



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Magistrale in Informatica

TESI DI LAUREA

**Performance Sustainability Trade-Off:
Uno Studio Empirico
sull'Ottimizzazione di Modelli di Tiny
Machine Learning**

RELATORE

Prof. Fabio Palomba

Dott. Vincenzo De Martino

Università degli Studi di Salerno

CANDIDATO

Domenico Antonio Gioia

Matricola: 0522501518

Anno Accademico 2023-2024

Questa tesi è stata realizzata nel

sesa^{lab}
SOFTWARE ENGINEERING
SALERNO

A quello che ero, che mi ha insegnato a sognare.
A quello che sono, che mi ha mostrato la forza di non arrendermi.
A quello che sarò, che spero continui a cercare, imparare e crescere.

Abstract

In un periodo in cui la sostenibilità energetica e l'efficienza delle prestazioni dei modelli di Machine Learning stanno diventando temi sempre più ricorrenti, questa tesi si pone l'obiettivo di esplorare il trade-off tra prestazioni, accuratezza e consumo energetico nei modelli di Tiny Machine Learning. L'obiettivo principale di questo lavoro è stato di valutare l'efficacia delle tecniche di ridimensionamento dei modelli di machine learning rispetto ai consumi energetici e le prestazioni. Attraverso una systematic literature review, sono stati individuati i principali approcci per ridurre la dimensione dei modelli di machine learning e i task a cui queste tecniche sono principalmente applicate. Successivamente, uno studio di benchmark è stato condotto su modelli di Computer Vision, quali AlexNet, ResNet18 e VGG16, per stimare come queste tecniche influenzino non solo l'accuratezza e le performance, ma anche il consumo energetico. I risultati hanno evidenziato che la quantizzazione è particolarmente efficace nel mantenere prestazioni simili ai modelli originali, nel ridurre drasticamente la dimensione dei modelli consentendo l'utilizzo in dispositivi con risorse di archiviazione limitate, e nel ridurre il consumo energetico. La low rank approximation offre una riduzione del consumo energetico ma con un minimo impatto sulle prestazioni, mentre il pruning ha mostrato risultati variabili, con alcune tecniche che portano a un calo significativo delle prestazioni.

Indice

Elenco delle Figure	iv
Elenco delle Tabelle	v
1 Introduzione	1
1.1 Motivazioni ed Obiettivi	3
1.2 Risultati Ottenuti	3
1.3 Struttura della Tesi	4
2 Stato dell'arte	6
2.1 Sostenibilità	6
2.2 Tiny Machine Learning	7
2.2.1 Sviluppo di un modello Tiny Machine Learning	9
2.2.2 Ottimizzazione dei Modelli di Machine Learning	10
2.3 Limitazioni e Obiettivi	11
3 Tecniche di Minimizzazione dei Modelli di Machine Learning: Una Systematic Literature Review	12
3.1 Systematic Literature Review	13
3.1.1 Research Questions	13
3.1.2 Definizione della Query di Ricerca	14

3.1.3	Database di ricerca	15
3.1.4	Criteri di inclusione ed esclusione	16
3.1.5	Metodo di Ricerca per condurre il processo di Snowballing . .	16
3.1.6	Metodo di Ricerca per condurre la Valutazione della Qualità .	17
3.1.7	Form di Estrazione Dati	19
3.1.8	Esecuzione della Ricerca	20
3.2	Analisi dei risultati	22
3.2.1	Analisi Bibliometrica	22
3.2.2	RQ1: Quali sono le tecniche usate per ridurre la dimensione di un modello di Machine Learning?	23
3.2.3	RQ2: Per quali task sono usate tecniche di ridimensionamento di modelli di Machine Learning?	27
3.2.4	RQ3. In quali fasi dello sviluppo le tecniche di ridimensiona- mento di modelli di Machine Learning sono implementate? .	28
3.2.5	RQ4. Quali requisiti non-funzionali le tecniche di ridimensio- namento di modelli di Machine Learning mirano ad ottimizzare?	31
3.2.6	RQ5. In che modo le tecniche di ridimensionamento di modelli di Machine Learning sono valutate?	32
4	Studio di Benchmark: Computer Vision	34
4.1	Studio di Benchmark	35
4.1.1	Research Question	35
4.2	Oggetti dello Studio	36
4.3	Soggetti dello Studio	36
4.4	Variabili dello Studio	37
4.5	Ipotesi Sperimentali, Esecuzione e Analisi	40
4.5.1	Ipotesi Sperimentali	40
4.5.2	Esecuzione dell'Esperimento	41
4.6	Analisi dei Dati	41
4.7	Analisi dei Risultati	42
4.7.1	Analisi del Modello Alexnet	42
4.7.2	Performance Sustainability Trade-Off per AlexNet	49

