

Politecnico di Torino

Politecnico di Torino

Corso di Laurea Magistrale in Ingegneria Informatica

A.a. 2025/2026

Sessione di laurea Settembre 2026

Ricerca cross-modale di serie temporali satellitari mediante apprendimento contrastivo

Relatori:

Prof. Paolo Garza

Dott. Daniele Rege Cambrin

Candidato:

Enrico Balbo

Sommario

Il telerilevamento satellitare mette a disposizione archivi di immagini della Terra in continua crescita, ma interrogarli richiede ancora competenze tecniche specialistiche. Un'interfaccia in linguaggio naturale, capace di associare una descrizione testuale all'evoluzione osservata in una *serie temporale* di immagini della stessa area, renderebbe questi archivi accessibili a un pubblico molto più ampio. Questa tesi affronta il problema del *retrieval cross-modale* per serie temporali satellitari: dato l'embedding di una sequenza di immagini multitemporali, recuperare la descrizione corretta del cambiamento (image-to-text) e viceversa (text-to-image). Il caso bi-temporale — una coppia prima/dopo, tipico della *change detection* — è trattato come caso particolare ($T = 2$) della formulazione generale, ed è quello su cui si concentrano i benchmark annotati oggi disponibili.

Il sistema proposto adotta un paradigma di apprendimento contrastivo in stile CLIP. Un encoder visivo pre-addestrato per il telerilevamento (CROMA) estrae le rappresentazioni di ciascun frame; un *modulo di fusione temporale* dedicato aggrega la sequenza — di lunghezza variabile — in un unico embedding; un text encoder CLIP codifica le descrizioni. Di tale modulo si formulano e confrontano due varianti: una che codifica il cambiamento per costruzione, tramite differenze tra frame consecutivi, e una basata su attenzione temporale. Le due modalità vengono allineate in uno spazio condiviso tramite una loss NT-Xent con temperatura apprendibile. L'addestramento combina un dataset auto-supervisionato su larga scala (SSL4EO-S12) con dataset di change captioning annotati da esseri umani (MOSAIC-SEN2-CC e SECOND-CC), mentre la valutazione si estende anche a un benchmark completamente out-of-domain (LEVIR-CC).

I contributi principali riguardano la modellazione — due varianti di modulo per l'aggregazione temporale, messe a confronto, e uno schema di generazione delle didascalie *frame-by-frame* — e un'analisi approfondita dello spazio di rappresentazione appreso, che identifica i fattori che abilitano e limitano il retrieval cross-modale in questo dominio. La valutazione, condotta con un protocollo rigoroso e comparabile tra benchmark di dimensioni diverse, quantifica le prestazioni del sistema sui compiti di retrieval image-to-text e text-to-image e sulla classificazione zero-shot dell'uso del suolo.

Ringraziamenti

Desidero ringraziare il relatore, prof. Paolo Garza, per avermi seguito lungo tutto il percorso di questo lavoro, e il correlatore, dott. Daniele Rege Cambrin, per la disponibilità costante e per i numerosi suggerimenti tecnici che hanno indirizzato le scelte progettuali e le analisi presentate in questa tesi. Un ringraziamento va anche a chi, nel gruppo di ricerca, ha contribuito con confronti e osservazioni nelle diverse fasi del lavoro.

Ringrazio il Politecnico di Torino per le risorse di calcolo messe a disposizione: senza l'accesso all'infrastruttura HPC di Ateneo gran parte degli esperimenti riportati in questo elaborato non sarebbe stata realizzabile.

Un pensiero particolare va alla mia famiglia, per il sostegno costante e per avermi permesso di portare a termine questo percorso di studi con serenità.

Infine ringrazio gli amici e i compagni di corso che hanno condiviso con me questi anni, per il confronto, l'aiuto reciproco e i momenti passati insieme.

Indice

Elenco delle tabelle VI

Elenco delle figure VII

Acronimi VIII

1 Introduzione 1
- 1.1 Telerilevamento e Change Detection . . . 1
- 1.2 Retrieval Cross-Modale per Serie Temporali Satellitari . . . 2
- 1.3 Sfide Aperte e Motivazione . . . 3
- 1.4 Contributi . . . 4
- 1.5 Struttura della Tesi . . . 5

2 Lavori Correlati 6
- 2.1 Fondamenti di Deep Learning per la Visione . . . 6
  - 2.1.1 Dal Percettrone alle Reti Neurali Profonde . . . 6
  - 2.1.2 Reti Neurali Convoluzionali . . . 7
  - 2.1.3 Reti Ricorrenti e i Loro Limiti . . . 8
  - 2.1.4 Il Transformer e il Meccanismo di Attenzione . . . 8
  - 2.1.5 Vision Transformer . . . 9
  - 2.1.6 Applicazioni di Visione Artificiale: Classificazione, Segmentazione, Rilevamento . . . 10
  - 2.1.7 Modelli Linguistici di Grandi Dimensioni . . . 10
- 2.2 Apprendimento di Rappresentazioni Contrastivo . . . 11
  - 2.2.1 Self-Supervised e Contrastive Learning . . . 11
  - 2.2.2 CLIP e il Pre-addestramento Vision-Language . . . 12
  - 2.2.3 La Loss InfoNCE / NT-Xent . . . 13
  - 2.2.4 Geometria dello Spazio di Rappresentazione Contrastivo . . . 14
- 2.3 Foundation Model per l'Osservazione della Terra . . . 15
  - 2.3.1 Immagini Satellitari e Firme Spettrali . . . 15
  - 2.3.2 Pre-addestramento Self-Supervised: SSL4EO-S12 . . . 16

2.3.3 CROMA 16
2.3.4 Aggregazione Temporale di Serie di Immagini Satellitari . . 18
2.3.5 Varianti CLIP per il Telerilevamento 19
2.3.6 CLOSP e GeoCLOSP: un Ponte Testuale tra Ottico e SAR . 20
2.4 Change Captioning nel Telerilevamento 21
2.4.1 Definizione del Task e Metriche di Valutazione 21
2.4.2 Architetture per la Generazione di Didascalie di Cambiamento 22
2.4.3 MOSAIC-SEN2-CC 22
2.4.4 SECOND-CC 23
2.4.5 LEVIR-CC 23
2.5 Protocolli di Valutazione per il Retrieval Cross-Modale 24

3 Metodologia 26
3.1 Formulazione del Problema 26
3.2 Architettura del Modello 27
3.2.1 Encoder Visivo: CROMA 28
3.2.2 Il Modulo di Fusione Temporale 29
3.2.3 Variante A: il Delta Encoder 29
3.2.4 Variante B: il Temporal Attention Encoder 31
3.2.5 Text Encoder: CLIP 32
3.2.6 Projection Head e Spazio di Embedding Condiviso 33
3.2.7 Loss Contrastiva con Temperatura Apprendibile 33
3.3 Dataset 33
3.3.1 SSL4EO-S12 34
3.3.2 MOSAIC-SEN2-CC 35
3.3.3 SECOND-CC 35
3.3.4 LEVIR-CC 35
3.4 Pipeline di Generazione delle Didascalie 35
3.4.1 Prompt Engineering Data-to-Text 36
3.4.2 V1: Didascalie Primo-vs-Ultimo Frame 36
3.4.3 V2: Didascalie Frame-by-Frame 36
3.5 Procedura di Addestramento 37
3.5.1 Strategia di Combinazione dei Dataset 38
3.5.2 Sblocco Selettivo dei Layer 38
3.5.3 Temporal Dropout 38
3.6 Protocollo di Valutazione 39
3.6.1 Metriche di Retrieval 39
3.6.2 Classificazione Zero-Shot dell'Uso del Suolo 39
3.6.3 Comparabilità tra Pool di Dimensione Diversa 39

4 Esperimenti 41
4.1 Setup Sperimentale 41
4.2 Storia dello Sviluppo 42
4.3 Risultati del Modello Migliore 43
4.4 Ablazione sullo Stile delle Didascalie: V1 vs V2 44
4.5 Il Ruolo di SSL4EO-S12 come Regolarizzatore 46
4.6 Esperimenti di Scaling 47
4.7 Analisi dello Spazio Latente 48
4.8 Analisi Geometrica dello Spazio Condiviso 52
4.9 Discussione 55

5 Conclusioni 58
5.1 Riepilogo dei Contributi 58
5.2 Limiti 59
5.3 Lavori Futuri 60

Bibliografia 61

Elenco delle tabelle

4.1 Retrieval SSL4EO-S12 del modello migliore 43
4.2 Retrieval T2I e I2T del modello migliore sui benchmark downstream 44
4.3 Confronto tra didascalie V1 45
4.4 Ablazione: modello addestrato senza SSL4EO-S12 nel training . . . 46
4.5 Il trade-off in-domain/downstream negli esperimenti di scaling . . . 48
4.6 Collasso dimensionale ai quattro stadi del sistema 53
4.7 Effetto della capacità della testa di proiezione testuale 53

Elenco delle figure

2.1 Schema del pre-addestramento contrastivo di CLIP 13
2.2 Esempi di patch da SSL4EO-S12 v1.1 17
2.3 Architettura di CROMA 18
2.4 Architettura di CLOSP/GeoCLOSP 20
2.5 Esempi di coppie bi-temporali e relative didascalie dal 23
2.6 Esempio dal dataset SECOND-CC 24
2.7 Esempi di coppie bi-temporali dal dataset LEVIR-CC 24

3.1 Architettura complessiva del modello proposto 28
3.2 Dettaglio del Delta Encoder 30
3.3 Dettaglio del Temporal Attention Encoder 32

4.1 Proiezione UMAP dello spazio di embedding condiviso 49
4.2 Similarità coseno tra i centroidi dei sotto-domini 51
4.3 Sintesi del percorso sperimentale 55

Acronimi

CLIP Contrastive Language-Image Pre-training
ViT Vision Transformer
MAE Masked Autoencoder
SSL Self-Supervised Learning
RS Remote Sensing
EO Earth Observation
NDVI Normalized Difference Vegetation Index
LULC Land Use / Land Cover
I2T Image-to-Text
T2I Text-to-Image
I2I Image-to-Image (retrieval)
mAP mean Average Precision
LLM Large Language Model
VLM Vision-Language Model
GAP Global Average Pooling
MLP Multi-Layer Perceptron
IA Intelligenza Artificiale
ML Machine Learning
DL Deep Learning
CV Computer Vision
NLP Natural Language Processing
CNN Convolutional Neural Network
RNN Recurrent Neural Network
LSTM Long Short-Term Memory
SAR Synthetic Aperture Radar
CD Change Detection
RSCC Remote Sensing Change Captioning
NT-Xent Normalized Temperature-scaled Cross-Entropy
InfoNCE Information Noise-Contrastive Estimation
UMAP Uniform Manifold Approximation and Projection
ReLU Rectified Linear Unit
SGD Stochastic Gradient Descent
ESA European Space Agency
T2RSIR Text-to-Remote-Sensing-Image Retrieval
MSI MultiSpectral Imagery

Capitolo 1

Introduzione

In questo capitolo si introduce il problema di ricerca affrontato dalla tesi, le motivazioni che ne sono all'origine e la struttura complessiva del lavoro. Si parte dal contesto applicativo — il telerilevamento e la Change Detection (CD) — per poi formalizzare il task di retrieval cross-modale, discutere le sfide aperte e riassumere i contributi.

1.1 Telerilevamento e Change Detection

Il telerilevamento, o Remote Sensing (RS), ovvero l'acquisizione di informazioni sulla superficie terrestre a distanza tramite sensori montati su piattaforme satellitari o aeree, è oggi una delle principali fonti di dati per il monitoraggio ambientale su scala planetaria. Programmi di osservazione della Terra, ossia Earth Observation (EO), come Copernicus dell'Agenzia Spaziale Europea (ESA), con la costellazione Sentinel, forniscono ogni giorno terabyte di immagini multispettrali con copertura globale e rivisitazione di pochi giorni [1]. Questa mole di dati, in costante crescita, apre possibilità senza precedenti per lo studio dei fenomeni che modificano il territorio, ma pone al tempo stesso il problema di come renderla effettivamente interrogabile e utile.

Tra i task fondamentali del telerilevamento, la Change Detection (CD) — l'identificazione delle differenze nello stato della superficie terrestre osservando la stessa area in istanti temporali diversi [2] — riveste un ruolo centrale. Individuare e caratterizzare i cambiamenti è essenziale in un ampio ventaglio di applicazioni: monitoraggio della deforestazione, gestione delle emergenze a seguito di eventi catastrofici (alluvioni, incendi, terremoti), pianificazione urbana, analisi dell'espansione agricola e studio degli effetti del cambiamento climatico. Nella sua formulazione classica, la CD confronta due sole osservazioni della stessa area — una coppia

bi-temporale "prima/dopo" — ed è stata affrontata come un problema di classificazione a livello di pixel, assegnando a ciascun pixel un'etichetta di "cambiato" o "non cambiato", o una specifica classe di transizione, e producendo come output mappe di cambiamento di natura puramente visiva. Molti fenomeni di interesse, tuttavia, non si esauriscono in un singolo salto tra due istanti ma si sviluppano come processi lungo il tempo — la crescita urbana in un decennio, il progredire della deforestazione, il ciclo stagionale della vegetazione. Osservare l'area come *serie temporale* di più acquisizioni, di cui la coppia bi-temporale è il caso minimo ($T = 2$), consente di coglierne l'evoluzione e non solo lo stato finale.

Una mappa binaria di cambiamento, tuttavia, non risponde alla domanda che tipicamente interessa un utente umano: *cosa* è cambiato, e in che modo. Per colmare questo divario è emerso più di recente il task di Remote Sensing Change Captioning (RSCC), in cui a un insieme di immagini della stessa area acquisite in tempi diversi viene associata una descrizione in linguaggio naturale del cambiamento osservato [2]. Il RSCC sposta la CD da un problema di segmentazione a un problema al confine tra visione e linguaggio, avvicinando l'output del sistema al modo in cui gli esseri umani comunicano l'informazione geospaziale.

Nella stessa direzione si colloca anche un lavoro recente, CLOSP (*Contrastive Language Optical SAR Pretraining*) [3], che allinea testo e immagini ottiche e radar ad apertura sintetica (SAR) in uno spazio condiviso, pur limitandosi a osservazioni singole e statiche. Tra gli sviluppi futuri i suoi autori indicano l'estensione a una dimensione temporale e l'uso di Large Language Model (LLM) per annotazioni più descrittive (Sezione 2.3.6): due direzioni che anche questa tesi affronta. L'impianto qui proposto è però costruito in modo autonomo e con scelte diverse — un encoder visivo già pre-addestrato in modo contrastivo tra ottico e radar (CROMA, *Contrastive Radar-Optical Masked Autoencoders*), benchmark di change captioning consolidati in letteratura (RSCC) e una generazione delle didascalie assistita da LLM. Con quel lavoro si condivide dunque la direzione generale, non l'architettura né i dati.

1.2 Retrieval Cross-Modale per Serie Temporali Satellitari

Il change captioning, nella sua formulazione generativa, produce una frase a partire dalle immagini di ingresso. Questa tesi adotta invece una prospettiva *complementare e discriminativa*: anziché generare una descrizione, si intende *allineare* immagini e testi in uno spazio di rappresentazione condiviso, in modo che a ciascuna sequenza di immagini corrisponda, per vicinanza geometrica, la sua descrizione corretta. È il paradigma introdotto su larga scala da Contrastive Language-Image

Pre-training (CLIP) [4] per le immagini naturali, qui trasferito al dominio del telerilevamento multitemporale.

Concretamente, il sistema apprende due funzioni di codifica: un encoder visivo che mappa una *serie temporale* di immagini della stessa area — da un minimo di due osservazioni fino a più acquisizioni — in un vettore $\mathbf{z}_{\text{img}} \in \mathbb{R}^d$, e un text encoder che mappa una descrizione in un vettore $\mathbf{z}_{\text{txt}} \in \mathbb{R}^d$ nello stesso spazio. L'obiettivo dell'addestramento è massimizzare la similarità tra gli embedding di coppie immagine–testo corrispondenti e minimizzarla tra coppie non corrispondenti. La coppia bi-temporale "prima/dopo" rimane il caso particolare $T = 2$ di questa formulazione. Una volta appreso lo spazio condiviso, il sistema abilita due modalità di *retrieval cross-modale*:

- **Image-to-Text (I2T)**: data una serie temporale di immagini, si recuperano le descrizioni testuali più affini, ordinandole per similarità nello spazio condiviso;
- **Text-to-Image (T2I)**: data una descrizione in linguaggio naturale (ad esempio "area edificata comparsa al posto di vegetazione"), si recuperano le serie temporali che meglio la rappresentano.

Il retrieval T2I costituisce, di fatto, un'interfaccia in linguaggio naturale per interrogare un archivio satellitare: un utente può descrivere a parole il tipo di cambiamento che cerca e ottenere gli scenari corrispondenti, senza dover formulare query tecniche o conoscere la struttura dei dati. È questa la principale motivazione applicativa del lavoro.

1.3 Sfide Aperte e Motivazione

Trasferire il paradigma CLIP dalle immagini naturali al telerilevamento multitemporale presenta diverse difficoltà specifiche. La prima è la natura *temporale* del dato: mentre CLIP allinea singole immagini a singole didascalie, qui l'unità di significato è l'*evoluzione* lungo una sequenza di osservazioni della stessa area — di lunghezza variabile, dalla coppia bi-temporale alla serie di più frame — che l'encoder visivo deve saper rappresentare in modo compatto e discriminativo. La seconda è la natura del segnale: le immagini Sentinel-2 sono multispettrali (12 bande) e a risoluzione moderata (10 m), molto diverse dalle immagini RGB naturali su cui i modelli vision-language sono tipicamente pre-addestrati.

La sfida più insidiosa è però il *domain gap* tra le sorgenti di dati impiegate. Costruire un sistema robusto richiede grandi quantità di coppie immagine–testo, ma i dataset disponibili differiscono profondamente tra loro: per contenuto visivo (scene, sensori, risoluzioni diverse), per stile e vocabolario delle descrizioni, e per la distribuzione degli eventi rappresentati. Un modello addestrato su un insieme eterogeneo di sorgenti deve generalizzare a eventi visivi e testuali potenzialmente molto

diversi da quelli visti in addestramento — pena il collasso delle rappresentazioni su pattern superficiali del training set.

A queste difficoltà se ne aggiunge una più sottile, legata alla natura del segnale. In un dominio a risoluzione moderata, dove i cambiamenti temporali sono spesso tenui, non è scontato che il modello apprenda una rappresentazione realmente ricca e discriminativa: il rischio è che lo spazio degli embedding *collassi* su poche dimensioni, sufficienti a distinguere le classi di uso del suolo ma non le singole scene, limitando così il retrieval a grana fine. Analizzare le proprietà dello spazio di rappresentazione appreso — e non fermarsi alle sole metriche aggregate — diventa quindi parte integrante del lavoro, tanto quanto la progettazione dell'architettura.

1.4 Contributi

I principali contributi di questa tesi sono i seguenti:

- **Un'architettura per il retrieval cross-modale su serie temporali** che integra un encoder visivo pre-addestrato per il telerilevamento (CROMA) con un *modulo di fusione temporale* dedicato — capace di trattare sequenze di lunghezza variabile, di cui la coppia bi-temporale è il caso minimo — e un text encoder CLIP, allineati tramite loss contrastiva Normalized Temperature-scaled Cross-Entropy (NT-Xent) con temperatura apprendibile. Di tale modulo vengono formulate e confrontate due varianti: il *Delta Encoder*, che codifica il cambiamento per costruzione calcolando differenze tra frame consecutivi, e un *Temporal Attention Encoder* che lo apprende tramite attenzione su frame e transizioni.

- **Una strategia di generazione automatica delle didascalie** in stile *data-to-text*, che converte metriche quantitative estratte dalle immagini (indici di vegetazione e distribuzioni di uso del suolo) in descrizioni in linguaggio naturale tramite un LLM, incluso uno schema *frame-by-frame* che modella l'evoluzione cronologica dell'intera serie temporale.

- **Uno studio dell'uso di dataset eterogenei** nell'addestramento contrastivo, che mostra il ruolo del dataset auto-supervisionato su larga scala come componente necessaria alla capacità di retrieval, e il ruolo critico del bilanciamento relativo delle sorgenti nel batch.

- **Un protocollo di valutazione rigoroso e un'analisi dello spazio di rappresentazione**: una metodologia di misura comparabile tra benchmark di dimensioni diverse e un'analisi delle proprietà dello spazio latente appreso — in particolare della ricchezza effettiva delle rappresentazioni — che individua i fattori che abilitano e limitano il retrieval cross-modale in questo dominio.

1.5 Struttura della Tesi

Il resto della tesi è organizzato come segue:

- Il **Capitolo 2** (*Lavori Correlati*) presenta i fondamenti teorici e lo stato dell'arte rilevanti per il lavoro: apprendimento profondo e Transformer per la visione, apprendimento contrastivo e modelli vision-language, foundation model per l'osservazione della Terra, il task di change captioning con i relativi dataset e i protocolli di valutazione del retrieval cross-modale.
- Il **Capitolo 3** (*Metodologia*) descrive in dettaglio l'architettura proposta, i dataset impiegati, la pipeline di generazione delle didascalie, la procedura di addestramento e il protocollo di valutazione.
- Il **Capitolo 4** (*Esperimenti*) riporta il setup sperimentale e i risultati sui compiti di retrieval e classificazione zero-shot, con la storia dello sviluppo del sistema, gli studi di ablazione e l'analisi dello spazio di rappresentazione appreso.
- Il **Capitolo 5** (*Conclusioni*) riassume i risultati e i contributi principali, ne discute i limiti e delinea le direzioni di lavoro futuro.

Capitolo 2

Lavori Correlati

Questo capitolo presenta i fondamenti teorici e lo stato dell'arte su cui si basa il lavoro di tesi. L'esposizione ha un obiettivo dichiaratamente didattico: rendere accessibili anche a un lettore non specialista di apprendimento profondo i concetti necessari a comprendere le scelte progettuali discusse nei capitoli successivi, senza per questo rinunciare al rigore tecnico. Si parte dai fondamenti dell'apprendimento profondo per la visione (Sezione 2.1) — dalle reti neurali alle architetture convoluzionali e ricorrenti, fino al Transformer e al Vision Transformer — per poi introdurre l'apprendimento contrastivo di rappresentazioni e i modelli vision-language (Sezione 2.2). Si passa quindi ai foundation model per l'osservazione della Terra, con particolare attenzione ai modelli e ai dataset impiegati in questa tesi (Sezione 2.3). Infine si descrive il task di change captioning nel telerilevamento, le architetture proposte in letteratura e i relativi benchmark (Sezione 2.4), e si richiamano i protocolli standard di valutazione del retrieval cross-modale (Sezione 2.5).

