0%	22.2%	42.5	7.2%	19.5%	26.8%
Img2recipe no instr								
Val	47.0	5.4%	14.6%	22.6%	29.8	7.0%	18.9%	28.5%
Test	45.0	4.4%	14.7%	24.3%	28.7	6.2%	20.6%	28.8%
Img2recipe from check eng								
Val	23.0	10.4%	22.5%	31.9%	13.65	13.5%	30.2%	43.9%
Test	30.0	9.8%	23.5%	32.0%	18.0	13.0%	30.0%	39.9%
$\text{Model}_{15}\_\text{rec}_{v1}$								
Val	8.0	16.5%	40.5%	54.8%	5.9	20.2%	48.8%	63.1%
Test	9.0	13.3%	39.7%	52.6%	6.3	17.2%	46.7%	59.9%
$\text{Model}_{100}\_\text{rec}_{v1}$								
Val	7.0	18.2%	44.0%	59.2%	4.9	22.7%	53.2%	67.5%
Test	8.0	14.9%	41.9%	57.2%	5.1	19.4%	36.9%	66.9%

Tabella 3.8: Tabella riassuntiva esperimenti sulla seconda versione del dataset delle ricette italiane

	All elements				*1000 elements*				*500 elements*			
	MedR	**R@1**	**R@5**	**R@10**	**MedR**	**R@1**	**R@5**	**R@10**	**MedR**	**R@1**	**R@5**	**R@10**
Img2recipe from 0												
Val	40.0	5.3%	16.5%	24.2%	26.4	7.6%	21.6%	31.4%	13.6	10.5%	30.1%	44.5%
Test	43.0	5.5%	17.1%	25.8%	27.2	7.8%	22.4%	32.9%	13.8	12.8%	32.5%	44.3%
$\text{Model}_{125}\_\text{rec}_{v2}$												
Val	14.0	12.3%	30.5%	42.3%	9.4	15.5%	37.9%	52.6%	5.2	22.0%	51.5%	65.9%
Test	16.0	10.9%	30.5%	41.4%	10.1	14.4%	36.7%	50.8%	5.5	21.6%	49.9%	64.9%
$\text{Model}_{R50}\_\text{rec}_{v2}$												
Val	26.0	7.6%	22.7%	33.1%	16.4	10.0%	28.4%	41.2%	8.6	14.9%	39.8%	54.2%
Test	23.6	6.4%	20.8%	32.2%	17.3	8.8%	27.6%	39.7%	9.0	14.1%	39.6%	53.2%

Tabella 3.9: Tabella riassuntiva dei modelli utilizzati

	Recipe1M	Recipe1M+	DB ricette italiane v1	DB ricette italiane v2
img2recipe	2.1	2.1	3.1	3.4
img2recipe no instruction	/	/	3.2	/
img2recipe from check eng	/	/	3.3	/
inverseCooking	2.1	/	3.2.3	3.5.3
transformer	2.1	/	3.5	3.6

Capitolo 4

Conclusioni

Il presente lavoro di tesi aveva come obiettivo quello di realizzare un classificatore in grado di associare correttamente un'immagine, passata come input, ad una ricetta e di conseguenza, di stimare, con buona precisione, gli ingredienti da cui è composto un piatto, semplificando così il complesso compito di tracciamento dei valori nutrizionali, per il mantenimento di una dieta sana. In seguito all'analisi approfondita dei diversi modelli proposti, si è giunti alla conclusione che, questo particolare task risulta estremamente complesso da ottimizzare in maniera efficace. Tuttavia nel corso degli anni, grazie a diverse innovazioni tecnologiche, è stato possibile ottenere dei risultati notevoli anche in questo campo. Il compito di questo lavoro di tesi è stato complicato ulteriormente dalla realizzazione di un nuovo dataset contenete unicamente ricette italiane, discostandosi così, dalle maggiori raccolte di dati utilizzate tipicamente in questo ambito e dovendosi confrontare con problematiche ad esso collegate, come l'overfitting e il cambiamento di dominio (domain shift). Fenomeni che, sono stati in parte attenuati nel corso dei diversi esperimenti, ma che, anche nel migliore dei casi, risultano comunque presenti. Nonostante queste difficoltà, il modello realizzato ottiene delle buone performance su entrambe le versioni del nuovo database, rendendo possibile proseguire nella realizzazione dell'applicativo descritto nelle fasi introduttive di questo lavoro. Inoltre, essendo l'image-to-recipe un task in continuo studio e evoluzione, nel corso degli anni sono stati rilasciati diversi lavori, non trattati nella presente tesi, ma che potrebbero apportare ulteriori migliorie a quanto realizzato finora. Uno tra tutti è *RecipeSnap* [51], un modello che riprende la tecnologia dei trasformatori e la ottimizza per essere utilizzata in maniera efficace anche su dispositivi mobile, senza quindi richiedere potenti GPU per l'implementazione e l'addestramento. Infine, nel tentativo di ridurre ulteriormente le problematiche legate alla nuova collezione di dati realizzata, si potrebbe procedere con la sua ulteriore espansione, raccogliendo nuovi dati dai diversi siti di ricette italiane, o ancora, utilizzare delle tecniche di domain adaptation, per ridurre la distanza tra la distribuzione dei