4.7.3	Analisi del Modello Resnet18	51
4.7.4	Performance Sustainability Trade-Off per ResNet18	57
4.7.5	Analisi del Modello VGG16	59
4.7.6	Performance Sustainability Trade-Off per VGG16	66
4.8	RQ6: In che modo le tecniche di ottimizzazione trovate impattano i modelli ML sugli attributi di qualità?	67
5	Minacce alla Validità	72
5.1	Minacce alla Validità Interna	72
5.2	Minacce alla Validità Esterna	72
5.3	Minacce alla Validità della Selezione degli Studi	73
5.4	Minacce alla Validità di Costrutto	73
5.5	Minacce alla Validità della Conclusione	73
6	Conclusioni	74
	Bibliografia	77
	Articoli Systematic Literature Review	79

Elenco delle figure

3.1	Processo per la Selezione degli articoli	20
3.2	Qualità articoli selezionati	22
3.3	Anno pubblicazione articoli selezionati	23
4.1	Test Shapiro-Wilk per Alexnet	47
4.2	Test T per Alexnet	48
4.3	Grafico dei Consumi Energetici di AlexNet Pre e Pruning	48
4.4	Grafico dei Consumi Energetici di AlexNet Pre, Quantizzazione e low rank Approximation	49
4.5	Test Shapiro-Wilk per ResNet18	56
4.6	Test T per ResNet18	56
4.7	Grafico dei Consumi Energetici di ResNet18 Pre e Pruning	57
4.8	Grafico dei Consumi Energetici di ResNet18 Pre, Quantizzazione e low rank Approximation	57
4.9	Test Shapiro-Wilk per VGG16	64
4.10	Test T e Test di Mann-Whitney U per VGG16	65
4.11	Grafico dei Consumi Energetici di VGG16 Pre e Pruning	65
4.12	Grafico dei Consumi Energetici di VGG16 Pre, Quantizzazione e low rank Approximation	66

Elenco delle tabelle

3.1	Form Estrazione Dati	19
3.2	Tecniche di Minimizzazione della Dimensione	24
3.3	Tecniche di Minimizzazione della Dimensione	25
3.4	Tecniche di Minimizzazione della Dimensione	26
3.5	Fasi Implementazione Tecniche	29
4.2	Variabili dello Studio	38
4.4	Performance AlexNet TB	43
4.6	Performance AlexNet TL	43
4.8	Performance TQ Alex net	44
4.10	Performance TPCNS Alexnet	44
4.12	Performance TPGNS Alex net	45
4.14	Performance TPNS Alexnet	45
4.16	Performance TPSPC Alexnet	46
4.18	Performance TB ResNet18	52
4.20	Performance TL ResNet18	52
4.22	Performance TQ ResNet18	53
4.24	Performance TPCNS ResNet18	53
4.26	Performance TPGNS ResNet18	54
4.28	Performance TPNS ResNet18	54

4.30	Performance TPSPC ResNet18	55
4.32	Performance TB VGG16	60
4.34	Performance TL VGG16	60
4.36	Performance TQ VGG16	61
4.38	Performance TPCNS VGG16	61
4.40	Performance TPGNS VGG16	62
4.42	Performance TPNS VGG16	62
4.44	Performance TPSPC VGG16	63