2.1 Fondamenti di Deep Learning per la Visione

2.1.1 Dal Percettrone alle Reti Neurali Profonde

Il Machine Learning (ML) è il ramo dell'intelligenza artificiale (Intelligenza Artificiale (IA)) che si occupa di costruire sistemi capaci di migliorare le proprie prestazioni su un compito sfruttando l'esperienza, ossia i dati, anziché regole scritte esplicitamente da un programmatore. Il Deep Learning (DL) è la branca del ML basata su reti neurali artificiali con più livelli (o *layer*) di trasformazioni non lineari [5]. La caratteristica distintiva del DL rispetto agli approcci di ML tradizionali è la capacità di apprendere direttamente dai dati grezzi rappresentazioni gerarchiche delle *feature* (le caratteristiche rilevanti del dato), senza bisogno di una loro definizione manuale — un paradigma noto come *representation learning*. Questa

proprietà è particolarmente preziosa nella Computer Vision (CV) (l'elaborazione automatica di immagini), dove le feature utili a un compito (i bordi di un oggetto, la texture di una superficie, la forma complessiva di un edificio) sono spesso troppo complesse e variabili da specificare a priori con regole scritte a mano.

Una rete neurale è composta da unità elementari, i *neuroni*, organizzate in layer successivi: ogni neurone calcola una combinazione lineare pesata dei propri ingressi e ne passa il risultato a una *funzione di attivazione* non lineare. La non linearità è essenziale — una pila di sole trasformazioni lineari collasserebbe in un'unica trasformazione lineare, incapace di modellare le relazioni complesse dei dati reali. L'addestramento consiste nel cercare i pesi che minimizzano una *funzione di costo* (o *loss*), misura dello scarto tra predizioni e valori desiderati: la minimizzazione avviene per discesa del gradiente (Stochastic Gradient Descent (SGD) e varianti moderne come Adam [6]), e il gradiente rispetto a ciascuno dei milioni di parametri è calcolato dall'algoritmo di *backpropagation*, che propaga l'errore all'indietro attraverso i layer applicando la regola della catena.

Il rischio principale è l'*overfitting*: la rete memorizza le particolarità del set di addestramento anziché la regolarità sottostante, con prestazioni ottime sui dati visti e scadenti su dati nuovi. Per questo i dati vengono suddivisi in *training*, *validation* e *test* — rispettivamente per addestrare, decidere quando fermarsi (*early stopping*) e misurare le prestazioni finali su dati mai osservati — e vi si affiancano tecniche di regolarizzazione come il *dropout* [7], che disattiva casualmente una frazione dei neuroni a ogni iterazione. Del *learning rate* si controlla infine l'andamento tramite uno *scheduler* che lo riduce gradualmente, preceduto da una fase di *warmup* in cui cresce da zero, così da evitare aggiornamenti bruschi quando i pesi sono ancora prossimi all'inizializzazione casuale.

2.1.2 Reti Neurali Convoluzionali

Una rete completamente connessa non è efficiente per le immagini: una sola immagine RGB 224×224 corrisponde a oltre 150.000 valori di ingresso, e un singolo layer denso con 1000 neuroni richiederebbe più di 150 milioni di parametri.

Le Convolutional Neural Network (CNN) risolvono il problema sfruttando tre principi ispirati alla struttura spaziale delle immagini [5]: la *connettività locale* (ogni neurone osserva una piccola regione dell'input), la *condivisione dei pesi* (lo stesso *filtro* è applicato a tutte le posizioni) e la conseguente *invarianza per traslazione*. Un layer convoluzionale fa scorrere i filtri appresi sull'immagine, calcolando a ogni posizione un prodotto scalare con la porzione sottostante; il risultato è una *feature map* che indica dove un certo pattern è presente. Impilando più layer, le feature dei primi catturano pattern semplici e locali (bordi, texture), quelle dei successivi — operando su un *campo recettivo* via via più ampio — pattern sempre più astratti

(parti di oggetti, scene). Layer di *pooling* riducono periodicamente la risoluzione spaziale, aumentando il campo recettivo e la robustezza a piccole traslazioni.

L'affermazione delle CNN risale a LeNet-5 [8], ma il salto di scala si ebbe nel 2012 con AlexNet [9], che vinse la ImageNet Large Scale Visual Recognition Challenge dimostrando la superiorità delle CNN profonde su dataset grandi. Vi seguirono VGG [10] e soprattutto ResNet [11], che con le *connessioni residue* — la somma dell'ingresso di un blocco alla sua uscita, che permette al gradiente di "saltare" i layer intermedi — rese addestrabili reti di centinaia di layer. Le CNN restano tuttora tra le famiglie di architetture più diffuse nel telerilevamento.

2.1.3 Reti Ricorrenti e i Loro Limiti

Prima del Transformer (Sezione 2.1.4), il trattamento di dati sequenziali era dominio delle Recurrent Neural Network (RNN): reti in cui l'uscita di un passo temporale rientra in ingresso al passo successivo, mantenendo uno *stato nascosto* che funge da memoria di quanto osservato fino a quel momento. Questa struttura presenta due limiti. Il primo è computazionale: poiché lo stato al passo t dipende da quello al passo $t-1$, il calcolo non è parallelizzabile lungo l'asse temporale. Il secondo è numerico: la backpropagation attraverso molti passi soffre di scomparsa (o esplosione) del gradiente, ostacolando l'apprendimento di dipendenze tra elementi distanti. Varianti come le Long Short-Term Memory (LSTM) [12] introducono meccanismi a *gate* che regolano quali informazioni mantenere o scartare, alleviando il secondo problema ma non il collo di bottiglia sequenziale — limiti che motivano direttamente l'architettura discussa nella sezione seguente.

2.1.4 Il Transformer e il Meccanismo di Attenzione

Introdotto da Vaswani et al. nel 2017 [13], il *Transformer* è un'architettura basata interamente sul meccanismo di *attenzione* (*attention*), che elimina la ricorrenza e con essa il collo di bottiglia sequenziale delle RNN, permettendo di elaborare tutti gli elementi di una sequenza in parallelo. L'idea centrale è la *self-attention*: ogni elemento di una sequenza (una parola in una frase, o — come si vedrà — una patch di un'immagine) aggiorna la propria rappresentazione come media pesata delle rappresentazioni di *tutti* gli altri elementi, indipendentemente dalla loro distanza. Nella frase "L'animale non ha attraversato la strada perché era troppo stanco", per capire a cosa si riferisca "era" occorre collegarlo a "l'animale", per quanto distante.

Concretamente, ogni elemento viene proiettato in tre vettori — *query*, *key* e *value* (Q, K, V) — tramite trasformazioni lineari apprese, secondo l'analogia di una ricerca in un catalogo: la query è ciò che si cerca, le key l'indice su cui ci si confronta, i value il contenuto restituito. L'attenzione a prodotto scalare scalato calcola, per ogni query, la similarità con tutte le key, la trasforma in pesi tramite

softmax e restituisce la media pesata dei value:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{2.1}$$

dove d_k è la dimensione dei vettori key, e il fattore $1/\sqrt{d_k}$ serve a mantenere numericamente stabile la softmax al crescere di d_k. Impilando più "teste" di attenzione in parallelo (*multi-head attention*) — ciascuna libera di specializzarsi nel catturare un diverso tipo di relazione, sintattica, semantica o posizionale — e più blocchi in sequenza, il Transformer modella dipendenze a lungo raggio con un grado di parallelismo molto superiore a quello delle reti ricorrenti. Poiché il meccanismo di attenzione, di per sé, non ha alcuna nozione dell'ordine degli elementi (scambiando l'ordine della sequenza in ingresso, il risultato per ciascun elemento non cambierebbe), è necessario aggiungere esplicitamente un *positional encoding* che codifichi la posizione di ciascun elemento nella sequenza.

Questa efficienza e scalabilità hanno permesso l'addestramento di modelli di dimensioni senza precedenti, alla base sia dei grandi modelli linguistici — nella variante che utilizza solo la parte encoder del Transformer originale, come BERT [14] — sia, come si vedrà, dei moderni modelli di visione.

2.1.5 Vision Transformer

Il Vision Transformer (ViT) [15] ha esteso l'architettura Transformer alla visione artificiale in forma pura, senza alcun layer convoluzionale. L'immagine viene suddivisa in una griglia regolare di *patch* di dimensione fissa (ad esempio 16×16 pixel: un'immagine di 224×224 pixel viene così suddivisa in $14 \times 14 = 196$ patch); ciascuna patch viene appiattita in un vettore e proiettata linearmente in un embedding, ottenendo una sequenza di "token visivi" del tutto analoga, dal punto di vista del Transformer, a una sequenza di parole. A questa sequenza vengono aggiunti un positional encoding, che reintroduce l'informazione sulla posizione spaziale altrimenti persa nella suddivisione in patch, ed eventualmente un token speciale di classe; la sequenza viene poi elaborata da uno stack di blocchi Transformer identico, nella sua struttura, a quello impiegato per il testo.

Rispetto alle CNN, il ViT non incorpora a priori alcun *inductive bias* spaziale: mentre una CNN assume implicitamente, per costruzione, che pixel vicini siano più correlati di pixel lontani (un'assunzione quasi sempre corretta per le immagini naturali), il ViT deve apprendere da zero, a partire dai dati, se e come la vicinanza spaziale sia rilevante. Questa assunzione implicita delle CNN è un'utile scorciatoia quando i dati di addestramento sono scarsi, ma diventa un limite quando i dati sono abbondanti abbastanza da permettere al modello di scoprire da sé le regolarità spaziali rilevanti, incluse quelle che un inductive bias troppo rigido potrebbe escludere a priori. Per questo motivo, un ViT addestrato da zero su dataset di

dimensioni moderate tende a essere meno efficiente in termini di dati rispetto a una CNN comparabile; tuttavia, quando pre-addestrato su quantità di immagini sufficientemente grandi, il ViT eguaglia o supera le migliori CNN, beneficiando maggiormente della scala. Questa proprietà lo rende l'architettura di elezione per i *foundation model* vision-language e per il telerilevamento discussi nel seguito; l'encoder visivo impiegato in questa tesi (Sezione 2.3.3) è appunto basato su ViT.

2.1.6 Applicazioni di Visione Artificiale: Classificazione, Segmentazione, Rilevamento

CNN e ViT sono "motori" di estrazione di feature riutilizzabili per compiti di visione diversi, che ricorrono anche nel telerilevamento e nei dataset impiegati in questa tesi. La *classificazione* assegna un'unica etichetta all'intera immagine; la *segmentazione semantica* ne assegna una a *ogni pixel*, producendo la mappa densa che caratterizza le mappe di Land Use / Land Cover (LULC) (Sezione 2.3.1) e quelle di cambiamento semantico di SECOND-CC (Sezione 2.4.4); il *rilevamento di oggetti* individua singole istanze. La *change detection* è una segmentazione applicata non a una singola immagine ma a una coppia o sequenza di osservazioni della stessa area, in cui l'etichetta di ogni pixel non descrive il tipo di superficie ma se e come sia cambiata. Il sistema proposto si colloca in parte fuori da questa tassonomia: anziché un'etichetta discreta o una mappa densa produce un embedding continuo dell'intera sequenza, da confrontare con embedding testuali in uno spazio condiviso (Sezione 2.2).

2.1.7 Modelli Linguistici di Grandi Dimensioni

Se il Vision Transformer applica il meccanismo di attenzione alle immagini, la sua controparte linguistica — i LLM — lo applica al testo, ed è una componente diretta di questa tesi, che se ne serve per generare le didascalie di addestramento. Un LLM è tipicamente un Transformer *decoder-only* addestrato in modo auto-supervisionato su enormi corpus testuali con un unico obiettivo: predire la parola (più precisamente, il *token*) successivo data la sequenza precedente. Ripetuto su scala di miliardi di parametri e migliaia di miliardi di token, questo semplice compito di completamento produce modelli capaci di generare testo fluente e di svolgere compiti linguistici mai visti esplicitamente in addestramento. Una fase successiva di *instruction tuning* — l'affinamento su esempi di istruzioni e risposte desiderate — li rende inoltre capaci di seguire richieste formulate in linguaggio naturale, come "descrivi il cambiamento osservato tra queste due date".

In questa tesi un LLM pre-addestrato e instruction-tuned (Qwen2.5-14B) è impiegato in un compito di tipo *data-to-text*: convertire dati strutturati estratti dalle immagini — distribuzioni di uso del suolo e indici di vegetazione — in

una descrizione in linguaggio naturale del cambiamento. Questa scelta permette di generare automaticamente didascalie per il grande volume di immagini auto-supervisionate (SSL4EO-S12), che non dispongono di annotazioni umane. È però anche il punto più delicato dell'intera pipeline: la ricchezza semantica di una didascalia così generata è limitata da quella dei dati di partenza — un modello linguistico può riformulare in modo fluente solo l'informazione che riceve in ingresso, non crearne di nuova. La generazione delle didascalie diventa quindi una scelta progettuale centrale, le cui implicazioni sulla qualità delle rappresentazioni apprese sono analizzate nel Capitolo 4.

2.2 Apprendimento di Rappresentazioni Contrastivo

2.2.1 Self-Supervised e Contrastive Learning

L'apprendimento supervisionato classico richiede, per ogni esempio di addestramento, un'etichetta fornita da un annotatore umano — un processo costoso e spesso il principale collo di bottiglia nella costruzione di sistemi di deep learning su larga scala. L'Self-Supervised Learning (SSL) propone un'alternativa: apprendere rappresentazioni utili da dati non etichettati, derivando automaticamente un segnale di supervisione dalla struttura intrinseca dei dati stessi, senza alcun intervento umano. Un primo filone di metodi si basa su *pretext task*: compiti artificiali, risolvibili automaticamente a partire dai dati grezzi — predire la rotazione applicata a un'immagine, ricomporne le tessere mescolate, colorarne una versione in scala di grigi — la cui soluzione richiede comunque di apprendere rappresentazioni visive utili, e la cui etichetta è nota per costruzione.

All'interno del più ampio paradigma SSL, l'apprendimento *contrastivo* è oggi tra gli approcci di maggior successo. L'idea è organizzare lo spazio delle rappresentazioni in modo che campioni "simili" (coppie *positive*) siano vicini e campioni "diversi" (coppie *negative*) siano lontani, secondo una qualche misura di similarità (tipicamente il coseno tra vettori normalizzati). In SimCLR [16], le coppie positive sono ottenute applicando due augmentation casuali indipendenti (ritaglio, variazione di colore, sfocatura gaussiana) alla stessa immagine, generando due "viste" distinte che il modello deve imparare a riconoscere come rappresentanti lo stesso contenuto semantico; tutte le altre immagini del batch fungono da negativi. Un limite pratico di questo approccio è che la qualità del segnale contrastivo dipende dal numero di negativi disponibili, il che richiede batch di addestramento molto grandi (con conseguente necessità di hardware costoso). MoCo [17] vi risponde mantenendo una coda di rappresentazioni negative calcolate nei batch precedenti tramite un secondo encoder aggiornato come media mobile dei pesi del primo, ottenendo molti negativi

senza batch proibitivi. Il successo di questi metodi ha mostrato che rappresentazioni apprese senza etichette possono rivaleggiare con quelle supervisionate, aprendo la strada al loro impiego come inizializzazione per i task a valle (*downstream*). Una volta pre-addestrato, un encoder può essere trasferito al task a valle in due modi principali: nel *linear probing*, i pesi pre-addestrati restano completamente congelati e si addestra soltanto un semplice classificatore lineare sopra le feature estratte, una procedura economica che misura direttamente la qualità delle rappresentazioni apprese; nel *fine-tuning*, invece, tutti o parte dei pesi pre-addestrati vengono ulteriormente aggiornati sul nuovo task, tipicamente ottenendo prestazioni migliori ma al costo di un rischio di *catastrophic forgetting* se l'adattamento è troppo aggressivo o il nuovo dataset troppo piccolo — un compromesso discusso concretamente in Sezione 2.3.5 e verificato sperimentalmente nel Capitolo 4 per gli encoder impiegati in questa tesi.

2.2.2 CLIP e il Pre-addestramento Vision-Language

CLIP (*Contrastive Language-Image Pre-training*) [4] porta l'apprendimento contrastivo nel contesto *cross-modale*, allineando immagini e testo in un unico spazio di embedding anziché limitarsi a confrontare diverse viste della stessa immagine. Il modello è costituito da due encoder separati — uno visivo e uno testuale — addestrati congiuntamente su circa 400 milioni di coppie immagine–didascalia raccolte dal web (il dataset WIT). L'obiettivo è massimizzare la similarità coseno tra l'embedding di un'immagine e quello della sua didascalia corretta, e minimizzarla rispetto a tutte le altre didascalie presenti nello stesso batch (Figura 2.1).

Il risultato è uno spazio semantico condiviso che consente la classificazione *zero-shot*, ossia senza alcun ulteriore addestramento su etichette specifiche del nuovo compito: per classificare un'immagine tra un insieme di classi candidate, è sufficiente costruire per ciascuna classe una frase-prompt (ad esempio `"a photo of a {classe}"`), calcolarne l'embedding testuale, confrontarlo con l'embedding dell'immagine, e scegliere la classe la cui frase risulta più simile. Modelli successivi come ALIGN [18] hanno confermato che la scala dei dati — più che raffinatezze architetturali — è il fattore determinante di questo approccio. Il paradigma CLIP si è inoltre dimostrato utile ben oltre la classificazione: spazi di embedding vision-language di questo tipo guidano oggi anche sistemi di generazione di immagini da testo e, come nel caso di questa tesi, sistemi di retrieval cross-modale. Non è tuttavia privo di limiti noti: CLIP fatica, ad esempio, su compiti che richiedono di contare oggetti o di ragionare su relazioni spaziali precise tra di essi, limitazioni che motivano in parte l'interesse per varianti specializzate per dominio (Sezione 2.3.5). CLIP resta comunque il paradigma di riferimento diretto di questa tesi: l'obiettivo è replicarne la struttura a due encoder e la loss contrastiva nel dominio del telerilevamento

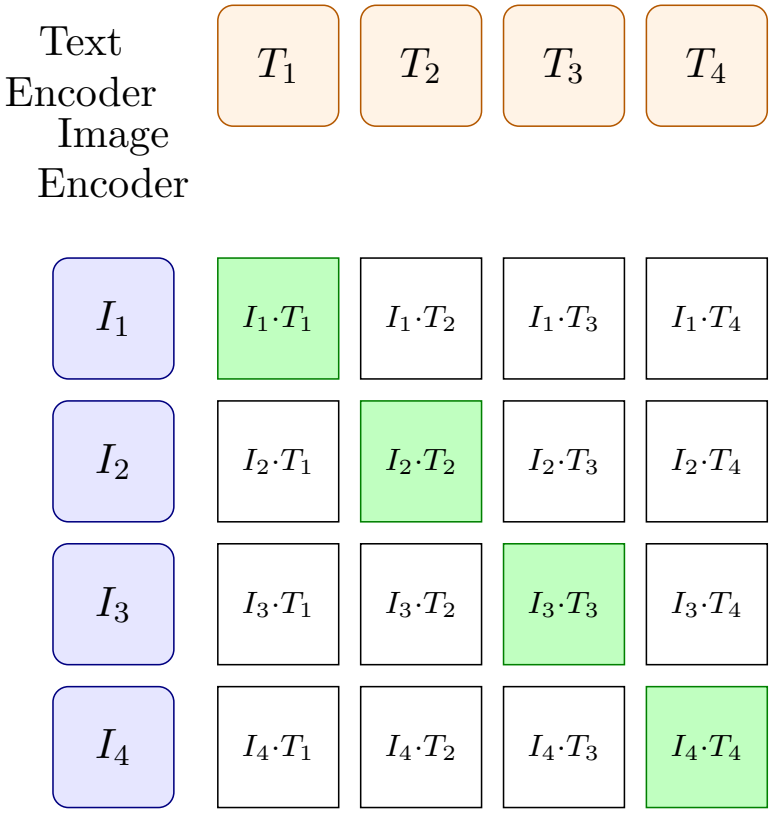


Figura 2.1: Schema del pre-addestramento contrastivo di CLIP. Gli embedding di immagini e testi vengono allineati in uno spazio condiviso: nella matrice di similarità del batch, gli elementi diagonali (coppie corrette, in verde) vengono massimizzati e quelli fuori diagonale (negativi) minimizzati. Rielaborato da [4].

multitemporale, sostituendo l'encoder visivo con uno specializzato e introducendo un modulo di aggregazione temporale.

2.2.3 La Loss InfoNCE / NT-Xent

La funzione di costo alla base di questi metodi è la Information Noise-Contrastive Estimation (InfoNCE) [19], nota nella sua variante normalizzata come Normalized Temperature-scaled Cross-Entropy (NT-Xent) [16]. Dati due insiemi di embedding normalizzati ℓ_2 — nel caso cross-modale, $\{\mathbf{z}_i^{\text{img}}\}$ e $\{\mathbf{z}_i^{\text{txt}}\}$ — con la coppia (i, i) come positiva, la loss per la direzione immagine→testo su un batch di dimensione N è

$$\mathcal{L}_{\text{i2t}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\Big(\text{sim}(\mathbf{z}_i^{\text{img}}, \mathbf{z}_i^{\text{txt}})/\tau\Big)}{\sum_{j=1}^{N} \exp\Big(\text{sim}(\mathbf{z}_i^{\text{img}}, \mathbf{z}_j^{\text{txt}})/\tau\Big)}, \tag{2.2}$$

dove $\text{sim}(\cdot,\cdot)$ è la similarità coseno e τ è un parametro di *temperatura* che regola la nitidezza della distribuzione. Intuitivamente, il termine dentro il logaritmo è una softmax sulle similarità tra l'immagine i e tutte le didascalie del batch: se, ad esempio, un'immagine ha similarità $[0.9, 0.2, 0.1, 0.3]$ con quattro didascalie candidate — di cui la prima è quella corretta — la softmax trasforma questi valori in una distribuzione di probabilità concentrata sulla prima candidata, e la loss spinge il modello a rendere questa concentrazione sempre più marcata nel corso dell'addestramento. La temperatura τ controlla quanto la distribuzione risultante sia "appuntita": valori piccoli di τ amplificano le differenze di similarità, rendendo la loss più sensibile ai negativi più difficili (quelli con similarità più alta pur essendo scorretti). La loss complessiva è la media simmetrica di $\mathcal{L}_{\text{i2t}}$ e dell'analoga $\mathcal{L}_{\text{t2i}}$, calcolata scambiando il ruolo di immagini e testi.

Un aspetto elegante di questa formulazione, rispetto a loss contrastive precedenti come la *triplet loss* (che richiede di selezionare esplicitamente, spesso con euristiche di *hard negative mining*, un singolo negativo per ogni esempio positivo — l'approccio adottato ad esempio da VSE++ [20]), è che tutte le altre didascalie del batch fungono automaticamente da negativi, senza alcuna selezione esplicita: batch più grandi offrono così, gratuitamente, un segnale contrastivo più ricco e diversificato. Un secondo aspetto rilevante per il seguito è che la temperatura τ, anziché essere fissata a priori come iperparametro, viene in CLIP resa un *parametro apprendibile* — scelta adottata anche in questa tesi — lasciando che sia l'addestramento stesso a calibrare il grado di confidenza ottimale delle predizioni.