dati di pre-addestramento e quella dei dati di training, permettendo così l'uso più efficace dei checkpoint, già addestrati, forniti dagli autori dei diversi modelli.

Appendice A

Categorie Food-101

- apple pie
- baby back ribs
- baklava
- beef carpaccio
- beef tartare
- beet salad
- beignets
- bibimbap
- bread pudding
- breakfast burrito
- bruschetta
- caesar salad
- cannoli
- caprese salad
- carrot cake
- ceviche
- cheesecake

- cheese plate
- chicken curry
- chicken quesadilla
- chicken wings
- chocolate cake
- chocolate mousse
- churros
- clam chowder
- club sandwich
- crab cakes
- creme brulee
- croque madame
- cup cakes
- deviled eggs
- donuts
- dumplings
- edamame
- eggs benedict
- escargots
- falafel
- filet mignon
- fish and chips
- foie gras
- french fries
- french onion soup

- french toast
- fried calamari
- fried rice
- frozen yogurt
- garlic bread
- gnocchi
- greek salad
- grilled cheese sandwich
- grilled salmon
- guacamole
- gyoza
- hamburger
- hot and sour soup
- hot dog
- huevos rancheros
- hummus
- ice cream
- lasagna
- lobster bisque
- lobster roll sandwich
- macaroni and cheese
- macarons
- miso soup
- mussels
- nachos

- omelette
- onion rings
- oysters
- pad thai
- paella
- pancakes
- panna cotta
- peking duck
- pho
- pizza
- pork chop
- poutine
- prime rib
- pulled pork sandwich
- ramen
- ravioli
- red velvet cake
- risotto
- samosa
- sashimi
- scallops
- seaweed salad
- shrimp and grits
- spaghetti bolognese
- spaghetti carbonara

- spring rolls
- steak
- strawberry shortcake
- sushi
- tacos
- takoyaki
- tiramisu
- tuna tartare
- waffles

Bibliografia

[1] Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber e Antonio Torralba. «Learning Cross-modal Embeddings for Cooking Recipes and Food Images». In: (2017) (cit. alle pp. ii, 2, 4–6, 12, 26).

[2] Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber e Antonio Torralba. «Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images». In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2019) (cit. alle pp. ii, 2, 6, 7, 10–13, 29, 37, 64).

[3] Amaia Salvador, Michal Drozdzal, Xavier Giro-i-Nieto e Adriana Romero. «Inverse Cooking: Recipe Generation From Food Images». In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Giu. 2019 (cit. alle pp. iii, 2, 12, 17–21, 45).

[4] Amaia Salvador, Erhan Gundogdu, Loris Bazzani e Michael Donoser. «Revamping Cross-Modal Recipe Retrieval with Hierarchical Transformers and Self-supervised Learning». In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Giu. 2021 (cit. alle pp. iii, 2, 19, 24–28).

[5] V. Miller, P. Webb, F. Cudhea et al. «Global dietary quality in 185 countries from 1990 to 2018 show wide differences by nation, age, education, and urbanicity». In: *Nat Food* 3 (2022), pp. 694–702. DOI: 10.1038/s43016-022-00594-9. URL: https://doi.org/10.1038/s43016-022-00594-9 (cit. a p. 1).

[6] *FatSecret*. URL: https://www.fatsecret.it/ (cit. a p. 2).

[7] *ParseHub | Free web scraping - The most powerful web scraper*. URL: https://www.parsehub.com/ (cit. alle pp. 2, 30).

[8] *Google Image Scraper*. URL: https://github.com/ohyicong/Google-Image-Scraper (cit. alle pp. 2, 65).

[9] Lukas Bossard, Matthieu Guillaumin e Luc Van Gool. «Food-101 – Mining Discriminative Components with Random Forests». In: *Computer Vision – ECCV 2014*. A cura di David Fleet, Tomas Pajdla, Bernt Schiele e Tinne Tuytelaars. Cham: Springer International Publishing, 2014, pp. 446–461. ISBN: 978-3-319-10599-4 (cit. a p. 4).