CAPITOLO 1

Introduzione

Negli ultimi anni, il **Machine Learning** ha rivoluzionato numerosi settori, permettendo di sviluppare sistemi capaci di apprendere dai dati e migliorare le proprie prestazioni nel tempo. Esistono moltissime applicazioni, che spaziano dal riconoscimento vocale, alla visione artificiale, fino alla diagnostica medica. Molti di questi applicativi si basano fortemente su modelli di machine learning complessi, in particolare le reti neurali profonde (Deep Learning), che richiedono enormi risorse computazionali per gestire l'addestramento e l'inferenza. Tuttavia, questa crescente adozione del machine learning comporta un aumento importante del consumo energetico, creando nuove sfide in termini di sostenibilità ambientale. Man mano che la diffusione del machine learning si espande, cresce anche la consapevolezza dell'impatto ambientale associato all'esecuzione di modelli di apprendimento automatico, specialmente su larga scala. Infatti, le moderne architetture di rete neurale richiedono grandi quantità di energia per processare i dati, alimentare le GPU e memorizzare i pesi dei modelli. Questa esigenza energetica si scontra con la necessità di ridurre l'impronta ecologica dell'intelligenza artificiale, in linea con i crescenti obiettivi globali di sostenibilità. Negli ultimi anni, il campo del **Tiny Machine Learning** (TinyML) ha guadagnato notevole attenzione grazie alla sua capacità di implementare algoritmi di apprendimento automatico su dispositivi con risorse limitate come microcontrollori e

sensori IoT (Internet of Things). Questi dispositivi, caratterizzati da limitata potenza computazionale e bassa disponibilità di memoria, rappresentano una sfida significativa per l'esecuzione di modelli di machine learning che presentano un'architettura molto articolata e richiedono un elevato numero di parametri per poter funzionare correttamente. Tali modelli, come ad esempio le reti neurali profonde, sono capaci di elaborare informazioni in modo sofisticato, permettendo di raggiungere alte prestazioni su compiti complessi, come il processing del linguaggio naturale o il riconoscimento di immagini. Questi modelli complessi, però, necessitano di risorse computazionali significative per essere utilizzati durante l'inferenza, come ad esempio una notevole quantità di memoria e una grande capacità di calcolo. Quindi, l'adozione su larga scala di TinyML è spesso ostacolata da problematiche legate alle **performance** e alla **sostenibilità energetica** del modello. Questa rappresenta una sfida significativa per l'esecuzione di modelli TinyML.

In un'epoca in cui la sostenibilità ambientale sta assumendo un ruolo sempre più centrale, ridurre il consumo energetico di questi dispositivi diventa cruciale. In questo contesto, si rende necessario esplorare tecniche avanzate di ottimizzazione che non solo permettano di minimizzare le dimensioni dei modelli di machine learning, ma che mantengano anche un equilibrio accettabile tra performance e sostenibilità energetica.

Negli ultimi anni, la ricerca delle tecniche di ottimizzazione ha fatto notevoli progressi, ed infatti sono stati sviluppati molti approcci per ridurre la dimensione dei modelli senza compromettere eccessivamente le prestazioni. Tuttavia, nonostante la disponibilità di tecniche, esistono ancora limitazioni. Ad esempio, è ancora poco chiaro quale sia il reale vantaggio di queste tecniche in termini di trade-off tra performance, accuratezza e sostenibilità energetica. Molti studi mancano di una valutazione sistematica che misuri con precisione l'impatto di tali tecniche su aspetti come il consumo energetico, la velocità di inferenza, l'accuratezza e l'uso di risorse hardware. Questa tesi si propone di indagare questo trade-off nel contesto del TinyML. Esploreremo diverse tecniche di ottimizzazione della dimensione dei modelli di machine learning, con l'obiettivo di valutare in modo sistematico il loro impatto sulle prestazioni del modello e sull'efficienza energetica.

1.1 Motivazioni ed Obiettivi

Nonostante i progressi della ricerca nel campo del TinyML e delle tecniche di ottimizzazione, permangono significative limitazioni. In particolare, è ancora poco chiaro quale sia il trade-off ottimale tra performance, accuratezza e sostenibilità energetica. Per questa ragione, questo lavoro si concentra su approcci che mirano a ridurre la **complessità computazione** e la **dimensione dei modelli** di machine learning. L'obiettivo è valutare l'impatto di queste tecniche su variabili chiave come l'accuratezza, il throughput e il consumo energetico, per identificare un equilibrio ottimale tra performance e sostenibilità.

Alla luce di queste considerazioni, questo lavoro di tesi si propone di rispondere a due principali obiettivi di ricerca:

- Individuazione degli approcci e delle tecniche di minimizzazione della dimensione dei modelli di machine learning tramite una **systematic literature review**, analizzando studi presenti in letteratura.
- Studio di benchmark su un insieme selezionato di modelli, per valutare come l'uso di tecniche di minimizzazione della dimensione di modelli di machine learning influenzi l'accuratezza, le prestazioni e il consumo energetico, identificando un trade-off tra performance e sostenibilità.