2.2.4 Geometria dello Spazio di Rappresentazione Contrastivo

Le metriche di retrieval dicono *quanto bene* un modello contrastivo recupera, ma non *come* sia fatto lo spazio che ha appreso. Due fenomeni documentati in letteratura descrivono altrettante patologie geometriche di questi spazi, ed entrambi si riveleranno necessari per interpretare i risultati sperimentali di questa tesi (Capitolo 4).

Il primo è il *modality gap* [21]. In un modello a due encoder come CLIP ci si aspetterebbe che, a convergenza, le rappresentazioni delle due modalità occupino la stessa regione dello spazio condiviso; in pratica immagini e testi si collocano invece in due coni nettamente distinti, separati da una distanza sistematica. Gli autori mostrano che il fenomeno nasce dalla combinazione di due fattori: l'*inizializzazione* — una rete profonda mappa i propri input in un cono ristretto dello spazio, e due encoder distinti producono quindi due coni già separati prima ancora dell'addestramento — e l'*ottimizzazione contrastiva*, che tende a preservare tale separazione in misura regolata dalla temperatura della loss. Il punto rilevante è che il gap *non* implica un allineamento fallito: le coppie corrette possono

restare molto più vicine tra loro di quelle casuali pur appartenendo a coni distinti. Ne consegue che una proiezione bidimensionale in cui le due modalità appaiono separate va interpretata con cautela, e che la qualità dell'allineamento va misurata con strumenti che siano insensibili a questa traslazione rigida.

Il secondo è il *collasso dimensionale* [22]. Mentre il collasso completo — tutti gli embedding che degenerano in un unico punto — è scongiurato dai campioni negativi, gli autori mostrano che anche l'apprendimento contrastivo soffre di una forma più sottile di degenerazione: gli embedding finiscono per occupare un sottospazio di dimensione molto inferiore a quella nominale dello spazio, lasciando inutilizzata gran parte della capacità rappresentativa disponibile. Il lavoro identifica nella dinamica di ottimizzazione — e in particolare nel ruolo del *projector* addestrabile — uno dei meccanismi che vi conducono. Per quantificare il fenomeno si ricorre all'*effective rank*, che misura in quante direzioni la varianza degli embedding è effettivamente distribuita: la formulazione adottata in questa tesi è quella di RankMe [23], dove il rango effettivo è definito come l'esponenziale dell'entropia dello spettro normalizzato degli autovalori della matrice di covarianza, e viene proposto come predittore della qualità delle rappresentazioni apprese. Tra le contromisure esplicitamente pensate per contrastare questa degenerazione si colloca VICReg [24], che affianca alla loss principale due termini di regolarizzazione: uno che impone una varianza minima a ciascuna dimensione e uno che ne decorrela le componenti, spingendo così lo spazio a distribuirsi su un numero maggiore di direzioni.

2.3 Foundation Model per l'Osservazione della Terra

2.3.1 Immagini Satellitari e Firme Spettrali

L'occhio umano percepisce soltanto tre bande dello spettro elettromagnetico (rosso, verde, blu); i sensori ottici della missione Sentinel-2 [1] ne acquisiscono invece 13, incluse regioni non visibili come il vicino e il medio infrarosso, con risoluzioni spaziali fra i 10 e i 60 metri per pixel. Questa ricchezza spettrale non è un dettaglio tecnico secondario: è precisamente ciò che rende osservabili in modo affidabile fenomeni altrimenti difficili da distinguere nel solo visibile. Ogni tipo di superficie (vegetazione, acqua, suolo nudo, aree edificate) possiede infatti una caratteristica *firma spettrale* — un pattern distintivo di riflettanza al variare della lunghezza d'onda — che consente di distinguerla combinando opportunamente le bande disponibili. Su questo principio si fondano gli *indici spettrali*: il più noto è il Normalized Difference Vegetation Index (NDVI) [25], definito come $(\text{NIR} - \text{Rosso})/(\text{NIR} + \text{Rosso})$, che sfrutta la forte riflettanza della vegetazione sana nel vicino infrarosso (NIR) a fronte del suo assorbimento nel rosso. Valori

prossimi a 0.6–0.8 indicano vegetazione densa, intorno a 0.1–0.2 suolo nudo o aree urbane, negativi corpi idrici. Un secondo descrittore quantitativo ampiamente usato è la classificazione in classi di LULC (uso e copertura del suolo — ad esempio bosco, coltura, area edificata, corpo idrico), tipicamente ottenuta da prodotti derivati come ESRI Land Cover o CORINE Land Cover. Nel presente lavoro, l'NDVI e la distribuzione delle classi di LULC vengono utilizzati come descrittori quantitativi del cambiamento, poi tradotti in didascalie testuali in linguaggio naturale (Sezione 2.4 e Capitolo 3).

2.3.2 Pre-addestramento Self-Supervised: SSL4EO-S12

L'assenza di grandi quantità di dati etichettati è un limite ricorrente nel telerilevamento: a differenza delle immagini naturali, per cui il web offre miliardi di esempi con didascalie scritte da utenti comuni, l'annotazione di immagini satellitari richiede quasi sempre competenze specialistiche (fotointerpretazione, conoscenza del territorio) difficili da reperire su larga scala. Questo rende l'SSL particolarmente attraente per l'osservazione della Terra. SSL4EO-S12 [26] è un dataset su larga scala pensato appositamente per il pre-addestramento auto-supervisionato: raccoglie, per decine di migliaia di località distribuite globalmente, quattro acquisizioni stagionali per ciascuna modalità — radar Sentinel-1 e ottiche Sentinel-2 — acquisite tra il 2019 e il 2021 con copertura nuvolosa inferiore al 10%, senza alcuna etichetta semantica (Figura 2.2). La sua natura multi-temporale e multi-stagionale lo rende una sorgente naturale per apprendere rappresentazioni del *cambiamento*, imponendo al tempo stesso di distinguere la variabilità stagionale dai veri segnali di trasformazione del territorio. In questa tesi SSL4EO-S12 è impiegato come principale sorgente di addestramento contrastivo, con didascalie generate automaticamente a partire da metriche estratte dalle immagini (si veda il Capitolo 3); come si mostrerà nel Capitolo 4, esso svolge un ruolo cruciale nella capacità di retrieval del sistema, pur non disponendo di alcuna annotazione redatta da esseri umani.

2.3.3 CROMA

CROMA (*Contrastive Radar-Optical Masked Autoencoders*) [27] è un foundation model per il telerilevamento che combina due obiettivi auto-supervisionati complementari. Il primo è il *masked autoencoding*, sulla scia del Masked Autoencoder (MAE) [28]: durante il pre-addestramento, ampie porzioni dell'immagine di ingresso vengono nascoste (mascherate), e il modello deve ricostruirle a partire dal contesto visibile circostante — un po' come risolvere un puzzle a cui manchino molti pezzi, sfruttando gli indizi offerti dai pezzi rimanenti. Per riuscire in questo compito in modo consistente, il modello è costretto a sviluppare una comprensione genuina della struttura della scena, e non una semplice memorizzazione dei pixel.

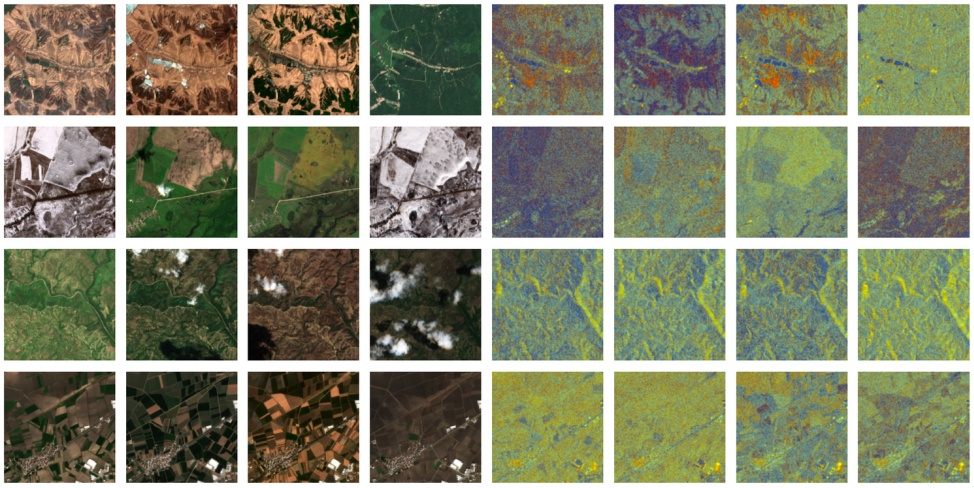


Figura 2.2: Esempi di patch da SSL4EO-S12 v1.1: quattro località (righe), ciascuna con quattro acquisizioni stagionali (colonne) nelle modalità Sentinel-2 ottico (a sinistra) e Sentinel-1 radar in pseudo-colore VV–VH (a destra). Adattata da [26].

Il secondo obiettivo è *contrastivo*, e allinea le rappresentazioni delle modalità radar (Sentinel-1) e ottica (Sentinel-2) della stessa area. Questa combinazione è particolarmente sensata nel telerilevamento: il segnale radar SAR attraversa le nubi ed è disponibile indipendentemente dall'illuminazione solare, ma è intrinsecamente più difficile da interpretare (a causa del rumore di *speckle* e di una fisica di formazione dell'immagine molto diversa da quella ottica); il segnale ottico è invece più intuitivo e semanticamente ricco, ma frequentemente oscurato dalla copertura nuvolosa. Allineando le due modalità in uno spazio di rappresentazione condiviso, il modello può trasferire struttura semantica dalla modalità più facile da interpretare (l'ottico) a quella più ostica (il radar). Architetturalmente, CROMA impiega tre encoder ViT: un encoder unimodale per il radar, uno unimodale per l'ottico, e un encoder multimodale con cross-attention che fonde le due modalità (Figura 2.3). Il risultato sono rappresentazioni robuste e trasferibili, che raggiungono lo stato dell'arte su numerosi benchmark di classificazione e segmentazione nel telerilevamento. In questo lavoro CROMA è adottato come *encoder visivo* pre-addestrato: i suoi pesi vengono in gran parte congelati e solo un numero minimo di layer viene reso addestrabile in fase di fine-tuning, per adattarlo al task cross-modale senza distruggere le rappresentazioni apprese durante il pre-addestramento (Capitolo 3).

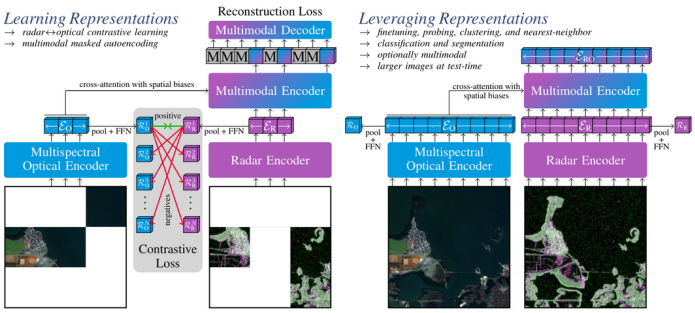


Figura 2.3: Architettura di CROMA. A sinistra, l'apprendimento delle rappresentazioni: tre encoder ViT (radar unimodale, ottico unimodale, e un encoder multimodale con cross-attention) sono addestrati con un obiettivo combinato di ricostruzione mascherata (MAE) e allineamento contrastivo radar–ottico. A destra, l'impiego delle rappresentazioni apprese nei task downstream. Adattata da [27].

2.3.4 Aggregazione Temporale di Serie di Immagini Satellitari

I foundation model discussi finora, CROMA compreso, operano su una *singola* acquisizione: producono un embedding per immagine. Un sistema che debba descrivere l'*evoluzione* di un'area richiede però un ulteriore stadio, che comprima la sequenza di embedding per-frame in un unico vettore — quello che nel seguito chiameremo *modulo di fusione temporale*. Le strategie possibili si collocano lungo uno spettro di complessità crescente.

All'estremo più semplice si trovano le aggregazioni statistiche prive di parametri, come la media o il massimo lungo l'asse temporale. Sono robuste e non aggiungono capacità da addestrare, ma sono anche *invarianti all'ordine* dei frame: due sequenze con gli stessi contenuti in ordine invertito — ad esempio un disboscamento e un rimboschimento — producono lo stesso embedding, il che le rende inadatte a descrivere un cambiamento orientato nel tempo. All'estremo opposto si collocano le reti ricorrenti e i Transformer applicati all'asse temporale, capaci di modellare dipendenze arbitrarie ma con un costo in parametri e dati proporzionalmente maggiore.

In ambito di telerilevamento, una soluzione intermedia particolarmente efficace è il *Temporal Attention Encoder* e la sua variante alleggerita L-TAE [29, 30], sviluppati per la classificazione di serie temporali di immagini satellitari. L'idea è applicare un meccanismo di self-attention lungo l'asse temporale, con due accorgimenti che ne riducono drasticamente il costo: le *query* non sono derivate dai singoli istanti ma sono un insieme di vettori appresi (*master query*), uno per testa di attenzione, il che permette di produrre direttamente un unico vettore per sequenza anziché uno per timestep; e ciascuna testa opera su un sottoinsieme disgiunto dei canali

(*channel grouping*) anziché sull'intera rappresentazione. Il risultato è un modulo che apprende *quali* istanti della serie siano informativi, pesandoli di conseguenza, con un numero di parametri contenuto.

Va però osservato che questi moduli sono stati concepiti per la *classificazione* della serie — riconoscere, ad esempio, la coltura presente in un appezzamento — un compito in cui l'informazione utile risiede prevalentemente nel contenuto degli istanti osservati. Il compito affrontato in questa tesi è diverso: l'obiettivo non è descrivere *cosa* sia presente, ma *come* sia cambiato. Poiché l'attenzione produce una combinazione convessa dei vettori in ingresso, un modulo che operi sui soli frame tende a restituire una rappresentazione della scena "media", in cui il segnale di cambiamento è implicito anziché esplicito. Questa osservazione motiva le due varianti di modulo di fusione temporale messe a confronto nel Capitolo 3: una che calcola esplicitamente le differenze tra frame consecutivi, e una basata su attenzione in cui l'insieme dei token include, accanto ai frame, le transizioni tra essi.

2.3.5 Varianti CLIP per il Telerilevamento

Il successo di CLIP ha stimolato numerosi tentativi di adattarlo al dominio satellitare, motivati dal fatto che un CLIP generalista, addestrato su immagini del web, ha incontrato di rado riprese dall'alto acquisite da satellite, con distribuzioni di colore, scale degli oggetti e vocabolario descrittivo molto diversi da quelli delle fotografie naturali. RemoteCLIP [31] effettua un fine-tuning di CLIP su coppie immagine–testo del telerilevamento, mentre GeoRSCLIP [32] lo addestra su RS5M, un dataset su larga scala (dell'ordine dei cinque milioni di coppie, come suggerisce il nome) di immagini satellitari MultiSpectral Imagery (MSI) con descrizioni generate automaticamente. Entrambi i modelli vengono valutati dagli autori, tra l'altro, sul task di Text-to-Remote-Sensing-Image Retrieval (T2RSIR): dato un archivio di immagini satellitari statiche (non temporali) e una query testuale, recuperare le immagini più pertinenti — lo stesso paradigma di retrieval adottato in questa tesi, qui esteso dal caso di un'unica immagine statica a quello di una sequenza temporale (Sezione 1.2). Questi modelli offrono un allineamento vision-language già specializzato per il dominio e costituiscono un'alternativa naturale a CLIP generalista come punto di partenza. Tuttavia, la loro maggiore specializzazione può renderli più fragili in fase di ulteriore fine-tuning: come si documenterà nel Capitolo 4, sbloccare anche un solo layer di un modello già fortemente specializzato può innescare fenomeni di *catastrophic forgetting* (la perdita catastrofica delle capacità pre-addestrate durante l'adattamento a un nuovo compito), laddove un CLIP generalista assorbe l'adattamento di dominio in modo più stabile. La scelta dell'encoder testuale è quindi essa stessa oggetto di analisi sperimentale in questa tesi.

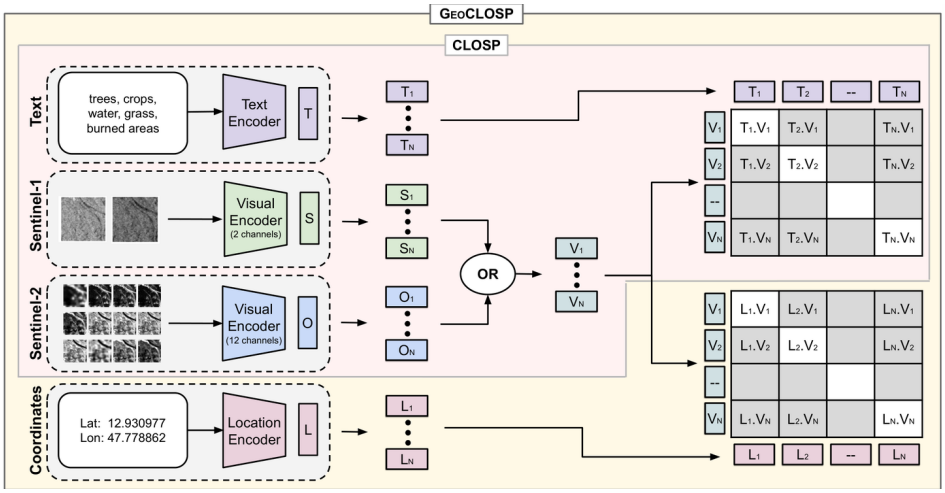


Figura 2.4: Architettura di CLOSP/GeoCLOSP: testo, immagini ottiche Sentinel-2 e immagini radar SAR Sentinel-1 vengono allineati in un unico spazio semantico condiviso tramite matrici di similarità calcolate per ciascuna coppia di modalità. Il testo funge da ponte tra ottico e SAR, che non vengono mai allineati direttamente. Adattata da [3].

2.3.6 CLOSP e GeoCLOSP: un Ponte Testuale tra Ottico e SAR

Un lavoro direttamente rilevante per questa tesi, di cui costituisce il punto di partenza concettuale, è CLOSP (*Contrastive Language Optical SAR Pretraining*) [3]. Gli autori introducono CrisisLandMark, un corpus su larga scala (oltre 647.000 coppie immagine–testo Sentinel-1 SAR e Sentinel-2 multispettrali) con annotazioni strutturate di uso/copertura del suolo (LULC, armonizzate da CORINE Land Cover e Dynamic World) ed eventi di crisi (alluvioni, incendi, terremoti), e propongono CLOSP, un'architettura contrastiva che allinea in un unico spazio semantico tre modalità: testo, immagini ottiche multispettrali e immagini radar SAR. Poiché coppie temporalmente allineate di SAR e ottico sono raramente disponibili (i tempi di rivisitazione di Sentinel-1 e Sentinel-2 non coincidono), CLOSP non impone alcun allineamento diretto tra le due modalità visive: il testo funge da *ponte* (Figura 2.4) — ciascuna modalità visiva viene allineata indipendentemente alla stessa annotazione testuale — permettendo comunque un trasferimento di conoscenza semantica dall'ottico, più facile da interpretare, al SAR, notoriamente più ostico. GeoCLOSP estende l'architettura con un encoder di localizzazione geografica (coordinate), rivelando un trade-off tra generalità semantica e specificità geografica: efficace per eventi legati a una posizione specifica (es. terremoti), meno per classi semantiche geograficamente diffuse (es. alberi, colture).

Nella sezione conclusiva del lavoro, gli autori identificano esplicitamente l'estensione a una *dimensione temporale* come passo naturale e prioritario: l'analisi di CLOSP è infatti limitata a immagini singole e statiche, mentre applicazioni critiche del telerilevamento — CD e analisi di tendenza — si basano sull'osservazione di un'area nel tempo. Gli autori suggeriscono che un'estensione temporale abiliterebbe query su processi dinamici (ad esempio, "deforestazione tra il 2020 e il 2024") e indicano l'uso di LLM multimodali per annotazioni e query più descrittive come ulteriore direzione promettente. Questa tesi raccoglie direttamente questi due suggerimenti — la dimensione temporale e la generazione di annotazioni assistita da LLM — come propria motivazione di partenza.

L'implementazione qui presentata diverge tuttavia da CLOSP su due assi. Sull'encoder visivo, anziché allineare da zero un backbone generalista, questa tesi adotta CROMA (Sezione 2.3.3), che dispone già di un allineamento SAR-ottico diretto ottenuto nel proprio pre-addestramento — quello stesso allineamento che CLOSP evita deliberatamente per la scarsità di coppie temporalmente coincidenti. Sui dati, anziché CrisisLandMark, si impiegano dataset annotati per il change captioning (RSCC, Sezione 2.4), scelta coerente con il focus sulla dimensione temporale anziché sull'unificazione multi-sensore, che resta fuori dallo scopo del lavoro. Restano parimenti fuori la pesatura adattiva semantico-geografica di GeoCLOSP e l'impiego di embedding pre-calcolati su scala globale.

2.4 Change Captioning nel Telerilevamento

2.4.1 Definizione del Task e Metriche di Valutazione

Il RSCC nasce all'incrocio di due campi consolidati ma storicamente separati: la *change detection*, che identifica e localizza a livello di pixel le regioni cambiate tra due o più osservazioni, e l'*image captioning*, che genera descrizioni testuali in linguaggio naturale a partire da un'immagine singola. Il RSCC unisce i due problemi: consiste nel generare una descrizione in linguaggio naturale del cambiamento osservato in un insieme di immagini della stessa area acquisite in tempi diversi [2] — nella pratica quasi sempre una coppia bi-temporale (caso $T = 2$, Sezione 1.1), poiché è questo il formato adottato dai benchmark annotati oggi disponibili. Ad esempio, data una coppia di immagini che ritraggono un campo agricolo vuoto seguito, mesi dopo, da un impianto di pannelli solari, una didascalia tipica potrebbe essere "un impianto fotovoltaico è stato costruito dove prima si trovava un terreno agricolo incolto" — una descrizione che richiede non solo di rilevare che qualcosa è cambiato, ma di comprenderne e verbalizzarne la natura semantica.

Nella formulazione generativa classica, un modello encoder-decoder (un encoder visivo che estrae feature dalle immagini di ingresso, un decoder testuale che le traduce in una frase) produce una descrizione, valutata con le metriche mutuate

dall'image captioning e dalla traduzione automatica: BLEU e METEOR, basate sulla sovrapposizione di n-grammi con i riferimenti umani, ROUGE, orientato al *recall*, CIDEr, che pesa gli n-grammi con uno schema TF-IDF per premiare i dettagli distintivi, e SPICE, che confronta grafi semantici della scena anziché sequenze superficiali di parole.

Questa tesi affronta il change captioning da una prospettiva *discriminativa* di retrieval anziché generativa; di conseguenza, il protocollo di valutazione non è quello generativo appena descritto ma quello del retrieval cross-modale (Sezione 2.5). Si tratta di una scelta legittima e diffusa in letteratura, ma che va dichiarata esplicitamente: i dataset descritti nel seguito sono nati come benchmark di captioning generativo, e il loro impiego come benchmark di retrieval in stile CLIP costituisce un adattamento del protocollo originale.