[10] Austin Myers et al. «Im2Calories: Towards an Automated Mobile Vision Food Diary». In: *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 1233–1241 (cit. a p. 4).

[11] Yoshiyuki Kawano e Keiji Yanai. «FoodCam: A Real-Time Food Recognition System on a Smartphone». In: 74.14 (lug. 2015), pp. 5263–5287. ISSN: 1380-7501. DOI: 10.1007/s11042-014-2000-8. URL: https://doi.org/10.1007/s11042-014-2000-8 (cit. a p. 4).

[12] Ruihan Xu, Luis Herranz, Shuqiang Jiang, Shuang Wang, Xinhang Song e Ramesh Jain. «Geolocalized Modeling for Dish Recognition». In: *Trans. Multi.* 17.8 (ago. 2015), pp. 1187–1199. ISSN: 1520-9210. DOI: 10.1109/TMM.2015.2438717. URL: https://doi.org/10.1109/TMM.2015.2438717 (cit. a p. 4).

[13] Tomasz Kusmierczyk, Christoph Trattner e Kjetil Nørvåg. «Understanding and Predicting Online Food Recipe Production Patterns». In: *Proceedings of the 27th ACM Conference on Hypertext and Social Media*. HT '16. Halifax, Nova Scotia, Canada: Association for Computing Machinery, 2016, pp. 243–248. ISBN: 9781450342476. DOI: 10.1145/2914586.2914632. URL: https://doi.org/10.1145/2914586.2914632 (cit. a p. 4).

[14] Xin Wang, Devinder Kumar, Nicolas Thome, Matthieu Cord e Frederic Precioso. «Recipe Recognition with Large Multimodal Food Dataset». In: *Proceedings of IEEE International Conference on Multimedia & Expo (ICME), Workshop CEA*. Turin, Italy, giu. 2015. DOI: 10.1109/ICMEW.2015.7169757. URL: https://hal.archives-ouvertes.fr/hal-01196959 (cit. alle pp. 4, 9, 35).

[15] Jingjing Chen e Chong-wah Ngo. «Deep-Based Ingredient Recognition for Cooking Recipe Retrieval». In: MM '16. Amsterdam, The Netherlands: Association for Computing Machinery, 2016, pp. 32–41. ISBN: 9781450336031. DOI: 10.1145/2964284.2964315. URL: https://doi.org/10.1145/2964284.2964315 (cit. a p. 4).

[16] Olga Russakovsky et al. *ImageNet Large Scale Visual Recognition Challenge*. 2015. arXiv: 1409.0575 [cs.CV] (cit. alle pp. 6, 22).

[17] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva e A. Torralba. «Places: A 10 Million Image Database for Scene Recognition». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (apr. 2018), pp. 1452–1464 (cit. a p. 6).

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren e Jian Sun. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV] (cit. a p. 8).

[19] Karen Simonyan e Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV] (cit. alle pp. 8, 16, 22).

[20] Tomas Mikolov, Kai Chen, Greg Corrado e Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL] (cit. a p. 8).

[21] Ilya Sutskever, Oriol Vinyals e Quoc V. Le. «Sequence to Sequence Learning with Neural Networks». In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'14. Montreal, Canada: MIT Press, 2014, pp. 3104–3112 (cit. a p. 9).

[22] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun e Sanja Fidler. «Skip-Thought Vectors». In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'15. Montreal, Canada: MIT Press, 2015, pp. 3294–3302 (cit. a p. 9).

[23] Andrej Karpathy e Li Fei-Fei. *Deep Visual-Semantic Alignments for Generating Image Descriptions*. 2015. arXiv: 1412.2306 [cs.CV] (cit. a p. 10).

[24] Oriol Vinyals, Alexander Toshev, Samy Bengio e Dumitru Erhan. *Show and Tell: A Neural Image Caption Generator*. 2015. arXiv: 1411.4555 [cs.CV] (cit. a p. 10).

[25] *PyTorch*. URL: https://pytorch.org/ (cit. alle pp. 11, 18, 25).

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser e Illia Polosukhin. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL] (cit. alle pp. 15, 22).

[27] Yunchao Gong, Yangqing Jia, Thomas Leung, Alexander Toshev e Sergey Ioffe. *Deep Convolutional Ranking for Multilabel Image Annotation*. 2014. arXiv: 1312.4894 [cs.CV] (cit. a p. 16).

[28] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe e Laurens van der Maaten. *Exploring the Limits of Weakly Supervised Pretraining*. 2018. arXiv: 1805.00932 [cs.CV] (cit. a p. 16).