Lo studio si focalizzerà principalmente su modelli di **Computer Vision** poiché sono i più coinvolti in questo ambito di discussione. Verranno utilizzate tecniche come la **quantizzazione**, il **pruning** e la **low-rank approximation** per ottimizzare modelli come **AlexNet**, **ResNet18** e **VGG16**, e si misureranno gli effetti di queste tecniche sulle prestazioni e sul consumo energetico.

1.2 Risultati Ottenuti

I risultati di questa ricerca hanno evidenziato come le tecniche di riduzione della dimensione dei modelli di machine learning, in particolare la quantizzazione, la low rank approximation e il pruning, influenzino positivamente il trade off tra prestazioni e sostenibilità energetica. La quantizzazione si rileva come tecnica più

efficace, permettendo una significativa riduzione del consumo energetico e della dimensione del modello, mantenendo livelli di accuratezza e di performance molto simili ai modelli originali. La low rank approximation ha offerto una buona riduzione energetica con un impatto minimo sulle prestazioni e accuratezza. Il pruning, invece, ha mostrato risultati misti a seconda della specializzazione utilizzata. Quello globale e strutturato per canali ha mostrato risultati variabili, mentre quello casuale non strutturato i peggiori. Il pruning non strutturato ha mostrato peggioramenti in termini di accuratezza e performance, mentre i consumi energetici restano invariati rispetto al modello baseline.

1.3 Struttura della Tesi

La tesi è organizzata come segue:

- **Capitolo 1: Introduzione** - Fornisce una panoramica del contesto relativo al machine learning, con focus specifico sulle problematiche legate alle performance e al consumo energetico nei modelli TinyML. Vengono introdotti gli obiettivi principali della tesi e le motivazioni alla base dello studio.
- **Capitolo 2: Stato dell'Arte** - Analizza lo stato dell'arte sull'emergente campo del Tiny Machine Learning e sulla sostenibilità. Vengono esplorate le sfide legate all'impatto ambientale dei modelli di machine learning, evidenziando la necessità di approcci più sostenibili per ottimizzare le risorse energetiche. Inoltre, vengono esaminati i progressi recenti del TinyML.
- **Capitolo 3: Tecniche di Minimizzazione dei Modelli di Machine Learning: Una Systematic Literature Review** - Descrive il processo di systematic literature review volto a identificare le tecniche, i task e le metriche più frequentemente utilizzati nella minimizzazione delle dimensioni dei modelli di machine learning. Viene fornita una sintesi delle evidenze emerse dalla letteratura.
- **Capitolo 4: Studio di Benchmark: Computer Vision** - Illustra lo studio condotto, in cui vengono valutate le prestazioni e la sostenibilità energetica dei

modelli ottimizzati attraverso diverse tecniche. Viene descritto il processo sperimentale, i modelli e i dataset utilizzati, nonché le metriche chiave utilizzate. Sono esaminati i risultati ottenuti dallo studio e vengono indicati i trade off tra performance e sostenibilità tra le tecniche di minimizzazione della dimensione dei modelli di machine learning.

- **Capitolo 5: Minacce alla Validità** - Esamina le potenziali minacce alla validità dello studio, discutendo le minacce alla validità della conclusione, interna, del costruito, esterna e della selezione degli studi.
- **Capitolo 6: Conclusioni** - Riassume i principali risultati e gli obiettivi raggiunti. Infine, vengono delineati i limiti dello studio e vengono proposti possibili sviluppi futuri.

CAPITOLO 2

Stato dell'arte

2.1 Sostenibilità

La sostenibilità è diventata un concetto chiave nelle scienze computazionali, particolarmente rilevante nel contesto del machine learning. In generale, il termine *sostenibilità* fa riferimento alla capacità di soddisfare i bisogni presenti senza compromettere la capacità delle future generazioni di soddisfare i propri bisogni [1]. Nel contesto della computazione, il concetto di sostenibilità è diventato sempre più rilevante, soprattutto nel campo del machine learning. La sostenibilità, infatti, implica non solo l'ottimizzazione delle performance dei modelli, ma anche la riduzione del consumo di risorse energetiche durante l'addestramento e l'esecuzione di questi modelli. Come osservato da Patterson et al [2], la crescente complessità dei modelli ha portato a un aumento significativo delle risorse richieste, sollevando preoccupazioni riguardo alla sostenibilità nel lungo termine. Pertanto, è necessario adottare approcci più sostenibili per ridurre l'impronta ecologica. Schwartz et al. [3] hanno esaminato l'impatto energetico dei modelli di deep learning, dimostrando che esistono significative opportunità per ridurre il footprint computazionale attraverso l'ottimizzazione del modello. Basti pensare che l'addestramento di grandi modelli di linguaggio naturale, come descrive Strubell et al.[4], può comportare un consumo energetico