2.4.2 Architetture per la Generazione di Didascalie di Cambiamento

Sebbene questa tesi non adotti un approccio generativo, è utile inquadrare brevemente le architetture proposte per il change captioning classico, poiché diverse di esse hanno originato i dataset qui impiegati. RSICCformer [2], tra i primi modelli proposti, combina un estrattore di feature convoluzionale con un encoder Transformer a doppio ramo, che mette in relazione le due osservazioni temporali tramite cross-attention, e un decoder che genera la didascalia. I lavori successivi hanno raffinato i meccanismi di attenzione per enfatizzare le regioni di reale cambiamento e sopprimere le variazioni irrilevanti — differenze di illuminazione, rumore del sensore, disallineamenti geometrici — dato che non tutte le differenze pixel-a-pixel sono semanticamente rilevanti. Un filone più recente integra segnali ausiliari oltre alle immagini RGB: MModalCC [33] vi affianca mappe di segmentazione semantica del cambiamento, mostrando che segnali semantici strutturati migliorano sensibilmente le didascalie generate. Questo filone è complementare all'approccio discriminativo qui adottato, che rinuncia alla generazione esplicita di testo per concentrarsi sull'allineamento diretto tra rappresentazioni visive e testuali.

2.4.3 MOSAIC-SEN2-CC

MOSAIC-SEN2-CC [34] è un dataset di change captioning basato su immagini multispettrali Sentinel-2 e incentrato su eventi catastrofici reali (alluvioni, incendi e altri eventi che alterano il territorio). Il dataset comprende 5232 coppie bi-temporali distribuite su otto categorie di evento (inclusa una categoria *no-change*), per un totale di 26160 didascalie: ogni coppia è corredata da cinque didascalie redatte da annotatori umani, con uno stile sintetico ed event-driven (Figura 2.5). Oltre al

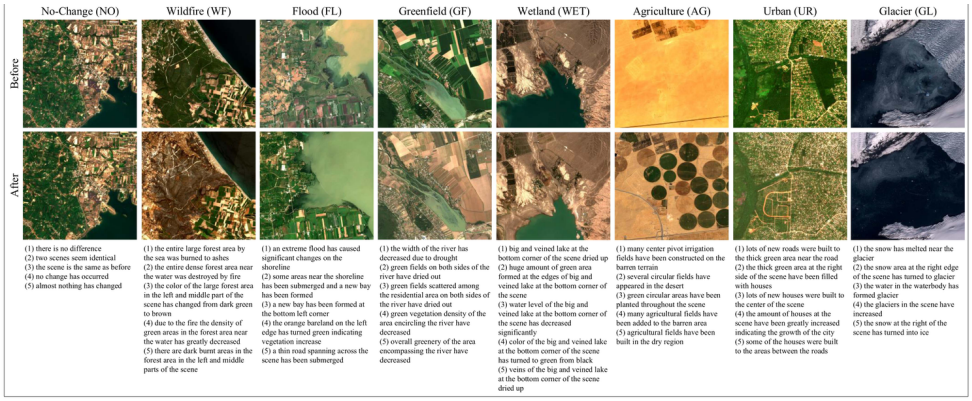


Figura 2.5: Esempi di coppie bi-temporali e relative didascalie dal dataset MOSAIC-SEN2-CC, per ciascuna delle otto categorie di evento (incendio, alluvione, zona umida, campo verde, agricoltura, area urbana, ghiacciaio, e *no-change*). Adattata da [34].

dataset, il lavoro originale propone un framework di adattamento delle didascalie verso stili testuali differenti. In questa tesi MOSAIC-SEN2-CC è impiegato sia come sorgente di addestramento sia come benchmark di test (Capitolo 4).

2.4.4 SECOND-CC

SECOND-CC [33] è un dataset di change captioning ad alta risoluzione (tra 0,5 e 3 metri per pixel), costruito a partire dal dataset di change detection SECOND e comprendente 6041 coppie di immagini e 30205 frasi complessive, anch'esso con cinque didascalie per coppia, accompagnato dal framework MModalCC descritto in Sezione 2.4.2 (Figura 2.6). Rispetto a MOSAIC, presenta un vocabolario più ripetitivo e una maggiore uniformità stilistica delle annotazioni, caratteristica che ne influenza il comportamento come benchmark di retrieval, come si osserverà nel Capitolo 4. Anche SECOND-CC viene utilizzato in questa tesi sia in addestramento sia in test.

2.4.5 LEVIR-CC

LEVIR-CC [2] è uno dei primi dataset su larga scala per il change captioning nel telerilevamento e tuttora uno dei più utilizzati come benchmark. Si basa su 10077 coppie di immagini RGB ad altissima risoluzione (circa 0,5 m per pixel) provenienti da Google Earth, per un totale di 50385 didascalie che descrivono il cambiamento a livello urbano (Figura 2.7). In questa tesi LEVIR-CC è utilizzato *esclusivamente in fase di test*, come benchmark *out-of-domain* puro: il modello

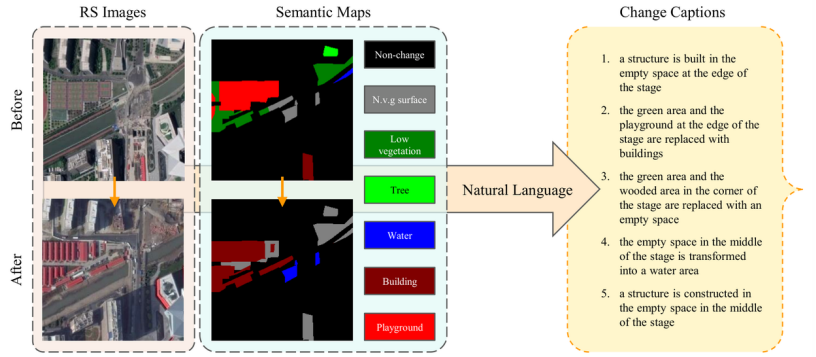


Figura 2.6: Esempio dal dataset SECOND-CC: coppia bi-temporale di immagini RS, relative mappe semantiche di cambiamento (per classe di uso del suolo) e didascalie in linguaggio naturale associate. Adattata da [33].

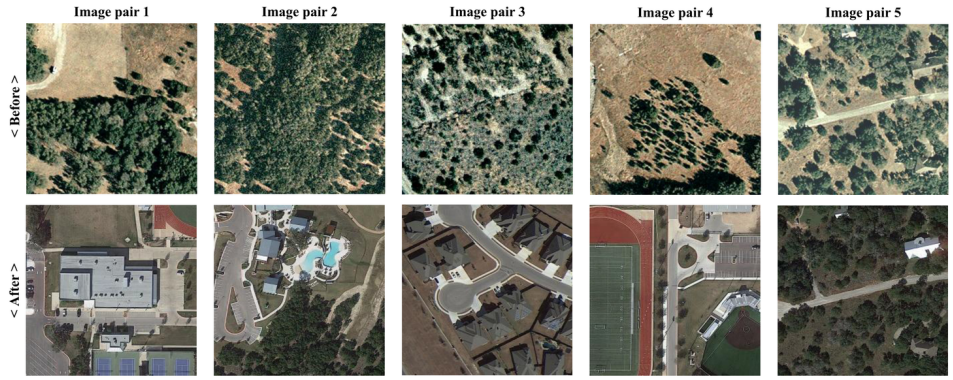


Figura 2.7: Esempi di coppie bi-temporali dal dataset LEVIR-CC: dalla trasformazione di aree boschive in complessi residenziali, commerciali e sportivi (coppie 1–4) a un caso di *no-change* (coppia 5). Adattata da [2].

non lo vede mai in addestramento. Ciò consente di misurare la generalizzazione in condizioni particolarmente sfidanti, dato il forte scarto — di risoluzione (0,5 m contro 10 m), di natura del segnale (RGB contro multispettrale) e di stile testuale — rispetto ai dati di training.

2.5 Protocolli di Valutazione per il Retrieval Cross-Modale

La valutazione del retrieval cross-modale segue protocolli consolidati nella letteratura vision-language [4, 20]. Data una query in una modalità (un'immagine

o un testo), il sistema ordina tutti i candidati dell'altra modalità per similarità decrescente nello spazio condiviso, e la qualità del ranking viene misurata con:

- **Recall@K**: la frazione di query per cui almeno un match corretto compare tra i primi K risultati del ranking. Valori tipici di K sono 1, 5 e 10.
- **Precision@K** e **mean Average Precision (mAP)**: rilevanti quando a una stessa immagine corrispondono più didascalie corrette (come nei dataset con cinque annotazioni per immagine discussi in Sezione 2.4), misurano rispettivamente la quota di risultati corretti tra i primi K e la precisione media integrata sull'intero ranking, premiando i sistemi che posizionano *tutti* i match corretti in alto, non solo il primo.
- **Mean Rank** e **Median Rank**: la posizione (media e mediana) del primo match corretto nel ranking completo. Sono indicatori particolarmente informativi perché, a differenza della Recall@K, tengono conto dell'intera distribuzione delle posizioni — non solo di quante query superano una soglia fissata — e permettono un confronto diretto con il valore atteso sotto ranking puramente casuale, pari a circa metà della dimensione del pool di candidati.

In questa tesi si riportano, accanto alla Recall@K, anche Precision@K, mAP e Mean/Median Rank, calcolati su un ranking esatto contro tutti i candidati del pool considerato. L'insieme di queste metriche, e il criterio di comparabilità tra pool di dimensione diversa adottato per confrontare in modo equo benchmark eterogenei, sono descritti in dettaglio nel Capitolo 3.

Capitolo 3

Metodologia

Questo capitolo descrive in dettaglio il sistema proposto. Si parte dalla formalizzazione del problema di retrieval cross-modale (Sezione 3.1), per poi presentare l'architettura del modello (Sezione 3.2) e i dataset impiegati (Sezione 3.3). Si descrive quindi la pipeline di generazione automatica delle didascalie (Sezione 3.4) e la procedura di addestramento (Sezione 3.5). Il capitolo si chiude con il protocollo di valutazione (Sezione 3.6), che definisce le metriche di retrieval e di classificazione zero-shot e i criteri di comparabilità tra benchmark di dimensioni diverse adottati per garantire misure affidabili.

3.1 Formulazione del Problema

Riprendendo la formulazione introdotta in Sezione 1.2, il sistema apprende due funzioni di codifica che mappano immagini e testo in uno spazio vettoriale condiviso. Sia $\mathcal{X} = (x_1, \ldots, x_T)$ una sequenza di $T \in \{2,3,4\}$ osservazioni Sentinel-2 della stessa area geografica, acquisite in istanti temporali distinti, dove ciascun frame $x_t \in \mathbb{R}^{12\times120\times120}$ ha 12 bande spettrali e risoluzione spaziale 120×120 pixel. Un encoder visivo f_{img} mappa l'intera sequenza in un embedding

$$\mathbf{z}_{\text{img}} = f_{\text{img}}(\mathcal{X}) \in \mathbb{R}^{512}, \tag{3.1}$$

mentre un encoder testuale f_{txt} mappa una didascalia y in

$$\mathbf{z}_{\text{txt}} = f_{\text{txt}}(y) \in \mathbb{R}^{512}. \tag{3.2}$$

Entrambi gli embedding vengono normalizzati ℓ_2 prima di essere confrontati, in modo che la similarità coseno tra due vettori coincida con il loro prodotto scalare. L'obiettivo di addestramento, dettagliato in Sezione 3.2.7, è apprendere f_{img} e f_{txt} in modo che, per ogni coppia corrispondente $(\mathcal{X}_i, y_i)$ del training set, $\mathbf{z}_{\text{img},i}$

e $\mathbf{z}_{\text{txt},i}$ risultino vicini nello spazio condiviso, mentre $\mathbf{z}_{\text{img},i}$ risulti distante dagli embedding testuali di didascalie non corrispondenti. Una volta appreso lo spazio condiviso, le due modalità di retrieval descritte in Sezione 1.2 — I2T e T2I — si riducono entrambe a un semplice ordinamento per similarità coseno decrescente tra un embedding query e un insieme di embedding candidati nell'altra modalità, secondo il protocollo dettagliato in Sezione 3.6.

La coppia bi-temporale "prima/dopo" (Sezione 1.1) è il caso particolare $T = 2$ di questa formulazione: l'encoder visivo deve saper trattare sequenze di lunghezza variabile senza modifiche architetturali, un requisito che guida direttamente la progettazione del modulo di fusione temporale descritto in Sezione 3.2.2.

3.2 Architettura del Modello

L'architettura proposta segue il paradigma a due encoder di CLIP (Sezione 2.2.2), sostituendo l'encoder visivo generalista con una pipeline specializzata per serie temporali di telerilevamento. Come mostrato in Figura 3.1, il ramo visivo codifica la sequenza di frame Sentinel-2 con CROMA e ne aggrega l'evoluzione temporale tramite il modulo di fusione temporale, mentre il ramo testuale codifica la didascalia con CLIP; entrambi i rami proiettano nello stesso spazio L_2-normalizzato, allineato dalla loss NT-Xent con temperatura apprendibile. La variante di modulo di fusione da impiegare è selezionabile in configurazione, senza modifiche al resto dell'architettura.

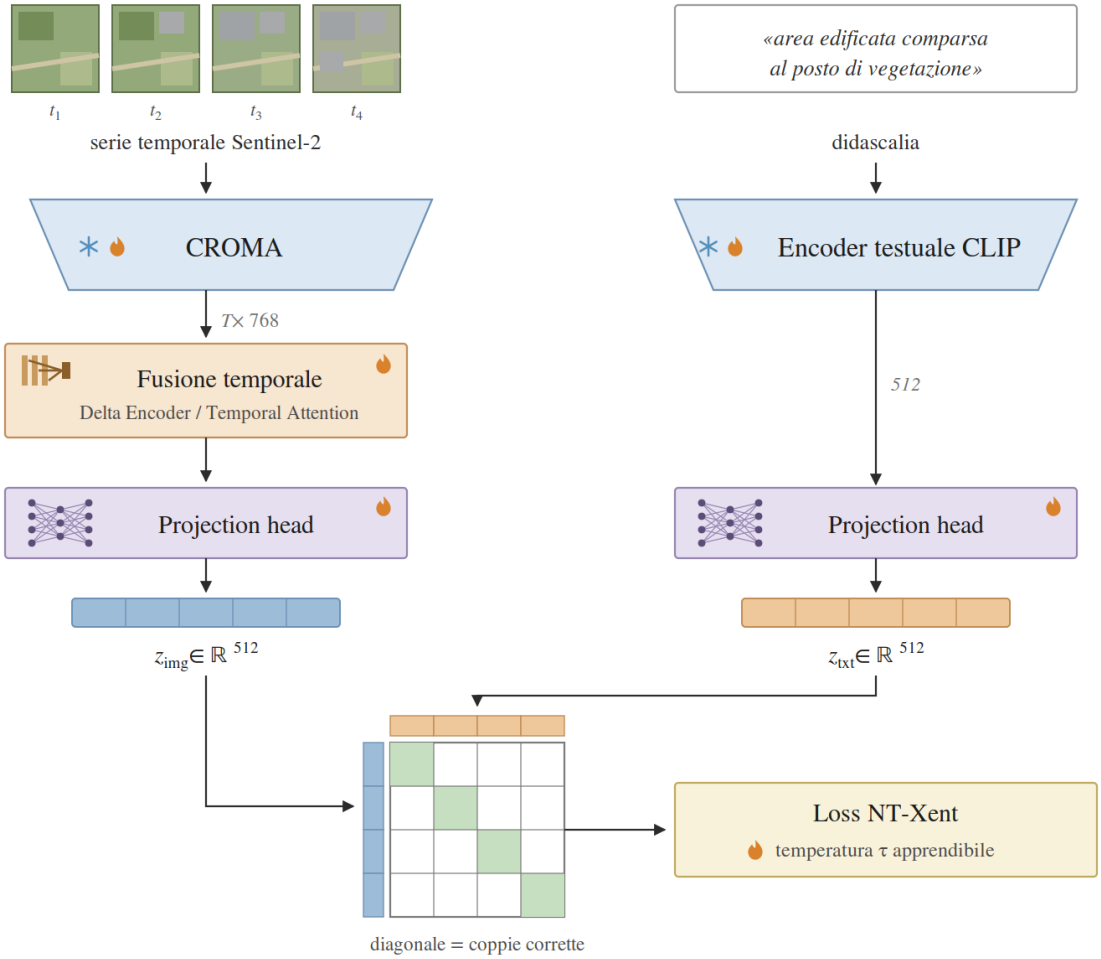


Figura 3.1: Architettura complessiva del modello proposto. Il ramo visivo codifica ogni frame della sequenza Sentinel-2 con CROMA e ne aggrega l'evoluzione temporale tramite il modulo di fusione temporale, di cui si confrontano due varianti (Sezione 3.2.2); il ramo testuale codifica la didascalia con CLIP. Entrambi i rami proiettano nello stesso spazio L_2-normalizzato a 512 dimensioni: nella matrice di similarità del batch, la diagonale (coppie corrette) viene massimizzata dalla loss NT-Xent con temperatura apprendibile (Sezione 3.2.7). Il fiocco di neve indica i pesi congelati, la fiamma quelli addestrabili.

3.2.1 Encoder Visivo: CROMA

CROMA (Sezione 2.3.3) processa ogni frame della sequenza indipendentemente attraverso un ViT-base (12 layer, dimensione 768, 16 teste di attenzione, patch 8×8), pre-addestrato con l'obiettivo combinato di masked autoencoding e allineamento contrastivo radar-ottico descritto in Sezione 2.3.3. Per ciascun frame si utilizza

il *global average pooling* sui token dell'encoder ottico, ottenendo una sequenza di embedding $(B, T, 768)$. Solo l'ultimo layer di CROMA viene reso addestrabile in fase di fine-tuning; il resto rimane congelato per preservare le rappresentazioni pre-addestrate. Questa scelta — motivata empiricamente e discussa in dettaglio in Sezione 3.5.2 — si è rivelata determinante: sbloccare l'intero encoder, o un encoder visivo già fortemente specializzato, espone il modello al rischio di *catastrophic forgetting* già introdotto in Sezione 2.3.5.

3.2.2 Il Modulo di Fusione Temporale

CROMA produce un embedding per ciascun frame della sequenza. Il *modulo di fusione temporale* è il componente che comprime questi T vettori in un unico embedding visivo $\mathbf{z}_{\text{img}} \in \mathbb{R}^{512}$, ed è il punto in cui l'architettura si specializza per il compito: è qui, e solo qui, che il *cambiamento* tra le osservazioni può essere reso esplicito. Formalmente, il modulo realizza una funzione

$$f_\theta : \mathbb{R}^{T\times 768} \times \{0,1\}^T \longrightarrow \mathbb{R}^{512}, \qquad \mathbf{z}_{\text{img}} = f_\theta\Big(\mathbf{e}_{1:T},\, \mathbf{m}\Big),$$

dove $\mathbf{e}_{1:T}$ sono gli embedding per-frame e $\mathbf{m}$ è una maschera booleana che marca i frame validi. Il condizionamento sulla maschera non è un dettaglio implementativo: permette di gestire sequenze di lunghezza variabile ($T \in \{2,3,4\}$, dove $T = 2$ è il caso bi-temporale dei benchmark di change captioning) ed è ciò che rende possibile il *Temporal Dropout* usato in addestramento (Sezione 3.5.3).

Il requisito che guida la progettazione di f_θ è la *sensibilità all'ordine*. Un'aggregazione statistica priva di parametri, come la media lungo l'asse temporale, è invariante a permutazioni: assegnerebbe lo stesso embedding a un disboscamento e al rimboschimento corrispondente, rendendo i due casi indistinguibili nel retrieval. Un primo modello basato proprio su un *global pooling* temporale si è infatti rivelato incapace di catturare il segnale di cambiamento (Capitolo 4), il che ha motivato lo sviluppo di moduli parametrici dedicati.

In questa tesi vengono formulate e messe a confronto due varianti di f_θ, che incarnano due strategie diverse per rendere il modello sensibile al cambiamento. La prima, il **Delta Encoder** (Sezione 3.2.3), lo codifica *per costruzione*, calcolando esplicitamente le differenze tra frame consecutivi. La seconda, il **Temporal Attention Encoder** (Sezione 3.2.4), lo *apprende*, lasciando a un meccanismo di attenzione il compito di pesare gli istanti e le transizioni rilevanti. Le due varianti sono intercambiabili a parità del resto dell'architettura, e il loro confronto sperimentale è riportato nel Capitolo 4.

3.2.3 Variante A: il Delta Encoder

Il Delta Encoder rende il cambiamento esplicito nella struttura stessa del calcolo. Per ogni coppia di frame consecutivi si calcolano la differenza e la somma degli

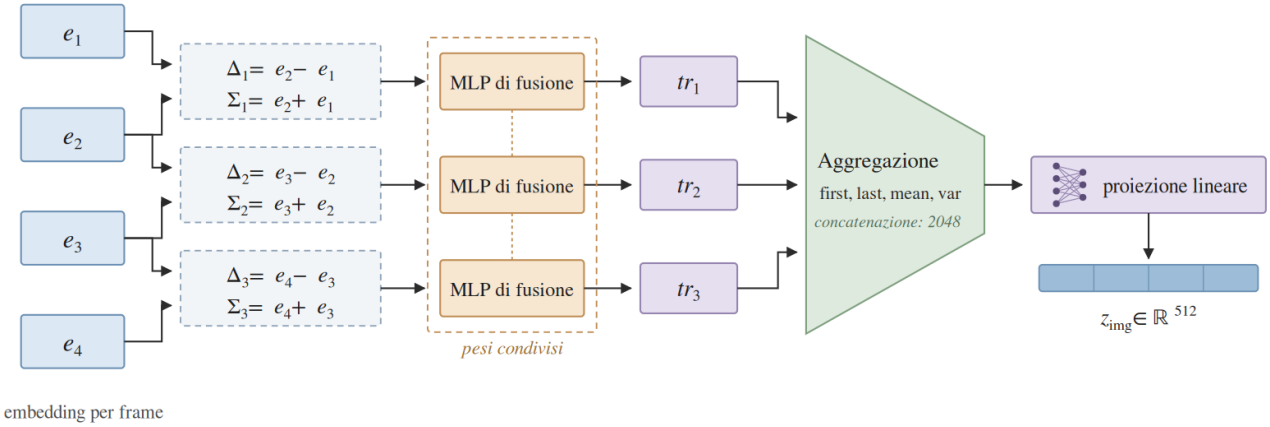


Figura 3.2: Dettaglio del Delta Encoder, illustrato per una sequenza di $T = 4$ frame (caso generale: $T \in \{2,3,4\}$). Per ogni coppia di frame consecutivi si calcolano le combinazioni differenza/somma degli embedding, trasformate da un MLP di fusione a pesi condivisi in un embedding di transizione tr_t. Le $T - 1$ transizioni vengono infine aggregate tramite concatenazione di primo elemento, ultimo elemento, media e varianza, e proiettate linearmente nell'embedding visivo finale $\mathbf{z}_{\mathrm{img}}$.

embedding, trasformate da un MLP di fusione condiviso (stessi pesi per ogni transizione) in un embedding di transizione. Le transizioni risultanti vengono infine aggregate concatenando primo elemento, ultimo elemento, media e varianza, e proiettate linearmente nell'embedding visivo finale (Figura 3.2).