[29] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu e Kaiming He. «Aggregated Residual Transformations for Deep Neural Networks». In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5987–5995. DOI: 10.1109/CVPR.2017.634 (cit. a p. 22).

[30] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV] (cit. a p. 22).

[31] Hao Wang, Doyen Sahoo, Chenghao Liu, Ee-peng Lim e Steven C. H. Hoi. *Learning Cross-Modal Embeddings with Adversarial Networks for Cooking Recipes and Food Images*. 2019. arXiv: 1905.01273 [cs.CV] (cit. alle pp. 23, 26).

[32] Ting Chen, Simon Kornblith, Mohammad Norouzi e Geoffrey Hinton. *A Simple Framework for Contrastive Learning of Visual Representations*. 2020. arXiv: 2002.05709 [cs.LG] (cit. a p. 24).

[33] Jonathan C. Stroud, Zhichao Lu, Chen Sun, Jia Deng, Rahul Sukthankar, Cordelia Schmid e David A. Ross. *Learning Video Representations from Textual Web Supervision*. 2021. arXiv: 2007.14937 [cs.CV] (cit. a p. 24).

[34] Diederik P. Kingma e Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG] (cit. a p. 26).

[35] Jing-Jing Chen, Chong-Wah Ngo, Fu-Li Feng e Tat-Seng Chua. «Deep Understanding of Cooking Procedure for Cross-Modal Recipe Retrieval». In: MM '18. Seoul, Republic of Korea: Association for Computing Machinery, 2018, pp. 1020–1028. ISBN: 9781450356657. DOI: 10.1145/3240508.3240627. URL: https://doi.org/10.1145/3240508.3240627 (cit. a p. 26).

[36] Micael Carvalho, Rémi Cadène, David Picard, Laure Soulier, Nicolas Thome e Matthieu Cord. *Cross-Modal Retrieval in the Cooking Context: Learning Semantic Text-Image Embeddings*. 2018. arXiv: 1804.11146 [cs.CL] (cit. a p. 26).

[37] Bin Zhu, Chong-Wah Ngo, Jingjing Chen e Yanbin Hao. «R2GAN: Cross-Modal Recipe Retrieval With Generative Adversarial Network». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Giu. 2019 (cit. a p. 26).

[38] Han Fu, Rui Wu, Chenghao Liu e Jianling Sun. *MCEN: Bridging Cross-Modal Gap between Cooking Recipes and Dish Images with Latent Variable Model*. 2020. arXiv: 2004.01095 [cs.CV] (cit. a p. 26).

[39] Hao Wang, Doyen Sahoo, Chenghao Liu, Ke Shu, Palakorn Achananuparp, Ee-peng Lim e Steven C. H. Hoi. *Cross-Modal Food Retrieval: Learning a Joint Embedding of Food Images and Recipes with Semantic Consistency and Attention Mechanism*. 2021. arXiv: 2003.03955 [cs.CV] (cit. a p. 26).

[40] Mikhail Fain, Niall Twomey, Andrey Ponikar, Ryan Fox e Danushka Bollegala. *Dividing and Conquering Cross-Modal Recipe Retrieval: from Nearest Neighbours Baselines to SoTA*. 2021. arXiv: 1911.12763 [cs.CV] (cit. a p. 26).

[41] *GialloZafferano*. URL: `https://www.giallozafferano.it/` (cit. alle pp. 30, 31).

[42] *Fatto in casa da Benedetta*. URL: `https://www.fattoincasadabenedetta.it/` (cit. a p. 30).

[43] *Misya*. URL: `https://www.misya.info/` (cit. alle pp. 30, 31).

[44] *Cookaround*. URL: `https://www.cookaround.com/` (cit. a p. 31).

[45] *Requests python*. URL: `https://requests.readthedocs.io/en/latest/` (cit. a p. 31).

[46] *Bigrams nltk python*. URL: `https://tedboy.github.io/nlps/generated/generated/nltk.bigrams.html` (cit. a p. 35).

[47] *Nltk python*. URL: `https://tedboy.github.io/nlps/generated/nltk.html` (cit. a p. 35).

[48] *Skip-thoughts vectors*. URL: `https://github.com/sanyam5/skip-thoughts` (cit. a p. 35).

[49] *Tokenize nltk python*. URL: `https://www.nltk.org/api/nltk.tokenize.html` (cit. a p. 44).

[50] *Googletrans 3.0.0*. URL: `https://pypi.org/project/googletrans/` (cit. alle pp. 58, 79).

[51] Jianfa Chen, Yue Yin e Yifan Xu. *RecipeSnap – a lightweight image-to-recipe model*. 2022. arXiv: `2205.02141 [cs.CV]` (cit. a p. 106).