pari a quello di una persona durante un ciclo di vita. Questo solleva preoccupazioni significative riguardo l'impatto ambientale delle tecnologie di machine learning, soprattutto perché ci si sposta gradualmente sempre di più verso l'utilizzo di modelli complessi e di grandi dimensioni. La sempre più ampia adozione di questi modelli di machine learning negli ambiti più disparati ha portato all'aumento dei consumi energetici e delle emissioni di CO₂-equivalente generate. A tal proposito, sono aumentati gli studi sul Green AI, come studiato da Georgiou et al.[5] e Verdecchia et al.[6]. Il primo articolo fornisce una panoramica dei lavori esistenti relativi al Green AI e la costante crescita del numero di articoli scientifici pubblicati negli anni. Il secondo studio, invece, si concentra principalmente sul confronto dei consumi energetici su differenti modelli di riferimento creati con differenti framework. L'ottimizzazione energy-aware è al centro di molte ricerche recenti. Ad esempio, il lavoro di Liao et al.[7] esplora come la sintonizzazione dei parametri e l'ottimizzazione dei modelli impattano su proprietà di performance come la latenza di inferenza e il consumo energetico, oltre l'accuratezza del modello. In modo simile, il lavoro di Hampau et al. [8] valuta empiricamente l'impatto di tre strategie di containerizzazione sull'uso di energia, tempo di esecuzione, uso della CPU e della memoria per il task di computer vision su dispositivi edge, suggerendo che WebAssembly e ONNX Runtime siano soluzioni energeticamente efficienti per ambienti a risorse limitate. Per quanto riguarda l'ambito del test delle prestazioni su rete adattive (AdNN) su dispositivi con risorse limitate, è stato progettato *DeepPerform* da Chen et al. [9], un tool per generare campioni di test che rilevano inefficienze computazioni, migliorando significativamente la capacità di trovare degradazioni delle prestazioni rispetto ai metodi tradizionali. Si rende necessario, ricercare metodi e ottimizzazioni per rendere i modelli più efficienti sia dal punto di vista computazionale che energetico.

2.2 Tiny Machine Learning

Il Machine Learning è attualmente impiegato in maniera sempre più diffusa sia nella teoria che negli esperimenti [10]. Negli ultimi anni a causa dell'accresciuto utilizzo di dispositivi embedded e l'espansione dell'Internet of Things, il campo del Tiny Machine Learning ha progressivamente guadagnato popolarità e interesse. L'av-

vento di tali tecnologie ha segnato una svolta nel settore dell'intelligenza artificiale per l'implementazione di sistemi intelligenti su dispositivi con risorse computazionali ridotte. Questi dispositivi, quali sensori, microcontrollori e dispositivi portatili, sono in grado di eseguire algoritmi di machine learning direttamente sulle proprie piattaforme, senza connessioni costanti a server remoti. L'applicabilità in tempo reale del Tiny Machine Learning ha trovato applicazione in una vasta gamma di settori, tra cui l'industria, l'agricoltura, la sicurezza, l'healthcare e molti altri [11]. Il Tiny Machine Learning si caratterizza per la sua semplicità hardware, operando con un consumo energetico inferiore a 1mW [12], e tale caratteristica consente ai dispositivi di funzionare anche con batterie a bottone standard, la cui durata può variare da pochi mesi a un anno. L'obiettivo fondamentale del Tiny machine learning è di massimizzare le prestazioni dell'apprendimento automatico ottimizzando l'hardware, il software e le discipline legate alla gestione dei dati.