L'intuizione alla base di questo schema è che il *contenuto assoluto* di un frame (ad esempio, "quest'area contiene molti alberi") sia meno informativo, ai fini della descrizione di un cambiamento, di *come tale contenuto varia* da un frame al successivo ("l'area boschiva si è ridotta"). Calcolare esplicitamente differenza e somma tra frame consecutivi costringe il modulo a rappresentare direttamente questa variazione, anziché doverla dedurre implicitamente da una rappresentazione posizionale del contenuto assoluto di ciascun frame. La combinazione differenza/somma è complementare: la prima isola ciò che è cambiato, la seconda conserva il contesto in cui il cambiamento avviene. L'aggregazione finale in [first, last, mean, var] preserva informazione sulla *traiettoria* della sequenza — un cambiamento avvenuto all'inizio o alla fine del periodo osservato, oppure una dinamica costante contro una discontinua, danno luogo a vettori distinti.

3.2.4 Variante B: il Temporal Attention Encoder

La seconda variante sostituisce l'aggregazione a struttura fissa del Delta Encoder con un meccanismo di attenzione che *apprende* quali elementi della sequenza siano informativi. Il progetto segue L-TAE [29] (Sezione 2.3.4) nei suoi due tratti distintivi: una *master query* appresa per ciascuna testa di attenzione — non derivata dall'input, così da produrre direttamente un unico vettore per sequenza anziché uno per istante — e il *channel grouping*, per cui ogni testa opera su un sottoinsieme disgiunto dei canali. Ne risulta un modulo con circa la metà dei parametri del Delta Encoder.

Applicare L-TAE nella sua forma originale ai soli frame sarebbe però inadeguato al nostro compito. L'attenzione produce una combinazione convessa dei vettori in ingresso: dati i soli embedding per-frame, l'uscita è una rappresentazione della scena "media", in cui il cambiamento resta implicito — esattamente l'informazione che il retrieval deve invece discriminare. La variante qui proposta introduce quindi una modifica *change-aware*: l'insieme dei token su cui opera l'attenzione non è costituito dai soli frame, ma dalla loro unione con le differenze tra frame consecutivi,

$$\mathcal{T} \;=\; \underbrace{\{\mathbf{e}_t\}_{t=1}^{T}}_{\text{contenuto}} \;\cup\; \underbrace{\{\mathbf{e}_{t+1}-\mathbf{e}_t\}_{t=1}^{T-1}}_{\text{transizioni}},$$

ciascun token essendo sommato a due embedding appresi che ne codificano il *tipo* (frame oppure transizione) e la *posizione* temporale. In questo modo il modello può distribuire l'attenzione tra aspetto della scena e dinamica, decidendo autonomamente quanto pesare l'uno e l'altra, invece di riceverne il bilanciamento imposto dall'architettura come nel Delta Encoder. Le differenze sono considerate valide solo se lo sono entrambi i frame che le generano, il che estende la gestione delle maschere anche a questa variante. Lo schema complessivo del modulo è illustrato in Figura 3.3.

Il confronto tra le due varianti non è quindi soltanto una scelta implementativa, ma mette alla prova un'ipotesi precisa: se sia preferibile *imporre* la struttura del cambiamento (Delta Encoder) o *lasciarla apprendere* (Temporal Attention Encoder). L'esito sperimentale, e la lettura che se ne può dare alla luce della geometria dello spazio appreso (Sezione 2.2.4), sono discussi nel Capitolo 4.

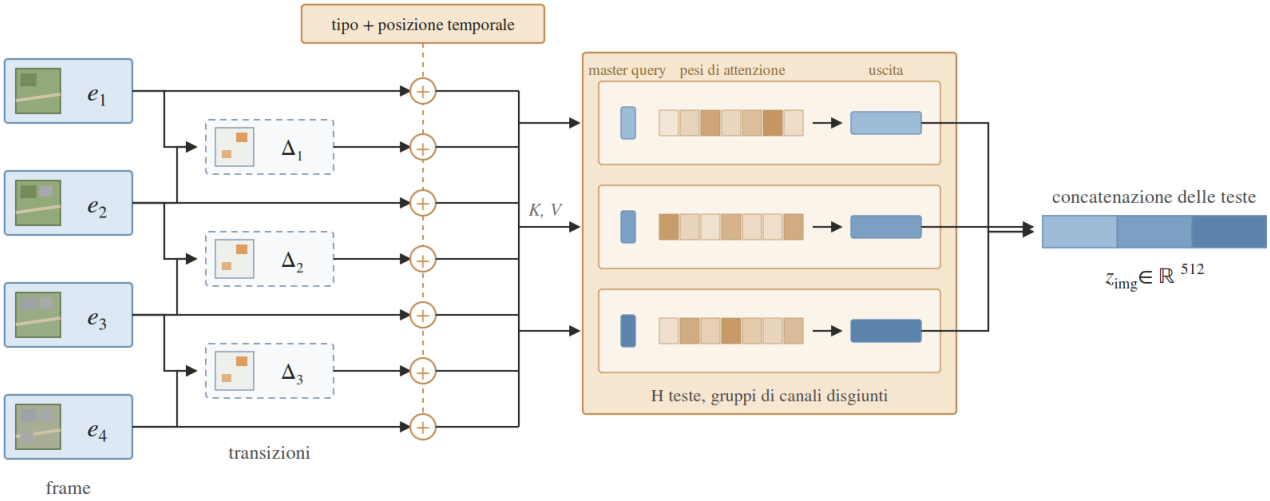


Figura 3.3: Dettaglio del Temporal Attention Encoder nella variante *change-aware*, per una sequenza di $T = 4$ frame. L'attenzione opera su un insieme di token che comprende sia i frame sia le transizioni tra frame consecutivi, ciascuno sommato a un embedding di tipo e di posizione temporale. Ogni testa dispone di una *master query* appresa e opera su un gruppo disgiunto di canali; le uscite delle teste vengono concatenate a formare l'embedding visivo finale $\mathbf{z}_{\text{img}}$.

3.2.5 Text Encoder: CLIP

Il ramo testuale utilizza l'encoder testuale di CLIP nella variante ViT-B/32 (Sezione 2.2.2): un Transformer di 12 layer con dimensione nascosta 512, che elabora la didascalia tokenizzata (lunghezza massima 77 token, sufficiente a coprire la quasi totalità delle didascalie generate o umane impiegate in questa tesi) e restituisce la rappresentazione associata al token speciale di fine sequenza come embedding dell'intera frase. Anche in questo caso solo l'ultimo layer del Transformer viene reso addestrabile, mentre gli undici restanti rimangono congelati (configurazione indicata nel seguito come *unfrozen light*, Sezione 3.5.2).

La scelta tra un text encoder generalista e uno già specializzato per il telerilevamento non è stata data per scontata: è stata verificata sperimentalmente confrontando CLIP con RemoteCLIP [31] (Sezione 2.3.5), un modello ri-addestrato su coppie immagine–testo di telerilevamento. Il confronto è riportato nel Capitolo 4: contrariamente all'attesa, il CLIP generalista assorbe l'adattamento di dominio in modo più stabile, mentre l'encoder già specializzato mostra maggiore rigidità nel riadattarsi al nostro stile di didascalie.

3.2.6 Projection Head e Spazio di Embedding Condiviso

Gli embedding prodotti dai due rami — 512-dimensionali per il testo (uscita diretta di CLIP) e 512-dimensionali per le immagini (uscita del modulo di fusione temporale) — non vengono confrontati direttamente, ma proiettati ciascuno attraverso una *projection head* dedicata: un piccolo MLP con struttura Linear($512 \to 512$)–LayerNorm–GELU–Linear($512 \to 512$). Questo modulo, interamente addestrabile, ha il compito di adattare le rappresentazioni prodotte da encoder con obiettivi di pre-addestramento molto diversi (masked autoencoding e allineamento radar-ottico per CROMA, next-token/masked language modeling per la componente linguistica di CLIP) a uno spazio semantico comune, ottimizzato specificamente per l'allineamento cross-modale. Dopo la proiezione, entrambi gli embedding vengono normalizzati ℓ_2 — operazione che vincola ogni vettore sulla superficie di un'ipersfera unitaria a 512 dimensioni, rendendo il prodotto scalare tra due vettori equivalente alla loro similarità coseno, la quantità effettivamente utilizzata dalla loss contrastiva (Sezione 3.2.7).

3.2.7 Loss Contrastiva con Temperatura Apprendibile

L'addestramento ottimizza la loss NT-Xent introdotta in Sezione 2.2.3 (Equazione 2.2), calcolata sugli embedding L_2-normalizzati $\mathbf{z}_{\text{img}}$ e $\mathbf{z}_{\text{txt}}$ prodotti dalle projection head. Coerentemente con CLIP, la temperatura τ non è fissata a un valore costante ma è un parametro apprendibile, aggiornato per backpropagation insieme al resto della rete e inizializzato a $\tau_0 = 1/e^{\log(1/0.07)}$ (equivalentemente, il suo logaritmo inverso è inizializzato a $\log(1/0.07)$, il valore adottato dal CLIP originale). Durante lo sviluppo del sistema è stata esplorata anche una variante con *memory bank* — una coda di embedding negativi calcolati nei batch precedenti, sulla falsariga di MoCo [17] (Sezione 2.2.1) — nel tentativo di aumentare il numero di negativi disponibili senza aumentare il batch size; la tecnica si è però rivelata instabile in questo contesto e non è stata adottata nella configurazione finale (Capitolo 4).

3.3 Dataset

Il sistema viene addestrato e valutato su quattro dataset con ruoli distinti nella pipeline sperimentale: SSL4EO-S12 (Sezione 2.3.2) come principale sorgente di addestramento contrastivo e unico benchmark propriamente *in-domain*; MOSAIC-SEN2-CC (Sezione 2.4.3) e SECOND-CC (Sezione 2.4.4) impiegati sia in addestramento sia in test; e LEVIR-CC (Sezione 2.4.5), riservato esclusivamente alla valutazione *out-of-domain*.

3.3.1 SSL4EO-S12

Come descritto in Sezione 2.3.2, SSL4EO-S12 fornisce sequenze temporali di immagini Sentinel-2 (12 bande ottiche, 120×120 pixel) di lunghezza variabile tra 2 e 4 frame, senza etichette semantiche native.

Selezione delle serie temporali. SSL4EO-S12 nasce per il pre-addestramento auto-supervisionato, non per il change captioning: le sue località sono campionate su scala globale senza alcun criterio legato al cambiamento, e la grande maggioranza delle serie mostra di conseguenza un'evoluzione minima o nulla — variazioni stagionali della vegetazione, differenze di illuminazione, al più rumore atmosferico. Addestrare su un pool così composto significherebbe fornire al modello un segnale fortemente sbilanciato verso l'assenza di cambiamento, con il duplice effetto di rendere le didascalie generate ripetitive (Sezione 3.4) e di offrire pochi esempi realmente informativi per il compito di retrieval.

Il pool di addestramento è stato perciò costruito applicando alle serie grezze una selezione a cascata, che scarta i campioni non adatti secondo criteri quantitativi espliciti:

- **validità dell'osservazione**: sono escluse le patch prevalentemente oceaniche e quelle in cui nuvole o neve coprono più del 20% dei pixel in almeno un frame, poiché in tali condizioni la superficie non è osservabile e ogni variazione misurata sarebbe un artefatto;
- **dinamica temporale non trascurabile**: sono scartate le serie "piatte", in cui la varianza temporale dell'NDVI è inferiore a una soglia minima, ossia quelle in cui nulla di misurabile accade lungo la sequenza;
- **entità del cambiamento**: si richiede che la variazione di NDVI tra il primo e l'ultimo frame superi una soglia ($|\Delta\text{NDVI}| > 0{,}15$) su almeno il 2% dei pixel dell'immagine, escludendo così le fluttuazioni di ampiezza irrilevante;
- **coerenza spaziale**: la regione che cambia deve formare almeno un agglomerato connesso di dimensione non inferiore al 2% dell'area. Questo criterio distingue un cambiamento reale e localizzato — un incendio, un'espansione urbana, un disboscamento — da variazioni sparse su pixel isolati, tipiche del rumore residuo del sensore.

Il pool risultante è stato infine **bilanciato a 10.000 campioni**, eliminando la sovra-rappresentazione della classe "Trees" presente nel pool grezzo. Il validation set held-out consta di **2.176 campioni**, verificati programmaticamente avere overlap nullo con il training set: ogni campione SSL4EO è annotato con metriche NDVI (variazione media, frazione di area interessata e varianza temporale) e con la

distribuzione delle classi LULC (tassonomia Dynamic World, 9 classi ufficiali più una classe segnaposto assegnata quando la maschera LULC di un tile è interamente invalida o nuvolosa — condizione che riguarda circa il 14% del training set). Le didascalie associate a ciascun campione non sono scritte da annotatori umani, ma generate automaticamente a partire da queste metriche quantitative tramite la pipeline descritta in Sezione 3.4.

3.3.2 MOSAIC-SEN2-CC

MOSAIC-SEN2-CC (Sezione 2.4.3) contribuisce **4.183 coppie** bi-temporali al training set e **528 coppie** al test set (dimensione nativa, confermata dal paper originale). Ogni coppia è annotata con cinque didascalie umane; in fase di addestramento viene utilizzata la prima didascalia disponibile per ciascun campione (Sezione 3.5.1), mentre in fase di valutazione tutte e cinque vengono considerate corrette ai fini del calcolo di Precision@K e mAP (Sezione 3.6.1).

3.3.3 SECOND-CC

SECOND-CC (Sezione 2.4.4) contribuisce **8.438 coppie** al training set e **1.227 coppie** al test set (dimensione nativa, split 7:1:2 confermato dal paper originale). Rispetto a MOSAIC-SEN2-CC, il vocabolario delle didascalie è più ripetitivo (circa 1.060 parole uniche complessive), una caratteristica che — come si vedrà — ne condiziona il comportamento come benchmark di retrieval, rendendo il compito paradossalmente più difficile a causa della ridotta discriminabilità testuale tra campioni diversi.

3.3.4 LEVIR-CC

LEVIR-CC (Sezione 2.4.5) non compare mai nel training set: le **1.929 coppie** del suo test set nativo vengono usate esclusivamente per misurare la generalizzazione del modello a un dominio visivo e testuale mai osservato in addestramento, secondo il protocollo out-of-domain descritto in Sezione 2.4.5.

3.4 Pipeline di Generazione delle Didascalie

Poiché SSL4EO-S12 non fornisce didascalie umane, il segnale testuale di addestramento per questo dataset viene generato automaticamente con un approccio *data-to-text*: metriche quantitative estratte dalle immagini (indici di vegetazione NDVI e distribuzione delle classi LULC, Sezione 2.3.1) vengono convertite in un prompt strutturato, sottoposto a un LLM (Qwen2.5-14B-Instruct [35]) che genera la didascalia in linguaggio naturale. Questa scelta consente di sfruttare l'enorme

scala di SSL4EO-S12 (e la sua natura multi-temporale) senza richiedere alcuna annotazione umana, al costo di uno stile testuale sistematicamente diverso da quello delle didascalie umane di MOSAIC-SEN2-CC e SECOND-CC — un aspetto del domain gap discusso esplicitamente nel Capitolo 1 e verificato empiricamente nel Capitolo 4.

3.4.1 Prompt Engineering Data-to-Text

Il generatore di prompt implementa diversi stili di prompt, selezionabili tramite configurazione Hydra. Lo stile adottato per SSL4EO-S12 è di tipo *causale-dinamico*: un system prompt istruisce l'LLM a produrre una narrazione temporale dinamica, con verbi attivi e una struttura causale esplicita (stato iniziale → transizione → stato finale), in contrasto con un primo stile esplorato durante lo sviluppo di tipo *accademico-rigido*, che produceva descrizioni statiche e rigide della sola distribuzione LULC finale, rivelatosi poco efficace (Capitolo 4). Per contenere il costo computazionale, i campioni privi di cambiamento rilevante (determinato da soglie sulle metriche NDVI) ricevono una didascalia estratta da un piccolo insieme di template predefiniti, senza invocare l'LLM, tramite un percorso di *early-exit*.

3.4.2 V1: Didascalie Primo-vs-Ultimo Frame

Nella prima versione della pipeline di captioning (V1), il confronto alla base della didascalia avviene esclusivamente tra il *primo* e l'*ultimo* frame della sequenza, indipendentemente dal numero di osservazioni intermedie: le metriche NDVI e LULC vengono calcolate come differenza netta tra stato iniziale e stato finale, e la didascalia risultante descrive il cambiamento complessivo osservato, senza riferimento esplicito alla sua evoluzione attraverso i frame intermedi. Questo schema è concettualmente vicino alla formulazione bi-temporale classica (Sezione 1.1), applicata anche a sequenze più lunghe collassando l'informazione intermedia.

3.4.3 V2: Didascalie Frame-by-Frame

La seconda versione (V2) supera questa limitazione analizzando i cambiamenti LULC *frame per frame* lungo l'intera sequenza temporale, anziché limitarsi al confronto tra i due estremi. La procedura, implementata in una procedura dedicata di estrazione delle transizioni cumulative, opera in questo modo: per ogni campione si calcola dapprima la distribuzione percentuale delle classi LULC in ciascun frame; si itera quindi sulle coppie di frame consecutivi, calcolando per ciascuna il delta percentuale di ogni classe e mantenendo solo le variazioni che superano una certa soglia; le transizioni frame-by-frame così ottenute vengono infine serializzate come tag strutturati in un campo dedicato del prompt e incluse nella richiesta fornita

all'LLM, il cui system prompt include un'istruzione esplicita di *sequenziamento temporale* che impone di raccontare la storia dei cambiamenti in ordine cronologico — ad esempio, "inizialmente si è verificato X, seguito da Y" — anziché limitarsi ad aggregarli in un'unica descrizione indifferenziata. Il risultato è un'unica didascalia compatta (entro i 77 token del tokenizer CLIP) che riflette l'intera evoluzione della sequenza. Come si mostrerà nel Capitolo 4, questo schema si rivela superiore allo schema V1 su tutte le metriche misurate, non solo su quelle più direttamente legate al contenuto LULC.

3.5 Procedura di Addestramento

Il modello viene addestrato con l'ottimizzatore AdamW [6] (learning rate 10^{-4}, weight decay 10^{-2}), uno scheduler a decadimento coseno con warmup iniziale, batch size 64 (128 negli esperimenti con dataset combinato in augmentation, Sezione 3.5.1), per un massimo di 100 epoche con early stopping (pazienza 10 epoche) sulla loss di validazione. La suddivisione interna del pool di training SSL4EO per l'early stopping — 87,5% dei campioni riceve gradiente diretto, il restante 12,5% è usato solo per il calcolo della loss di validazione, con seed fisso 42 — è concettualmente e proceduralmente distinta dal validation set held-out esterno, impiegato esclusivamente in fase di valutazione finale e mai osservato durante l'addestramento.

Risorse di calcolo e vincoli progettuali. Tutti gli esperimenti sono stati eseguiti su un cluster di calcolo ad alte prestazioni con GPU NVIDIA A40 (48 GB di memoria), una per addestramento. Questa disponibilità ha condizionato in modo sostanziale lo spazio delle scelte progettuali, e va tenuta presente nella lettura dei risultati. In primo luogo ha determinato la *taglia* dei modelli impiegati: sia l'encoder visivo sia quello testuale sono varianti *base* (ViT-B, circa 90 milioni di parametri ciascuno), mentre varianti *large* o *huge* — che in letteratura mostrano guadagni consistenti proprio sui compiti di allineamento cross-modale — avrebbero richiesto memoria e tempi di addestramento non compatibili con le risorse a disposizione. In secondo luogo ha condizionato la *profondità del fine-tuning*: sbloccare l'intero backbone di entrambi gli encoder, oltre a esporre al rischio di catastrophic forgetting discusso in Sezione 3.5.2, avrebbe richiesto di ridurre drasticamente la dimensione del batch, penalizzando la loss contrastiva — che dal numero di negativi per batch trae gran parte del proprio segnale. La configurazione adottata (backbone in gran parte congelati, ultimo layer addestrabile, batch di 128 campioni) è quindi il punto di equilibrio tra capacità di adattamento al dominio e vincoli di memoria, non un ottimo assoluto: parte del margine di miglioramento discusso nelle conclusioni resta subordinata alla disponibilità di risorse di calcolo maggiori.

3.5.1 Strategia di Combinazione dei Dataset

Una delle scelte progettuali più rilevanti, emersa da un percorso sperimentale documentato nel Capitolo 4, è l'inclusione diretta di MOSAIC-SEN2-CC e SECOND-CC nel training contrastivo, insieme a SSL4EO-S12, anziché il ricorso — esplorato in una fase precedente dello sviluppo e rivelatosi inefficace — a tecniche di adattamento testuale (*caption adaptation*) applicate a un modello addestrato sul solo SSL4EO. Il training set combinato consta quindi di SSL4EO-S12 (10.000 serie), MOSAIC-SEN2-CC (4.183 coppie) e SECOND-CC (8.438 coppie), mescolati direttamente nello stesso ciclo di addestramento.

3.5.2 Sblocco Selettivo dei Layer

Come anticipato in Sezione 3.2.1 e Sezione 3.2.5, la configurazione di fine-tuning adottata — indicata come *unfrozen_light* — sblocca esclusivamente l'ultimo layer di CROMA e l'ultimo layer del text encoder CLIP, mantenendo tutti gli altri parametri di entrambi gli encoder congelati. Questa scelta è stata confrontata sperimentalmente (Capitolo 4) sia con configurazioni completamente congelate, sia con l'uso di varianti CLIP già specializzate per il telerilevamento (RemoteCLIP, Sezione 2.3.5) nella stessa configurazione di sblocco parziale: un CLIP generalista sblocca in modo stabile, mentre un encoder già specializzato mostra segni di *catastrophic forgetting* anche quando solo un singolo layer viene reso addestrabile.

3.5.3 Temporal Dropout

Poiché i dataset di change captioning downstream (MOSAIC-SEN2-CC, SECOND-CC, LEVIR-CC) forniscono esclusivamente coppie bi-temporali ($T = 2$), mentre SSL4EO-S12 offre sequenze di lunghezza variabile fino a $T = 4$, il modello rischierebbe di specializzarsi implicitamente su sequenze più lunghe durante l'addestramento su SSL4EO, per poi trovarsi in un regime diverso in fase di valutazione downstream. Il *Temporal Dropout* mitiga questo disallineamento: quando un campione SSL4EO ha lunghezza superiore a 2 e con probabilità 0,5, la sequenza viene sottocampionata a $T = 2$ (primo e ultimo frame) o $T = 3$ durante l'addestramento, simulando artificialmente la struttura bi-temporale dei dataset downstream anche quando si allena su SSL4EO. Il supporto nativo a maschere 2D in entrambe le varianti del modulo di fusione temporale (Sezione 3.2.2) rende questa augmentation possibile senza alcuna modifica architetturale.