I principali vantaggi del Tiny Machine Learning sono:

- **Latenza ridotta:** Poiché i modelli operano direttamente sui dispositivi edge, non è necessario trasferire i dati a un server per l'inferenza, riducendo la latenza e garantendo risposte più rapide.
- **Risparmio energetico:** I microcontrollori richiedono una quantità di energia molto ridotta, consentendo loro di funzionare per lunghi periodi senza bisogno di essere caricati. Inoltre, l'assenza di un'infrastruttura server estesa porta a un significativo risparmio di energia, costi e risorse.
- **Privacy dei dati:** Poiché i modelli operano direttamente sui dispositivi edge, i dati non vengono trasferiti e conservati sui server, aumentando la sicurezza e la riservatezza dei dati.
- **Larghezza di banda ridotta:** Per l'inferenza, è necessaria poca o nessuna connettività Internet. I dispositivi edge acquisiscono i dati e li elaborano direttamente, riducendo la necessità di trasferire costantemente dati grezzi dai sensori ai server.

I requisiti del flusso di lavoro per le applicazioni Tiny Machine Learning presentano somiglianze con quelli dei tradizionali flussi di lavoro di machine learning. Tuttavia,

ciò che distingue il Tiny Machine Learning è la capacità di eseguire diverse funzioni su dispositivi di dimensioni ridotte. Un esempio di framework largamente utilizzato per l'apprendimento automatico su dispositivi edge è *Tensorflow Lite*¹ per microcontrollori che è stato appositamente progettato per l'implementazione di modelli di machine learning su sistemi embedded caratterizzati da limitate risorse di memoria. Un altro framework ampiamente utilizzato è *Pytorch*², che offre strumenti per ottimizzare ed eseguire modelli di machine learning su dispositivi mobili ed edge.

L'espansione dell'Internet of Things e l'aumento dell'utilizzo di dispositivi embedded hanno aperto nuove prospettive nel campo del Tiny Machine Learning, portando una crescente consapevolezza dell'importanza di ottimizzare modelli di machine learning per adattarli a risorse computazionali ridotte. Questa crescente adozione è stata alimentata dall'esigenza di eseguire algoritmi di machine learning direttamente sui dispositivi stessi, senza dipendenza da connessioni costanti a server remoti. Tuttavia, le limitate capacità dei dispositivi embedded rappresentano una sfida significativa che richiede approcci innovativi per garantire prestazioni ottimali dei modelli. Pertanto, emerge la necessità di sviluppare tecniche di ottimizzazione specifiche per il Tiny Machine Learning, che consentano di massimizzare l'efficienza del software, dell'hardware e della gestione dei dati. Malgrado ciò, le ottimizzazioni dei modelli possono potenzialmente comportare modifiche ai requisiti non funzionali, che devono essere oggetto di analisi durante lo sviluppo dell'applicazione.

2.2.1 Sviluppo di un modello Tiny Machine Learning

Lo sviluppo di modelli di Tiny Machine Learning richiede un approccio mirato che tenga conto delle limitazioni specifiche dei dispositivi embedded. Spesso, si parte da modelli creati tramite il Classical Machine Learning [13] che poi vengono ottimizzati. Il processo per implementare un modello di Tiny Machine Learning richiede diversi passaggi chiave:

- Selezione del task e del modello: Il primo passo consiste nel definire il compito di machine learning da affrontare e selezionare il modello più adatto alle

¹*Tensorflow Lite*: <https://www.tensorflow.org/>

²*Pytorch*: <https://pytorch.org/>

esigenze del problema e alle risorse computazionali disponibili.

- **Ottimizzazione delle risorse:** Una volta scelto il modello, è importante ottimizzarne l'architettura per adattarlo alle limitate risorse computazionali del dispositivo target. Questa ottimizzazione può riguardare la riduzione del numero di parametri, la semplificazione di calcoli o l'implementazione di operazioni meno costose dal punto di vista computazionale.
- **Tecniche di ottimizzazione:** Utilizzare tecniche di ottimizzazione specifiche per il Tiny Machine Learning, come la quantizzazione, il pruning e la knowledge distillation per ridurre la complessità del modello senza comprometterne le prestazioni.
- **Adattamento ai vincoli di memoria e potenza:** È importante tenere conto dei vincoli di memoria e potenza del dispositivo, in maniera tale da assicurare che il modello ottimizzato possa essere utilizzato efficientemente senza sovraccaricare le risorse disponibili.
- **Validazione e test:** È importante testare e validare il modello ottimizzato per verificare che i requisiti non funzionali non compromettano la qualità e per garantire che mantenga prestazioni accettabili.