3.6 Protocollo di Valutazione

3.6.1 Metriche di Retrieval

Per ciascun dataset e ciascuna direzione (I2T e T2I) vengono calcolate tutte le metriche introdotte in Sezione 2.5: Recall@K per $K \in \{1,5,10\}$, Precision@K e mAP (rilevanti per i dataset con più didascalie corrette per immagine), e Mean/Median Rank, calcolate ogni volta su un ranking esatto contro tutti i candidati del pool considerato — per i benchmark downstream il rispettivo test set nativo completo, per SSL4EO-S12 il pool a dimensione fissa descritto in Sezione 3.6.3. Per l'encoder visivo viene inoltre calcolata una metrica ausiliaria, *I2I Recall@K by-label*: dato un embedding immagine query, si verifica se le K immagini più simili nello spazio condiviso condividono la stessa classe LULC dominante, un proxy della qualità pura dell'encoder visivo indipendente dall'allineamento testuale.

3.6.2 Classificazione Zero-Shot dell'Uso del Suolo

Sfruttando il paradigma zero-shot introdotto da CLIP (Sezione 2.2.2), il modello viene valutato anche su un compito di classificazione della classe LULC dominante di ciascun campione SSL4EO, tramite *prompt ensembling*: per ciascuna delle classi candidate si costruiscono più frasi-prompt equivalenti, se ne mediano gli embedding testuali, e si assegna al campione la classe la cui frase mediata risulta più simile al suo embedding visivo. Il report di classificazione include precision, recall e F1 per singola classe, oltre alle medie macro e pesata, calcolate escludendo le classi con supporto nullo nel set di valutazione (nello specifico, la classe "Water", praticamente assente rispetto alle altre sia dal training sia dal validation set SSL4EO) — un dettaglio metodologicamente rilevante, discusso più avanti in questa sezione.

3.6.3 Comparabilità tra Pool di Dimensione Diversa

I quattro test set impiegati in questa tesi hanno pool di retrieval di dimensione molto diversa (da 2.176 candidati per SSL4EO a oltre 9.000 per LEVIR-CC): a parità di modello, un pool più piccolo produce meccanicamente una Recall@K più alta per puro effetto della dimensione, rendendo il confronto diretto dei valori grezzi tra dataset fuorviante. Un primo riferimento di lettura, applicabile a ogni pool, è il *fattore di lift*: il rapporto tra il valore osservato e il valore atteso sotto un ranking puramente casuale per quello specifico pool (calcolabile analiticamente dalla dimensione del pool e dal numero di didascalie corrette per campione, tramite la distribuzione ipergeometrica per il caso multi-didascalia). Questo fattore è *scale-invariant* e quindi confrontabile direttamente tra dataset di dimensione diversa, a differenza della percentuale grezza; il valore atteso a caso per il Mean Rank è

semplicemente metà della dimensione del pool, ed è il riferimento rispetto al quale vanno letti i mean rank riportati nel Capitolo 4.

Confronto a parità di dimensione del pool. Le metriche di retrieval dipendono da quanti candidati compongono il pool: recuperare la didascalia corretta tra 528 alternative è più facile che farlo tra 2.176. Confrontare direttamente SSL4EO, il cui pool non è fissato da alcun protocollo esterno, con benchmark la cui dimensione di test è stabilita dai rispettivi paper sarebbe quindi fuorviante. Le metriche SSL4EO riportate nel Capitolo 4 sono perciò sempre misurate su un pool ridotto a 528 candidati — la stessa dimensione del test di MOSAIC-SEN2-CC — anziché sul pool intero. È la prassi consolidata nel retrieval cross-modale: su MS-COCO si riportano abitualmente i risultati su un pool ridotto a 1.000 immagini, mediati su cinque partizioni, accanto a quelli sull'intero test set da 5.000 [36, 20].

La riduzione non viene effettuata una sola volta: un singolo sottoinsieme potrebbe risultare per caso più facile o più difficile della media. L'estrazione di 528 campioni (senza ripetizioni al suo interno) è quindi ripetuta 200 volte in modo indipendente, ricalcolando ogni volta il ranking tra i soli campioni estratti; si riportano media e deviazione standard sui 200 valori così ottenuti. Rispetto alle cinque partizioni disgiunte di MS-COCO si usano estrazioni casuali perché 2.176 campioni non sono divisibili in partizioni da 528 senza scarti; il maggior numero di ripetizioni fornisce inoltre una stima più stabile. Poiché la procedura riusa gli embedding già calcolati, il costo computazionale è trascurabile.

Capitolo 4

Esperimenti

Questo capitolo riporta il setup sperimentale (Sezione 4.1) e ripercorre brevemente la storia dello sviluppo del sistema (Sezione 4.2), per poi presentare i risultati del modello migliore (Sezione 4.3) e gli studi di ablazione che ne hanno motivato le scelte progettuali: lo stile delle didascalie (Sezione 4.4), il ruolo di SSL4EO-S12 (Sezione 4.5) e la scalabilità del training (Sezione 4.6). Si presenta quindi un'analisi qualitativa dello spazio latente appreso (Sezione 4.7), cui fa seguito un'analisi quantitativa della geometria dello spazio condiviso e degli interventi architetturali che ne derivano (Sezione 4.8). Chiude il capitolo una discussione trasversale dei risultati (Sezione 4.9). Salvo diversa indicazione, tutte le metriche riportate in questo capitolo sono state misurate secondo il protocollo di valutazione descritto in Sezione 3.6: validation set SSL4EO-S12 realmente held-out con didascalie rigenerate, test set nativi completi per ciascun benchmark. Per SSL4EO-S12 le metriche di retrieval sono sempre riportate come bootstrap a dimensione fissa $N = 528$ (Sezione 3.6.3, 200 sottocampionamenti, seme 42), e non sul pool completo: questa è l'unica convenzione che rende i valori confrontabili sia tra modelli sia con il test set MOSAIC-SEN2-CC, che ha esattamente quella dimensione, e sempre in direzione testo–immagine (T2I) salvo diversa indicazione. A partire dall'ablazione sullo stile delle didascalie (Sezione 4.4), che stabilisce la superiorità di V2 su ogni metrica misurata, tutti gli esperimenti successivi adottano V2 come configurazione di partenza: è a essa che si riferisce l'etichetta "V2 (riferimento)" nelle tabelle seguenti. I modelli dell'analisi geometrica (Sezione 4.8) sono inoltre stati addestrati ex novo, con seme di inizializzazione fissato esplicitamente.

4.1 Setup Sperimentale

Tutti gli esperimenti sono stati eseguiti sul cluster HPC Legion@PoliTo, su GPU NVIDIA A40/A100 (partizione `gpu_a40`), mentre le fasi di preparazione dati sono

state eseguite su nodi CPU dedicati (partizione `cpu_skylake`). L'intera pipeline — preparazione dati, generazione delle didascalie, addestramento contrastivo e valutazione — è implementata in PyTorch con la libreria HuggingFace Transformers per gli encoder pre-addestrati, orchestrata tramite Hydra per la gestione delle configurazioni sperimentali, con i job di addestramento e valutazione sottomessi come job SLURM. Salvo dove diversamente specificato, gli iperparametri di addestramento sono quelli riportati in Sezione 3.5 (ottimizzatore AdamW, learning rate 10^{-4}, batch size 64–128, early stopping con pazienza 10 epoche sulla loss di validazione). Il protocollo di valutazione — descritto in dettaglio in Sezione 3.6 — è stato tenuto rigorosamente fisso tra tutti i modelli confrontati in questo capitolo, condizione necessaria perché i confronti tra le diverse configurazioni sperimentali siano validi.

4.2 Storia dello Sviluppo

Il sistema descritto nel Capitolo 3 è il risultato di un percorso sperimentale iterativo, riassunto in questa sezione perché motiva diverse scelte progettuali altrimenti poco intuitive. **Attenzione metodologica**: i valori numerici di questa sezione, tratti dai log originali di ciascuna fase, precedono la revisione del protocollo di valutazione; sono riportati esclusivamente per illustrare la *direzione relativa* delle decisioni prese (quale variante ha motivato la successiva), non come misure assolute affidabili — quelle sono riportate, per il modello finale, nelle sezioni successive.

Una prima versione del sistema (fase 1) impiegava CROMA completamente congelato, un semplice *global pooling* temporale (media degli embedding dei frame, senza alcun modulo di aggregazione dedicato) e didascalie generate con lo stile di prompt *accademico-rigido* (Sezione 3.4.1), che si limitava a descrivere rigidamente la distribuzione LULC finale. Questa configurazione soffriva di un severo collasso lessicale delle didascalie generate (un indice di diversità testuale, *type-token ratio*, pari a soli 0,0005). Una seconda fase ha introdotto lo stile *causale-dinamico*, poi adottato in via definitiva, con una narrativa temporale dinamica al posto della descrizione statica, confermando sperimentalmente che l'encoder visivo congelato, da solo, non fosse sufficiente: il fine-tuning si è rivelato necessario. Una terza fase ha sostituito il global pooling con il Delta Encoder (Sezione 3.2.3), rivelando un fatto diagnosticamente importante: pur ottenendo un I2I Recall@1 su MOSAIC-SEN2-CC del 68,2% — a dimostrazione che l'encoder visivo generalizzava bene — il retrieval cross-modale restava pressoché nullo, isolando il text encoder (e non l'encoder visivo) come principale collo di bottiglia. Una quarta fase ha esplorato la riscrittura offline delle didascalie umane di MOSAIC-SEN2-CC tramite template multipli e ha introdotto il *Temporal Dropout* (Sezione 3.5.3), senza tuttavia risolvere il problema di fondo. La svolta è arrivata in una quinta fase, con tre cambiamenti architetturali introdotti simultaneamente: la ristrutturazione

del dataset di addestramento con l'inclusione diretta di MOSAIC-SEN2-CC e SECOND-CC nel training contrastivo (Sezione 3.5.1, in sostituzione del fallimentare adattamento testuale isolato), lo sblocco selettivo di un solo layer per encoder con CLIP generalista anziché una variante già specializzata (Sezione 3.5.2), e l'introduzione della temperatura apprendibile nella loss NT-Xent (Sezione 3.2.7) — una *memory bank* esplorata nella stessa fase si è invece rivelata instabile e non è stata adottata. Questa configurazione (*best model*, Sezione 4.3) è il punto di partenza per tutti gli studi di ablazione descritti nel resto del capitolo, condotti in una sesta e ultima fase di sviluppo.

4.3 Risultati del Modello Migliore

Il modello migliore adotta la configurazione descritta nel Capitolo 3: CROMA e CLIP text encoder con un solo layer sbloccato ciascuno, dataset di training combinato (SSL4EO-S12 10k + MOSAIC-SEN2-CC train + SECOND-CC train), didascalie SSL4EO in stile V1 (Sezione 3.4.2). La Tabella 4.1 riporta le metriche di retrieval sul validation set SSL4EO-S12 held-out, a dimensione di pool fissa $N = 528$.

Metrica	**I2T**	**T2I**
Recall@1	9,1%±0,5	8,7%±0,6
Recall@5	26,5%±1,2	25,8%±1,3
Recall@10	39,0%±1,5	38,3%±1,4
Mean Rank	97	99

Tabella 4.1: Retrieval SSL4EO-S12 del modello migliore (validation set held-out, bootstrap a $N = 528$, media ± deviazione standard su 200 sottocampionamenti). Il mean rank di 99 è nettamente e sistematicamente migliore del valore atteso sotto ranking casuale (≈264,5, metà della dimensione del pool) — un segnale di apprendimento reale e significativo.

La qualità pura dell'encoder visivo, misurata indipendentemente dall'allineamento testuale tramite I2I Recall@K by-label, risulta nettamente superiore: 71,9% a $K = 1$, 92,2% a $K = 5$, 96,6% a $K = 10$ — a conferma che il collo di bottiglia principale del sistema risiede nell'allineamento cross-modale, non nella qualità delle rappresentazioni visive in sé. Sul compito di classificazione zero-shot dell'uso del suolo (Sezione 3.6.2, 2.091 campioni valutati), il modello raggiunge un'accuratezza del 52,1%, una precision macro del 47,7% e un F1 macro (esclusa la classe priva di supporto "Water") del 43,3% (38,5% se calcolato sul totale grezzo delle 9 classi originali).

Dataset	**Pool**	**T2I@10**	**I2T@10**	**mAP**	**Lift**
MOSAIC-SEN2-CC	528	24,1%	25,4%	4,0%	12,7×
SECOND-CC	1.227	19,7%	19,6%	3,2%	**24,2**×
LEVIR-CC (out-of-domain)	1.929	2,9%	3,1%	1,7%	5,6×
SSL4EO-S12 (per confronto)	528	38,3%	39,0%	—	20,2×

Tabella 4.2: Retrieval T2I e I2T del modello migliore sui benchmark downstream (test set nativi completi), con SSL4EO-S12 riportato per confronto. La colonna *Lift* esprime la T2I Recall@10 come multiplo del valore atteso sotto ranking casuale, pari a $10/N$ sul rispettivo pool: è l'unica normalizzazione che rende confrontabili benchmark di dimensione diversa. LEVIR-CC, mai visto in addestramento, misura la generalizzazione out-of-domain pura.

La Tabella 4.2 riporta le metriche sui benchmark downstream, sui rispettivi test set nativi completi. Poiché i tre pool hanno dimensione molto diversa — da 528 a 1.929 candidati — e una stessa Recall@10 è tanto più difficile da ottenere quanto più grande è il pool, la tabella affianca alle percentuali il fattore di *lift* sul caso casuale.

Letta in questi termini, la graduatoria fra i due benchmark in-domain si rovescia. In percentuale grezza MOSAIC-SEN2-CC (24,1%) appare nettamente superiore a SECOND-CC (19,7%), ma il primo sceglie fra 528 candidati e il secondo fra 1.227: normalizzando, SECOND-CC raggiunge un lift di 24,2× contro i 12,7× di MOSAIC-SEN2-CC, ed è quindi il dominio su cui il modello si discosta di più dal caso. Anche LEVIR-CC, il cui 2,9% grezzo potrebbe suggerire un fallimento completo, si colloca in realtà a 5,6× il caso casuale: la generalizzazione out-of-domain è modesta ma non nulla, un giudizio che la sola percentuale non permetterebbe di formulare.

4.4 Ablazione sullo Stile delle Didascalie: V1 vs V2

Il modello V2 è identico al modello migliore in ogni aspetto tranne lo stile delle didascalie SSL4EO-S12, generate secondo lo schema frame-by-frame V2 anziché lo schema primo-vs-ultimo-frame V1 (Sezioni 3.4.2–3.4.3).

I due schemi differiscono in ciò che viene misurato sulla sequenza prima di interrogare l'LLM, non nel modello di linguaggio né nel resto della pipeline. In V1 le metriche NDVI e la distribuzione delle classi LULC vengono calcolate come differenza netta tra il primo e l'ultimo frame: il prompt contiene quindi un solo bilancio complessivo, e la didascalia che ne risulta descrive lo stato iniziale, quello finale e il salto tra i due. In V2 la distribuzione LULC viene invece calcolata su

Metrica (SSL4EO-S12)	**V1 (best)**	**V2**	**Δ**
T2I Recall@1	8,7%	9,5%	+0,8 pp
T2I Recall@10	38,3%	40,9%	**+2,6 pp**
Mean Rank T2I	99	97,2	migliore
Zero-Shot LULC Acc.	52,1%	64,1%	**+12,0 pp**
Zero-Shot LULC F1	43,3%	51,5%	**+8,2 pp**

Tabella 4.3: Confronto tra didascalie V1 (confronto primo/ultimo frame) e V2 (frame-by-frame) sul validation set SSL4EO-S12, retrieval a $N = 528$. V2 supera V1 su ogni metrica misurata, incluso il retrieval — non solo sullo zero-shot LULC, come inizialmente suggerito da una misurazione condotta su un pool di valutazione non correttamente separato dal training set.

ogni frame, e le variazioni tra frame consecutivi che superano una soglia vengono serializzate come transizioni ordinate nel tempo; il system prompt impone all'LLM di raccontarle in sequenza cronologica anziché aggregarle. La Tabella 4.3 confronta i due modelli.

Il vantaggio di V2 si spiega con il fatto che il bilancio netto è una statistica *non iniettiva* sulla sequenza: percorsi temporali diversi possono produrre lo stesso saldo tra primo e ultimo frame, e in particolare un'area che perde vegetazione per poi riacquistarla riceve in V1 una didascalia di sostanziale stabilità, identica a quella di un'area che non è mai cambiata. Campioni visivamente distinti finiscono così a condividere lo stesso testo, e la loss contrastiva non ha alcun modo di separarli: sono collisioni di retrieval iscritte nel dato, non nel modello. Descrivendo il percorso anziché il saldo, V2 le elimina — e nel farlo nomina esplicitamente le classi LULC coinvolte in ciascuna transizione anziché le sole due distribuzioni estreme, il che spiega perché il guadagno maggiore si osservi sulla classificazione zero-shot, il compito che dipende più direttamente da quel vocabolario.

Sui benchmark downstream, che non dipendono dallo stile delle didascalie SSL4EO-S12, la differenza tra i due modelli è trascurabile ed entro il rumore sperimentale, come atteso, poiché V2 modifica esclusivamente le didascalie del dominio SSL4EO-S12. Il vantaggio di V2 è quindi specifico al dominio in cui la modifica è stata applicata, un'ulteriore conferma indiretta della validità del confronto.

Metrica	Con SSL4EO (best)	Senza SSL4EO	Δ
SSL4EO T2I Recall@1	8,7%	0,2%	−8,5 pp
SSL4EO T2I Recall@10	38,3%	2,3%	−36,0 pp
SSL4EO Mean Rank T2I	99	236,3 (≈caso)	crollo totale
Zero-Shot LULC Accuracy	52,1%	35,2%	−16,9 pp
Zero-Shot LULC F1	43,3%	10,9%	−32,4 pp
I2I by-label Recall@1	71,9%	63,8%	−8,1 pp
MOSAIC-SEN2-CC T2I@10	24,1%	25,2%	+1,1 pp
SECOND-CC T2I@10	19,7%	18,2%	−1,5 pp

Tabella 4.4: Ablazione: modello addestrato senza SSL4EO-S12 nel training set. Il mean rank su SSL4EO-S12 crolla a 236,3, prossimo al valore atteso sotto ranking casuale (≈264,5) — contro 99 quando SSL4EO-S12 è incluso in addestramento.

4.5 Il Ruolo di SSL4EO-S12 come Regolarizzatore

Per isolare il contributo di SSL4EO-S12 al training combinato (Sezione 3.5.1), è stato addestrato un modello identico in ogni altro aspetto ma escludendo SSL4EO-S12 dal training set, lasciando MOSAIC-SEN2-CC e SECOND-CC invariati. La Tabella 4.4 riporta il confronto.

Questo è probabilmente il risultato più netto e conclusivo di tutta la sperimentazione: senza SSL4EO-S12 in addestramento, la capacità di retrieval cross-modale fine-grained sul dominio SSL4EO-S12 collassa completamente, mentre resta sostanzialmente invariata sui benchmark downstream (il modello vede comunque MOSAIC-SEN2-CC e SECOND-CC in addestramento in entrambi i casi). SSL4EO-S12 non funge dunque solo da regolarizzatore per la classificazione zero-shot, come inizialmente ipotizzato nelle fasi preliminari dello sviluppo (Sezione 4.2), ma è una componente *necessaria* per qualunque capacità di retrieval cross-modale fine-grained: senza di essa, il modello impara ad allineare grossolanamente immagini e testo sui soli domini downstream visti in training, ma non sviluppa alcuna sensibilità al segnale di cambiamento più sottile presente in SSL4EO-S12.

La spiegazione più plausibile risiede in ciò che a prima vista sembrerebbe il principale difetto di SSL4EO-S12 come sorgente testuale: la sua scarsa varietà. Le sue didascalie sono generate a partire da un insieme ristretto e controllato di grandezze — indici di vegetazione e classi LULC — e ruotano quindi attorno a un vocabolario di poche decine di concetti, laddove le didascalie umane — in particolare quelle di MOSAIC-SEN2-CC — attingono a un lessico assai più ampio. Ma sono anche di gran lunga le più numerose del training set combinato (10.000 serie contro 4.183 e 8.438 coppie), e questo rovescia il segno della limitazione: ogni

concetto ricorre in migliaia di contesti visivi diversi, fornendo alla loss contrastiva la ripetizione necessaria ad ancorarlo stabilmente nello spazio condiviso. È la differenza tra vedere molte volte gli stessi pochi concetti e vederne molti una volta sola; per il retrieval conta la prima. Le didascalie umane, prese da sole, offrono un vocabolario più ampio ma troppo poche occorrenze per concetto perché l'allineamento si consolidi — e il crollo dell'F1 zero-shot dal 43,3% al 10,9%, ben più marcato di quello dell'accuratezza, mostra esattamente questo: ciò che si perde non è una capacità generica, ma il radicamento delle singole categorie semantiche. A ciò si aggiunge che SSL4EO-S12 è l'unica sorgente realmente multi-temporale del training set: è quindi anche l'unica in cui il modulo di fusione riceva supervisione su sequenze più lunghe di due istanti.

4.6 Esperimenti di Scaling

Un esperimento successivo ha applicato simultaneamente due modifiche al training del modello migliore, motivate dall'ipotesi che ciascuna, isolatamente, producesse un miglioramento: l'ampliamento di SSL4EO-S12 da 10.000 a 20.000 campioni bilanciati e l'uso di tutte le cinque didascalie umane disponibili per immagine in MOSAIC-SEN2-CC/SECOND-CC anziché la sola prima — quest'ultima moltiplicando per circa cinque le righe di training di questi due dataset. Il risultato (Tabella 4.5, colonna *scaled_20k*) non è un miglioramento ma un trade-off netto: tutto ciò che è specifico di SSL4EO-S12 peggiora, i due benchmark downstream migliorano.

L'ipotesi più plausibile è che l'espansione delle didascalie multiple, proporzionalmente molto più aggressiva (5×) del raddoppio di SSL4EO-S12, alteri il bilanciamento del batch a sfavore di quest'ultimo nonostante il suo pool nominale sia cresciuto. Un secondo esperimento lo verifica: combinando le didascalie V2, che modificano solo SSL4EO-S12, con un'*augmentation* testuale che modifica solo i due dataset downstream — a ciascuna didascalia umana di MOSAIC-SEN2-CC e SECOND-CC viene affiancata una riscrittura tramite LLM, così che ogni immagine sia vista due volte con due formulazioni diverse — e lasciando SSL4EO-S12 ai suoi 10.000 campioni originari, ci si attenderebbe di sommare benefici già misurati separatamente. Si ottiene invece lo stesso trade-off, riprodotto quasi cifra per cifra pur avendo rimosso la variabile della scala: è l'espansione delle didascalie multiple, e il peso che essa sposta nel batch, la causa dominante del fenomeno.

Nessuna delle due configurazioni è stata quindi adottata come default. Il risultato indica però una via d'uscita concreta: un sampler di bilanciamento per dataset, che disaccoppi il numero di didascalie per immagine dal peso del dataset nel batch, permetterebbe di ottenere il beneficio del multi-caption senza il costo sul segnale SSL4EO-S12 (Sezione 5.3).

Metrica	V2 (riferimento)	Scaled_20k	V2+mix
SSL4EO T2I Recall@10	**40,9%**	35,1%	37,7%
Zero-Shot LULC Accuracy	**52,1%**	45,2%	44,2%
MOSAIC-SEN2-CC T2I@10 (raw)	24,1%	**26,4%**	**26,5%**
SECOND-CC T2I@10 (raw)	19,7%	**19,9%**	**20,5%**

Tabella 4.5: Il trade-off in-domain/downstream negli esperimenti di scaling. Due configurazioni molto diverse — scaled_20k, che raddoppia SSL4EO-S12 e quintuplica le didascalie downstream, e V2+mix, che lascia invariata la scala di SSL4EO-S12 — producono lo stesso identico pattern.

4.7 Analisi dello Spazio Latente

Per complementare le metriche quantitative con un'ispezione della struttura dello spazio appreso, gli embedding sono stati proiettati in due dimensioni tramite UMAP [37], preferito a t-SNE per la sua migliore capacità di preservare la macrostruttura globale dei dati e non i soli vicinati locali. L'analisi considera otto gruppi — immagini e testi, questi ultimi sia raw sia adapted dove disponibili, di SSL4EO-S12, MOSAIC-SEN2-CC e SECOND-CC — con 400 campioni ciascuno. LEVIR-CC è escluso perché mai visto in addestramento: la posizione dei suoi embedding descriverebbe il comportamento del modello fuori dominio, non la geometria dello spazio che esso ha effettivamente appreso. Sono escluse, in tutti i domini, le didascalie di "nessun cambiamento": in SSL4EO-S12 derivano da template fissi (Sezione 3.4.1), mentre nei benchmark con annotazione umana consistono in poche formule ripetute alla lettera (una sola frase copre il 28% del test set di SECOND-CC). In entrambi i casi si tratta di stringhe artificialmente coincidenti, che occupano un unico punto dello spazio e che la proiezione renderebbe come isolotti privi di significato geometrico.

Il primo elemento che la proiezione mette in evidenza (Figura 4.1) riguarda il divario visivo tra domini, e corregge un'attesa naturale ma sbagliata. Non sono i benchmark downstream a raggrupparsi contrapponendosi ai dati di addestramento: il divario segue il *sensore*. MOSAIC-SEN2-CC condivide con SSL4EO-S12 la sorgente Sentinel-2, e le rispettive immagini occupano un unico addensamento indistinguibile; le immagini di SECOND-CC, acquisite ad alta risoluzione (Sezione 2.4.4), formano un gruppo nettamente staccato da entrambe. Che il modello separi proprio i domini acquisiti da sensori diversi, e confonda quelli acquisiti dallo stesso, indica rappresentazioni visive che codificano le caratteristiche fisiche dell'acquisizione più della provenienza del dataset.

Il secondo riguarda la posizione reciproca delle due modalità. Immagini e testi occupano regioni distinte, come atteso in un modello CLIP-like, ma questa

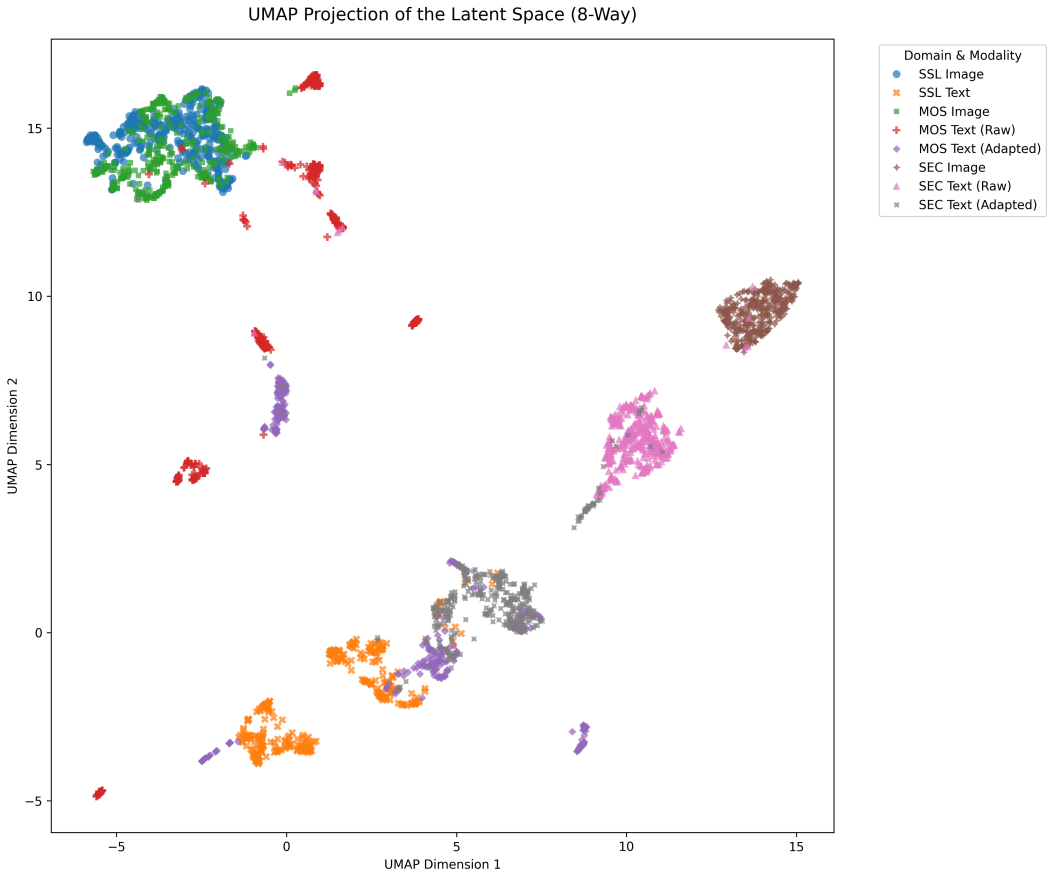


Figura 4.1: Proiezione UMAP dello spazio di embedding condiviso, otto sotto-domini, al netto delle didascalie di "nessun cambiamento". Immagini SSL4EO-S12 e MOSAIC-SEN2-CC occupano un unico addensamento, mentre le immagini SECOND-CC formano un gruppo nettamente separato. Gli addensamenti isolati appartengono alle didascalie umane di MOSAIC-SEN2-CC e raggruppano ciascuno un singolo tipo di fenomeno — incendi, prosciugamento di corpi d'acqua, inondazioni, fusione di nevi e ghiacci — descritto con un lessico specifico e ricorrente.

separazione non predice la qualità del recupero: SSL4EO-S12, il dominio le cui didascalie sono più lontane dalle rispettive immagini, è a parità di dimensione del pool quello su cui il retrieval funziona *meglio* (Sezione 4.3). Ciò che conta per il ranking è infatti la separazione *relativa* all'interno del dominio — sulle coppie SSL4EO-S12 la similarità media vale 0,547 per le coppie corrette contro 0,232 per quelle casuali — mentre lo scostamento globale tra le due nuvole si comporta come una traslazione rigida, che il confronto per similarità non vede. Che la struttura condivisa dalle due modalità sia reale è confermato dal *Centered Kernel Alignment*, che annulla per costruzione lo scostamento tra i centroidi: vale 0,303 sugli accoppiamenti corretti e crolla a 0,004 quando la corrispondenza immagine–didascalia viene permutata a caso. La Sezione 4.8 riprenderà questa lettura in termini quantitativi, mostrando che tale scostamento può essere annullato a posteriori con un guadagno misurabile.

Un terzo elemento riguarda infine le didascalie umane riscritte tramite LLM (*adapted*), prodotte per ridurne il divario stilistico rispetto a quelle generate automaticamente. Nella proiezione esse non si sovrappongono alle didascalie umane da cui derivano, ma occupano regioni proprie, in parte contigue a quella delle didascalie generate per SSL4EO-S12: la riscrittura sposta dunque effettivamente il testo verso lo stile bersaglio. Lo fa però in modo solo parziale e disomogeneo tra i due corpora — il 61% delle didascalie riscritte di SECOND-CC risulta più vicino al centroide SSL4EO-S12 che a quello delle proprie originali, contro il 46% di MOSAIC-SEN2-CC. Il risultato è coerente con il solo impiego di tale riscrittura che si sia rivelato praticabile — come *augmentation* affiancata al testo originale, mai in sua sostituzione (Sezione 4.6): essa sposta una parte del corpus senza sostituirne in modo uniforme il registro.

La struttura appena descritta è quantificabile senza passare da una proiezione. La Figura 4.2 riporta la similarità coseno tra i centroidi dei sei sotto-domini principali, con le righe ordinate per modalità: la diagonale, identicamente pari a uno e priva di contenuto informativo, è mascherata perché non comprima la scala dei colori sui valori effettivamente osservati.

I valori confermano numericamente la prima lettura e precisano la seconda. Il raggruppamento per sensore è netto: 0,95 tra i centroidi immagine di SSL4EO-S12 e MOSAIC-SEN2-CC, contro 0,31 e 0,42 verso SECOND-CC. La distanza tra modalità, invece, non è una costante del modello ma dipende dal dominio: fra immagini e rispettive didascalie si passa da +0,75 su MOSAIC-SEN2-CC e +0,71 su SECOND-CC a −0,14 su SSL4EO-S12. È proprio SSL4EO-S12, il dominio in cui le due modalità sono più lontane, quello su cui il retrieval funziona meglio a parità di pool: la vicinanza tra i centroidi non misura la qualità dell'allineamento, ma solo la sovrapposizione globale delle due nuvole — una traslazione rigida, cui il confronto per similarità è insensibile.

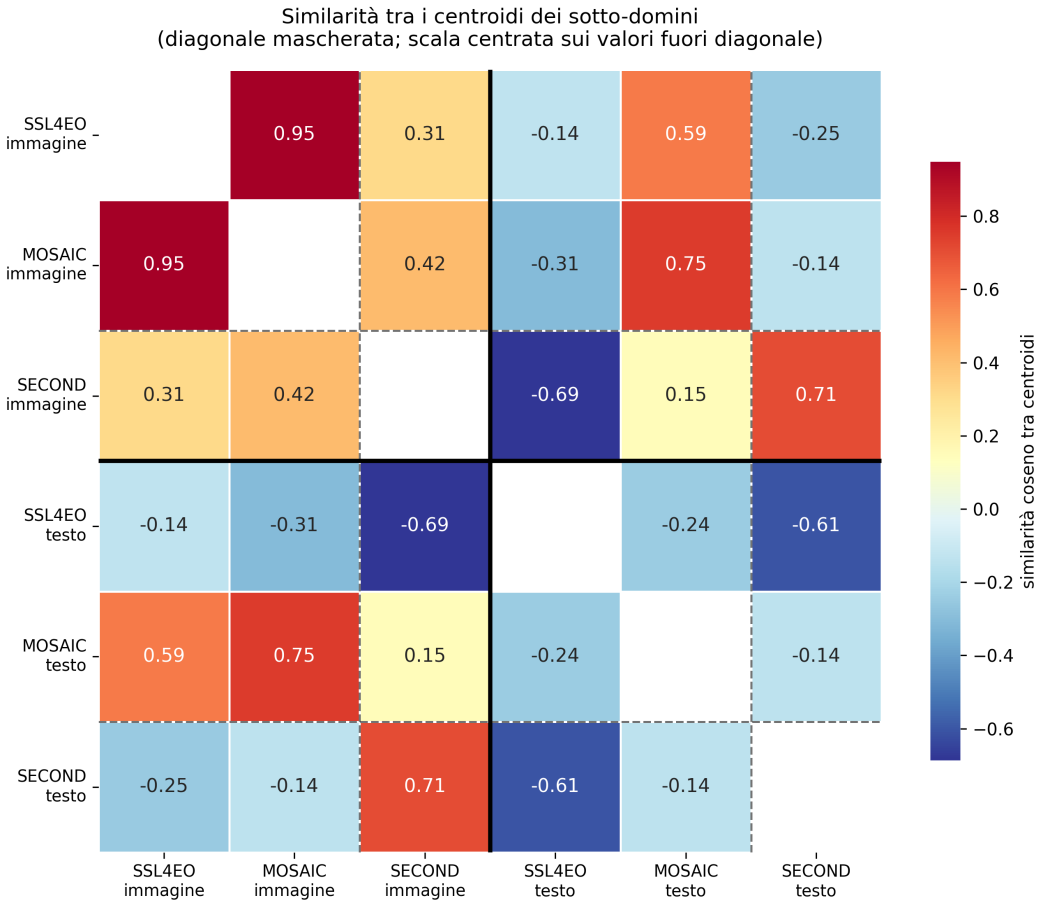


Figura 4.2: Similarità coseno tra i centroidi dei sei sotto-domini principali (immagine e testo, didascalie originali, per SSL4EO-S12, MOSAIC-SEN2-CC e SECOND-CC). La linea continua separa le due modalità, quella tratteggiata i due gruppi di sensore. I centroidi immagine di SSL4EO-S12 e MOSAIC-SEN2-CC coincidono quasi (0,95) e si staccano entrambi da SECOND-CC (0,31 e 0,42). Nei quadranti fuori diagonale l'allineamento tra le due modalità varia fortemente col dominio, dal +0,75 di MOSAIC-SEN2-CC al −0,14 di SSL4EO-S12.

4.8 Analisi Geometrica dello Spazio Condiviso

La separazione tra le nuvole di punti delle due modalità, osservata nella sezione precedente, è uno dei due fenomeni che la letteratura sull'apprendimento contrastivo ha caratterizzato in termini quantitativi (Sezione 2.2.4); l'altro è il collasso dimensionale dello spazio di embedding. Questa sezione applica al sistema completo i corrispondenti strumenti diagnostici e ne deriva due interventi migliorativi, riassumendo una serie sperimentale condotta a valle di quelle finora descritte.

Il modulo di fusione temporale non è il fattore limitante. Il Capitolo 3 ha presentato due realizzazioni del modulo di fusione: il Delta Encoder (Sezione 3.2.3) e il Temporal Attention Encoder (Sezione 3.2.4), quest'ultimo ispirato a L-TAE [29]. A parità di ogni altro componente, le due varianti risultano equivalenti sul retrieval SSL4EO-S12, con un lieve vantaggio del modulo ad attenzione sul trasferimento a MOSAIC-SEN2-CC (25,6% contro 24,1%), ottenuto per giunta con *meno della metà dei parametri* (1,19 M contro 2,62 M). Il Delta Encoder è stato mantenuto come configurazione di riferimento perché adottato in tutti gli esperimenti precedenti, ma il confronto porta un'indicazione più generale: due realizzazioni molto diverse del modulo di fusione, con un costo in parametri che differisce di oltre il doppio, producono prestazioni sostanzialmente equivalenti. Non è dunque qui che si colloca il fattore limitante del sistema.

Il collasso dimensionale è introdotto dalle teste di proiezione. L'analisi centrale di Jing et al. [22] confronta lo spettro dei valori singolari nello spazio delle *rappresentazioni* (uscita degli encoder) con quello nello spazio degli *embedding* (a valle della proiezione), per stabilire in quale punto della rete il collasso venga introdotto. Condotta sui quattro stadi del sistema (Tabella 4.6), essa mostra che il collasso non è ereditato dai backbone pre-addestrati: attraversando le teste di proiezione il rango effettivo [23] cala su entrambi i rami, quasi dimezzandosi su quello visivo. Le rappresentazioni prodotte dagli encoder sono dunque più ricche di quanto lo spazio condiviso riesca a preservare — una constatazione che localizza con precisione un punto su cui intervenire.

Il modality gap è manipolabile a costo nullo. L'analisi di Liang et al. [21] attribuisce la separazione tra le due modalità non all'addestramento ma all'inizializzazione, poiché ogni rete profonda mappa i propri ingressi in un cono ristretto dello spazio. La replica sul sistema completo lo conferma: il divario tra i centroidi vale 1,214 su un modello a pesi casuali, contro 0,275 sul modello addestrato — con una differenza rispetto a quanto riportato dagli autori, ossia che nel nostro regime di scala l'addestramento *riduce* il divario anziché limitarsi a preservarlo. Traslando gli embedding testuali lungo la direzione che congiunge i due centroidi, il retrieval

Stadio	**Dimensioni**	**Rango effettivo**
Immagine — encoder (pre-proiezione)	768	36,1
Immagine — embedding (post-proiezione)	512	**19,3**
Testo — encoder (pre-proiezione)	512	24,2
Testo — embedding (post-proiezione)	512	**18,5**

Tabella 4.6: Collasso dimensionale ai quattro stadi del sistema, secondo il metodo di Jing et al. [22], su 1.500 coppie del validation set SSL4EO-S12. Il rango effettivo è calcolato secondo la formulazione di RankMe [23].

Testa testuale	**Rango eff.**	**Dir. collassate**	**T2I R@10**	**+ gap**	**Zero-shot**
MLP — V2 (riferimento)	18,5	36 / 512	40,9%	41,2%	62,7%
Lineare	27,0	26 / 512	41,3%	42,2%	62,6%
Identità	**31,0**	**1 / 512**	**43,7%**	**44,4%**	**65,5%**

Tabella 4.7: Effetto della capacità della testa di proiezione testuale sulla geometria e sulle prestazioni. L'incertezza sulla Recall@10, pari a circa 1,5 punti (deviazione standard sui 200 sottocampionamenti), colloca il passaggio alla testa lineare entro il margine di misura e quello alla testa identità oltre. Il rango effettivo e le direzioni collassate si riferiscono agli embedding testuali; la colonna "+ gap" aggiunge la chiusura del modality gap a posteriori. L'ordinamento tra le tre configurazioni è stato riprodotto su un secondo seme di inizializzazione.

migliora in modo regolare al ridursi del divario, fino a +0,8 punti percentuali quando lo si annulla completamente. Trattandosi di una semplice traslazione di embedding già calcolati, l'intervento non richiede ri-addestramento ed è stato adottato come passo di post-elaborazione.

Rimuovere la testa di proiezione migliora il retrieval. La diagnosi del collasso è stata verificata nella sua natura causale per via sperimentale: se è la testa a introdurre il collasso, rimuoverla deve rimuoverlo. L'intervento riprende l'idea alla base di DirectCLR [22] ed è applicabile al ramo testuale, dove ingresso e uscita hanno la stessa dimensionalità e la proiezione può essere sostituita dall'identità; sul ramo visivo non lo è, poiché la componente corrispondente svolge anche l'aggregazione temporale e opera un cambio di dimensionalità. La Tabella 4.7 riporta l'esito.

Tutti gli indicatori si muovono nella stessa direzione, in modo monotono, al ridursi della capacità della testa: il rango effettivo degli embedding testuali raddoppia, le direzioni collassate passano da 36 a 1, e ogni metrica di prestazione migliora — inclusa la loss di validazione, e con un numero *inferiore* di parametri. L'effetto si propaga inoltre al ramo visivo, il cui rango effettivo sale da 19,3 a

25,5 pur non avendone modificato la testa, a riprova del fatto che il collasso di una modalità si trasmette all'intero spazio condiviso. Combinati, i due interventi derivati dall'analisi geometrica portano la Recall@10 dal 40,9% al 44,4%, il valore più alto ottenuto in questo lavoro, **senza aggiungere alcun parametro né alcun costo computazionale**.

Il risultato può apparire in contrasto con una prassi consolidata, dato che il paradigma contrastivo prevede quasi sempre una testa di proiezione [16], ma la contraddizione si risolve osservando *quale rappresentazione costituisca il prodotto finale*. Nel self-supervised learning classico, contesto in cui la testa è stata introdotta, l'obiettivo è ottenere un buon *encoder*: la testa viene addestrata insieme al modello ma scartata al termine, e assolve nel frattempo a una funzione protettiva, assorbendo il collasso in una componente destinata a essere eliminata. Nel retrieval cross-modale la situazione si rovescia, poiché è proprio l'embedding a valle della testa il prodotto finale, e il collasso che essa introduce ricade per intero sulla rappresentazione utilizzata. Nel ramo testuale qui considerato viene inoltre meno la ragione strutturale che in CLIP giustifica la proiezione — portare due encoder di dimensionalità diversa in uno spazio comune — poiché l'uscita dell'encoder ha già la dimensione dello spazio condiviso.

Il limite residuo non è geometrico. Alcuni interventi ulteriori, verificati per completezza, delimitano ciò che l'azione sulla geometria può ottenere. Rendere l'allineamento dei centroidi parte esplicita della loss, anziché applicarlo a posteriori, non produce benefici: il divario a fine addestramento resta invariato, coerentemente con la lettura di Liang et al. secondo cui esso costituisce un equilibrio stabile dell'ottimizzazione contrastiva. Il *whitening* degli embedding e la rimozione delle prime componenti principali, due tecniche note per ridistribuire la varianza in spazi collassati, degradano gravemente il retrieval.

Questi esiti convergono su una distinzione che ha valore generale, oltre il caso qui studiato: il collasso *indotto dall'architettura* è un difetto, ed è rimovibile con benefici misurabili; la bassa dimensionalità *intrinseca all'informazione disponibile* non lo è, e i tentativi di correggerla per via geometrica — dalla regolarizzazione VICReg [24] al whitening — falliscono perché spingono verso una geometria che i dati non supportano. La varianza concentrata nelle prime direzioni non è un artefatto da correggere: è il segnale stesso. Ne consegue che il margine di miglioramento ancora disponibile va cercato nella ricchezza informativa delle annotazioni, non nella geometria dello spazio — la stessa conclusione cui conducono, per una via del tutto indipendente, gli esperimenti sullo stile delle didascalie (Sezione 4.4).

4.9 Discussione

La Figura 4.3 riassume il percorso sperimentale. Le configurazioni confrontate occupano un intervallo ristretto di valori assoluti: riportarne lo scostamento dal riferimento, anziché il valore, è ciò che rende leggibile la differenza fra gli interventi.

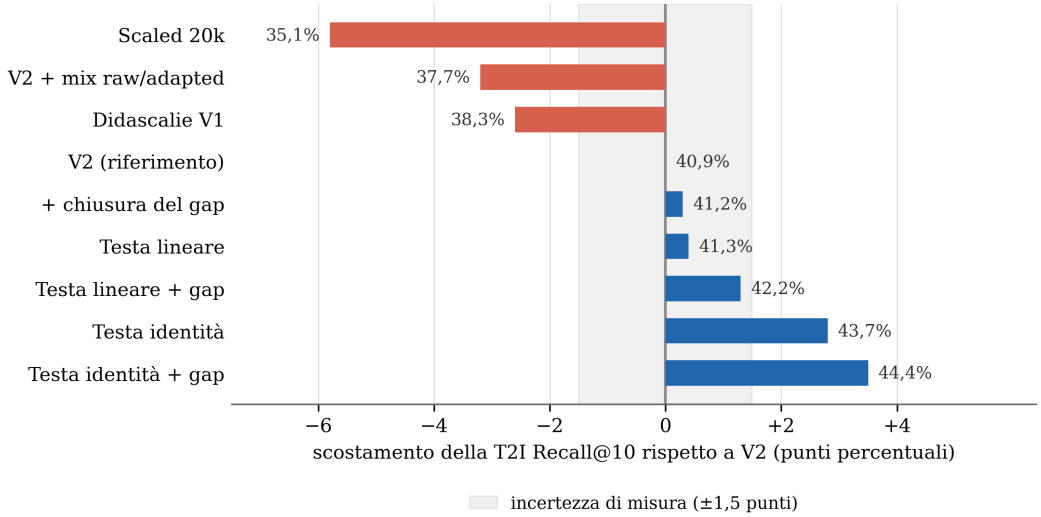


Figura 4.3: Sintesi del percorso sperimentale: scostamento della T2I Recall@10 di ciascuna configurazione rispetto a quella di riferimento (V2, 40,9%). La banda grigia rappresenta l'incertezza di misura (±1,5 punti, deviazione standard sui 200 sottocampionamenti): vi ricadono la chiusura del gap applicata da sola e la testa lineare, mentre la testa identità e la sua combinazione con la chiusura del gap la superano. È esclusa la configurazione priva di SSL4EO-S12 (−38,6 punti), il cui crollo è discusso separatamente in Sezione 4.5.

I risultati presentati in questo capitolo convergono su un principio strutturale ricorrente: la *diversità lessicale delle didascalie* è la risorsa che sostiene il retrieval. Testi più differenziati rendono ogni campione più discriminabile dagli altri nello spazio condiviso, mentre testi uniformi o ancorati a poche categorie semantiche fisse producono collisioni che nessun modello può risolvere, perché le distinzioni fini tra campioni non sono codificate nel dato. Lo si osserva in due esperimenti indipendenti: nel passaggio da V1 a V2 (Sezione 4.4), dove descrivere il percorso temporale anziché il solo saldo elimina didascalie identiche per sequenze diverse; e nella riscrittura tramite LLM, che avvicina il testo umano allo stile generato ma ne sposta soltanto una parte, convergendo per il resto su formulazioni ricorrenti (Sezione 4.7). Il compromesso con l'ancoraggio semantico esplicito — utile invece alla classificazione zero-shot — attraversa tutti gli esperimenti sulle didascalie

riportati nel capitolo.

Un secondo principio, emerso dagli esperimenti di scaling (Sezione 4.6), è che il bilanciamento relativo dei dataset nel batch di addestramento — non solo la loro dimensione nominale — è un fattore critico: espandere un dataset moltiplicandone le righe (come nel caso del multi-caption) ne aumenta il peso implicito nel gradiente complessivo, a scapito degli altri dataset combinati nello stesso training, anche quando la loro dimensione nominale resta invariata.

Un terzo principio, emerso dall'analisi geometrica dello spazio condiviso (Sezione 4.8), riguarda il rapporto tra architettura e rappresentazione. Gli strumenti diagnostici mutuati dalla letteratura hanno permesso di localizzare con precisione un difetto strutturale del sistema — il dimezzamento del rango effettivo operato dalle teste di proiezione — e di verificarne la natura causale per via sperimentale. La sua rimozione ha prodotto il miglioramento più netto ottenuto per via architetturale in tutto il lavoro, per giunta con un numero *inferiore* di parametri e senza alcun costo computazionale aggiuntivo. Il risultato ha inoltre una portata che eccede il caso specifico: mostra che la testa di proiezione, componente pressoché universale nel paradigma contrastivo, è controproducente quando l'embedding a valle — e non l'encoder che la precede — costituisce il prodotto finale del sistema, come accade in ogni applicazione di retrieval cross-modale.

Lo stesso percorso ha reso possibile una seconda distinzione, di carattere generale: quella tra il collasso *indotto dall'architettura* e la bassa dimensionalità *intrinseca all'informazione disponibile*. Il primo è un difetto rimovibile con benefici misurabili; la seconda non lo è, e i tentativi di correggerla per via geometrica falliscono sistematicamente. Poter separare i due casi — anziché trattare ogni spettro compresso come un problema da correggere — è di per sé un risultato utile alla progettazione di sistemi analoghi.

I due assi di miglioramento individuati in questo capitolo — quello che agisce sui *dati* (le didascalie V2, Sezione 4.4) e quello che agisce sull'*architettura* (la rimozione della testa di proiezione) — sono stati misurati separatamente a partire dalla medesima configurazione di riferimento, e producono guadagni di entità comparabile. Poiché intervengono su componenti indipendenti del sistema, la loro composizione rappresenta un'opportunità concreta e immediatamente verificabile (Sezione 5.3).

Resta però il fatto che l'asse dei dati è quello con il margine maggiore, e va detto con chiarezza dove esso sia oggi più debole: nella generazione automatica delle didascalie di SSL4EO-S12, che è al tempo stesso la sorgente testuale più numerosa del training set e quella che più direttamente determina il retrieval in-domain (Sezione 4.5). Il segnale da cui quelle didascalie derivano è per costruzione ristretto: indici di vegetazione e distribuzione delle classi LULC, cioè una descrizione della *composizione* della scena, non di ciò che vi accade. Un vocabolario di poche decine di categorie non può esprimere la causa, la forma o il contesto di un cambiamento, e

per quanto lo si riformuli — è quanto mostra il passaggio da V1 a V2, che ne estrae più informazione senza cambiarne la natura — quel tetto resta. È il principale fattore limitante del sistema, e non è di natura architetturale.

Che questa scelta fosse obbligata, e non subottimale, e quali altri limiti il sistema presenti, sono questioni riprese in Sezione 5.2. Va però osservato fin d'ora che i confronti con lavori esterni sono qui poco informativi, dato che nessuno affronta la combinazione specifica di serie temporali multispettrali e change captioning: le ablation interne — con e senza SSL4EO-S12, il confronto con il caso casuale, il bootstrap a dimensione di pool fissa — restano l'evidenza primaria della validità del metodo proposto.

Su tale evidenza il quadro complessivo è solido. Il sistema apprende un segnale cross-modale reale e sistematicamente distinguibile dal caso, su un compito — il retrieval a grana fine su serie temporali multispettrali descritte in linguaggio naturale — per il quale non esisteva né un metodo di riferimento né un benchmark consolidato, e con un ordine di grandezza in meno di dati rispetto ai lavori CLIP-like. Lungo il percorso, il lavoro ha prodotto una pipeline di generazione automatica delle annotazioni, un protocollo di valutazione costruito ex novo per un compito privo di benchmark di riferimento, un'ablation che stabilisce il ruolo necessario di SSL4EO-S12, e un intervento architetturale che migliora il retrieval riducendo i parametri. Ciascuno di questi elementi è indipendentemente riutilizzabile, e insieme delimitano dove si collochi oggi il margine di miglioramento: non nell'architettura, ormai caratterizzata in dettaglio, ma nella ricchezza delle annotazioni disponibili.

Capitolo 5

Conclusioni

5.1 Riepilogo dei Contributi

Questa tesi ha affrontato il problema del retrieval cross-modale per serie temporali satellitari, estendendo il paradigma contrastivo di CLIP (Sezione 2.2.2) dal dominio delle immagini naturali statiche a quello del telerilevamento multitemporale. I contributi del lavoro, già anticipati in Sezione 1.4, possono essere raggruppati in tre categorie complementari.

Sul piano della **modellazione**, il sistema proposto integra un encoder visivo pre-addestrato per il telerilevamento (CROMA, Sezione 2.3.3) con un modulo di aggregazione temporale dedicato — il Delta Encoder (Sezione 3.2.3) — capace di trattare sequenze di lunghezza variabile calcolando esplicitamente differenze e somme tra frame consecutivi, e un text encoder CLIP, allineati tramite loss contrastiva NT-Xent con temperatura apprendibile. A questo si affianca una strategia di generazione automatica delle didascalie in stile data-to-text, che converte metriche quantitative (NDVI, distribuzione LULC) in descrizioni testuali tramite un LLM, con uno schema frame-by-frame (V2, Sezione 3.4.3) che si è dimostrato superiore, su ogni metrica misurata, allo schema iniziale basato sul solo confronto tra primo e ultimo frame. Lo studio sistematico dell'uso di dataset eterogenei nell'addestramento contrastivo ha inoltre mostrato che SSL4EO-S12 non è un semplice regolarizzatore accessorio, ma una componente necessaria per qualunque capacità di retrieval cross-modale fine-grained (Sezione 4.5): la sua assenza porta il segnale di retrieval in prossimità del caso casuale.

Un secondo filone di contributi riguarda la **geometria dello spazio di embedding condiviso**. Applicando al sistema completo gli strumenti diagnostici proposti in letteratura (Sezione 2.2.4), si è mostrato che il collasso dimensionale non è ereditato dagli encoder pre-addestrati ma introdotto dalle teste di proiezione, e la verifica sperimentale di questa diagnosi — la rimozione della testa di proiezione

testuale, combinata con la chiusura a posteriori del modality gap — ha prodotto il miglioramento più netto ottenuto per via architetturale, portando la Recall@10 dal 40,9% al 44,4% con un numero *inferiore* di parametri (Sezione 4.8). Ne discendono due letture di portata più generale, sviluppate in Sezione 4.9: che la testa di proiezione è controproducente quando è l'embedding a valle a costituire il prodotto finale del sistema, e che il collasso indotto dall'architettura va tenuto distinto dalla bassa dimensionalità intrinseca all'informazione disponibile, perché solo il primo è un difetto rimovibile.

Sul piano **metodologico**, il lavoro ha richiesto di costruire un protocollo di misura adeguato a un compito per il quale non esiste un benchmark di riferimento (Sezione 3.6). Tre scelte lo caratterizzano: un validation set realmente disgiunto dai dati di addestramento e con didascalie rigenerate; l'adozione del bootstrap a dimensione di pool fissa (Sezione 3.6.3) come unica convenzione di misura, che rende i valori confrontabili sia tra modelli sia tra benchmark di dimensione nativa diversa; e il ricorso sistematico al valore atteso sotto ranking casuale come termine di paragone, senza il quale una percentuale grezza non è interpretabile. Il protocollo è stato tenuto rigorosamente fisso tra tutte le configurazioni confrontate, condizione senza la quale nessuna delle ablation riportate nel Capitolo 4 sarebbe stata conclusiva. In un sistema in cui ciascuna fase — preparazione dei dati, generazione delle annotazioni, addestramento, valutazione — può introdurre distorsioni che non producono errori visibili ma solo numeri plausibili, la costruzione della misura non è un dettaglio implementativo: è parte del risultato.

5.2 Limiti

Il sistema proposto presenta limiti che è opportuno dichiarare esplicitamente. Il principale non è architetturale ma riguarda la sorgente del segnale testuale: le didascalie di SSL4EO-S12 descrivono la composizione della scena, non ciò che vi accade, e nessuna riformulazione del prompt può aggiungere l'informazione che quel segnale non contiene (Sezione 4.9). La scelta era tuttavia obbligata: non esistono archivi multi-temporali di grandi dimensioni annotati in linguaggio naturale — i corpora di change captioning disponibili sono bi-temporali e di dimensione limitata (Sezione 2.4), mentre gli archivi estesi come SSL4EO-S12 sono privi di annotazione — e l'annotazione umana di decine di migliaia di serie eccede le risorse di un lavoro di tesi. La generazione automatica a partire da grandezze derivabili dalle immagini era quindi l'unica via percorribile per ottenere supervisione testuale su scala adeguata, e i risultati ottenuti ne misurano tanto l'efficacia quanto il limite.

A ciò si aggiunge la scala del training set, dell'ordine di alcune decine di migliaia di coppie immagine–testo, modesta rispetto al pre-addestramento CLIP-style e alle varianti specializzate per il telerilevamento (Sezione 2.3.5): i valori assoluti

di retrieval ottenuti, per quanto significativamente superiori al caso casuale, ne risentono, e un confronto diretto con quei lavori andrebbe interpretato con cautela. In terzo luogo, il benchmark LEVIR-CC (Sezione 2.4.5) resta sostanzialmente fuori portata in condizioni di puro zero-shot, un limite atteso dato il fortissimo scarto di dominio (risoluzione, natura del segnale, stile testuale) rispetto ai dati di addestramento, ma che segnala comunque i limiti attuali della generalizzazione del sistema a domini radicalmente diversi. Infine, gli esperimenti di scaling (Sezione 4.6) hanno rivelato un trade-off non ancora risolto tra le prestazioni in-domain (SSL4EO-S12) e quelle sui benchmark downstream quando la composizione del batch di addestramento viene alterata, a indicare che l'attuale strategia di combinazione dei dataset (Sezione 3.5.1) non è ancora pienamente ottimizzata.

5.3 Lavori Futuri

Diverse direzioni di lavoro futuro emergono direttamente dai risultati e dai limiti discussi in questa tesi. La più immediata, motivata dagli esperimenti di scaling (Sezione 4.6), è l'implementazione di un *sampler bilanciato per dataset*, con pesi espliciti e indipendenti dal numero di righe di ciascuna sorgente: un tale meccanismo permetterebbe di disaccoppiare "quante didascalie ha ogni immagine" (utile alla diversità testuale) da "quanto peso ha ciascun dataset nel batch" (che oggi scala automaticamente, e indesiderabilmente, con quel conteggio), consentendo verosimilmente di ottenere il beneficio del multi-caption senza il costo collaterale osservato sul segnale SSL4EO-S12. Una seconda direzione naturale è la combinazione delle due tecniche che agiscono sui testi dei due gruppi di dataset — le didascalie V2 frame-by-frame per SSL4EO-S12 e l'augmentation tramite riscrittura per i benchmark downstream — con un simile sampler bilanciato, per verificare se i due miglioramenti possano finalmente sommarsi anziché confliggere. Una terza direzione è la composizione dei due assi di miglioramento rimasti sperimentalmente distinti (Sezione 4.9), verificabile con un singolo addestramento aggiuntivo poiché intervengono su componenti indipendenti del sistema. A essa si affianca l'estensione al ramo visivo dell'intervento sulla testa di proiezione, oggi preclusa dal fatto che il modulo di fusione temporale svolge simultaneamente aggregazione e proiezione: separare le due funzioni permetterebbe di applicare anche a questo ramo il trattamento rivelatosi efficace su quello testuale. Sul piano della loss contrastiva, un'esplorazione più sistematica di tecniche di *hard-negative mining* — la selezione mirata dei negativi più difficili anziché il ricorso ai soli negativi in-batch (Sezione 2.2.3) — potrebbe rafforzare il segnale di apprendimento su un dominio come SSL4EO-S12, dove l'uniformità delle scene rende gran parte dei negativi in-batch facili da distinguere e quindi poco informativi.

Bibliografia

[1] M. Drusch et al. «Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services». In: *Remote Sensing of Environment* 120 (2012), pp. 25–36 (cit. alle pp. 1, 15).

[2] Chenyang Liu, Rui Zhao, Hao Chen, Zhengxia Zou e Zhenwei Shi. «Remote Sensing Image Change Captioning With Dual-Branch Transformers: A New Method and a Large Scale Dataset». In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), p. 5633520 (cit. alle pp. 1, 2, 21–24).

[3] Daniele Rege Cambrin, Lorenzo Vaiani, Giuseppe Gallipoli, Luca Cagliero e Paolo Garza. «CLOSP: A Unified Semantic Space for SAR, MSI, and Text in Remote Sensing». In: *arXiv preprint arXiv:2507.10403* (2025) (cit. alle pp. 2, 20).

[4] Alec Radford et al. «Learning Transferable Visual Models From Natural Language Supervision». In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*. 2021. arXiv: `2103.00020 [cs.CV]` (cit. alle pp. 3, 12, 13, 24).

[5] Yann LeCun, Yoshua Bengio e Geoffrey Hinton. «Deep Learning». In: *Nature* 521.7553 (2015), pp. 436–444 (cit. alle pp. 6, 7).

[6] Diederik P. Kingma e Jimmy Ba. «Adam: A Method for Stochastic Optimization». In: *arXiv preprint arXiv:1412.6980* (2014) (cit. alle pp. 7, 37).

[7] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever e Ruslan Salakhutdinov. «Dropout: A Simple Way to Prevent Neural Networks from Overfitting». In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958 (cit. a p. 7).

[8] Yann LeCun, Léon Bottou, Yoshua Bengio e Patrick Haffner. «Gradient-Based Learning Applied to Document Recognition». In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cit. a p. 8).

[9] Alex Krizhevsky, Ilya Sutskever e Geoffrey E. Hinton. «ImageNet Classification with Deep Convolutional Neural Networks». In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2012 (cit. a p. 8).

[10] Karen Simonyan e Andrew Zisserman. «Very Deep Convolutional Networks for Large-Scale Image Recognition». In: *arXiv preprint arXiv:1409.1556* (2014) (cit. a p. 8).

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren e Jian Sun. «Deep Residual Learning for Image Recognition». In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. a p. 8).

[12] Sepp Hochreiter e Jürgen Schmidhuber. «Long Short-Term Memory». In: *Neural Computation* 9.8 (1997), pp. 1735–1780 (cit. a p. 8).

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser e Illia Polosukhin. «Attention Is All You Need». In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2017 (cit. a p. 8).

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee e Kristina Toutanova. «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding». In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. 2019 (cit. a p. 9).

[15] Alexey Dosovitskiy et al. «An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale». In: *International Conference on Learning Representations (ICLR)*. 2021 (cit. a p. 9).

[16] Ting Chen, Simon Kornblith, Mohammad Norouzi e Geoffrey Hinton. «A Simple Framework for Contrastive Learning of Visual Representations». In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*. 2020 (cit. alle pp. 11, 13, 54).

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie e Ross Girshick. «Momentum Contrast for Unsupervised Visual Representation Learning». In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020 (cit. alle pp. 11, 33).

[18] Chao Jia et al. «Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision». In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*. 2021 (cit. a p. 12).

[19] Aaron van den Oord, Yazhe Li e Oriol Vinyals. «Representation Learning with Contrastive Predictive Coding». In: *arXiv preprint arXiv:1807.03748* (2018) (cit. a p. 13).

[20] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros e Sanja Fidler. «VSE++: Improving Visual-Semantic Embeddings with Hard Negatives». In: *British Machine Vision Conference (BMVC)*. 2018 (cit. alle pp. 14, 24, 40).

[21] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung e James Zou. «Mind the Gap: Understanding the Modality Gap in Multi-modal Contrastive Representation Learning». In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2022 (cit. alle pp. 14, 52).

[22] Li Jing, Pascal Vincent, Yann LeCun e Yuandong Tian. «Understanding Dimensional Collapse in Contrastive Self-supervised Learning». In: *International Conference on Learning Representations (ICLR)*. 2022 (cit. alle pp. 15, 52, 53).

[23] Quentin Garrido, Randall Balestriero, Laurent Najman e Yann Lecun. «RankMe: Assessing the Downstream Performance of Pretrained Self-Supervised Representations by Their Rank». In: *International Conference on Machine Learning (ICML)*. 2023 (cit. alle pp. 15, 52, 53).

[24] Adrien Bardes, Jean Ponce e Yann LeCun. «VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning». In: *International Conference on Learning Representations (ICLR)*. 2022 (cit. alle pp. 15, 54).

[25] Compton J. Tucker. «Red and Photographic Infrared Linear Combinations for Monitoring Vegetation». In: *Remote Sensing of Environment* 8.2 (1979), pp. 127–150 (cit. a p. 15).

[26] Benedikt Blumenstiel, Nassim Ait Ali Braham, Conrad M. Albrecht, Stefano Maurogiovanni e Paolo Fraccaro. «SSL4EO-S12 v1.1: A Multimodal, Multiseasonal Dataset for Pretraining, Updated». In: *arXiv preprint arXiv:2503.00168* (2025) (cit. alle pp. 16, 17).

[27] Anthony Fuller, Koreen Millard e James R. Green. «CROMA: Remote Sensing Representations with Contrastive Radar-Optical Masked Autoencoders». In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2023. arXiv: `2311.00566 [cs.CV]` (cit. alle pp. 16, 18).

[28] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár e Ross Girshick. «Masked Autoencoders Are Scalable Vision Learners». In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022 (cit. a p. 16).

[29] Vivien Sainte Fare Garnot e Loic Landrieu. «Lightweight Temporal Self-Attention for Classifying Satellite Images Time Series». In: *Advanced Analytics and Learning on Temporal Data (AALTD), ECML/PKDD Workshops*. 2020 (cit. alle pp. 18, 31, 52).

[30] Vivien Sainte Fare Garnot, Loic Landrieu, Sebastien Giordano e Nesrine Chehata. «Satellite Image Time Series Classification with Pixel-Set Encoders and Temporal Self-Attention». In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020 (cit. a p. 18).

[31] Fan Liu, Delong Chen, Zhangqingyun Guan, Xiaocong Zhou, Jiale Zhu, Qiaolin Ye, Liyong Fu e Jun Zhou. «RemoteCLIP: A Vision Language Foundation Model for Remote Sensing». In: *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024) (cit. alle pp. 19, 32).

[32] Zilun Zhang, Tiancheng Zhao, Yulong Guo e Jianwei Yin. «RS5M and GeoRSCLIP: A Large-Scale Vision-Language Dataset and a Large Vision-Language Model for Remote Sensing». In: *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024) (cit. a p. 19).

[33] Ali Can Karaca, M. Enes Özelbaş, Saadettin Berber, Orkhan Karimli, Turabi Yildirim e M. Fatih Amasyali. «Robust Change Captioning in Remote Sensing: SECOND-CC Dataset and MModalCC Framework». In: *arXiv preprint arXiv:2501.10075* (2025) (cit. alle pp. 22–24).

[34] Busra Tuzlupinar, Enes Ozelbas, Mehmet Fatih Amasyali e Ali Can Karaca. «Introducing MOSAIC-SEN2-CC: A Multispectral Dataset and Adaptation Framework for Remote Sensing Change Captioning». In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 18 (2025) (cit. alle pp. 22, 23).

[35] An Yang, Baosong Yang, Beichen Zhang et al. «Qwen2.5 Technical Report». In: *arXiv preprint arXiv:2412.15115* (2024) (cit. a p. 35).

[36] Andrej Karpathy e Li Fei-Fei. «Deep Visual-Semantic Alignments for Generating Image Descriptions». In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 (cit. a p. 40).

[37] Leland McInnes, John Healy e James Melville. «UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction». In: *arXiv preprint arXiv:1802.03426* (2018) (cit. a p. 48).