2.2.2 Ottimizzazione dei Modelli di Machine Learning

L'ottimizzazione dei modelli di Tiny Machine Learning può essere eseguita in due fasi principali:

- **Durante l'addestramento:** Durante questa fase, l'architettura del modello viene progettata per ridurre al minimo le risorse computazionali richieste. Il *pruning incrementale*, come descrive Han et al. [14], è un esempio di tecnica usata durante l'addestramento dei modelli che riduce progressivamente i pesi meno significativi della rete.
- **Post-addestramento:** Dopo l'addestramento, il modello può essere ulteriormente ottimizzato mediante varie tecniche. La quantizzazione è una delle più

comuni, in cui i pesi vengono ridotti da rappresentazioni a 32-bit a 8-bit senza una significativa perdita di precisione, come mostrato da Jacob et al. [15].

2.3 Limitazioni e Obiettivi

Nonostante i notevoli progressi nel campo del Tiny Machine Learning, la letteratura attuale presenta alcune limitazioni rilevanti. Molti studi si focalizzano su singole tecniche, come la quantizzazione o il pruning, senza esplorare in modo esaustivo una panoramica integrata che consideri l'intera gamma di metodi utilizzabili. Inoltre, le ricerche spesso trascurano l'impatto del consumo energetico dei modelli ottimizzati, limitandosi a valutazioni basate principalmente su metriche di accuratezza. Di conseguenza, manca una visione completa che permetta di comprendere appieno i trade-off tra performance e sostenibilità.

Questo lavoro di tesi si propone di avanzare lo stato dell'arte attraverso due contributi. In primo luogo, verrà condotta una ricerca approfondita di tutte le tecniche di minimizzazione dei modelli di machine learning utilizzate in letteratura. Saranno esplorate le metriche di valutazione comunemente impiegate, nonché i requisiti non funzionali che tali tecniche cercano di ottimizzare. Inoltre, verranno esaminati i task su cui queste tecniche sono applicate e verranno esaminati i momenti specifici in cui queste tecniche vengono utilizzate durante il processo di addestramento o post-addestramento dei modelli. In secondo luogo, la ricerca si concentrerà sull'analisi dei trade-off tra performance e sostenibilità, valutando in che modo le tecniche di ottimizzazione influenzano le prestazioni dei modelli in termini di accuratezza e consumo energetico.

Tecniche di Minimizzazione dei Modelli di Machine Learning: Una Systematic Literature Review

L'obiettivo del presente studio è duplice: da un lato, individuare le tecniche e gli approcci noti in letteratura per ridurre le dimensioni di un modello di machine learning, dall'altro, esplorare il contesto in cui queste tecniche vengono maggiormente applicate. Nello specifico, si cercherà di comprendere quale sia il task di machine learning più frequente su cui tali tecniche vengono utilizzate, e saranno analizzati i requisiti non funzionali e le metriche che vengono prese in considerazione negli studi esistenti. Una volta ottenute queste informazioni, si procederà a realizzare uno studio di benchmark, focalizzato sul task più popolare. Questo studio avrà lo scopo di osservare come variano i requisiti non funzionali e le metriche sull'accuratezza del modello in seguito all'applicazione delle diverse tecniche di ottimizzazione. L'intento è fornire una valutazione completa dell'impatto che queste tecniche hanno sul modello.

Il processo e i risultati della systematic literature review sono reperibili al repository GitHub: <https://github.com/antgioia/benchmarkTinyML>

3.1 Systematic Literature Review

Nel contesto della ricerca scientifica, una *Systematic Literature Review* rappresenta uno strumento fondamentale per raccogliere, valutare e sintetizzare in modo strutturato le conoscenze esistenti su un determinato argomento. L'obiettivo di questo capitolo è presentare il processo e i risultati della systematic literature review condotta per identificare e analizzare le tecniche di ottimizzazione della dimensione applicabili ai modelli di Machine Learning.

Questa systematic literature review è stata progettata seguendo le linee guida di Kitchenham et al. [16]. Per mitigare le minacce dovute all'incompletezza della ricerca e assicurare un'analisi esaustiva delle tecniche di ottimizzazione della dimensione dei modelli di Machine Learning, è stato implementato un processo di ricerca potenziato basato sulla tecnica dello *snowballing*[17]. Quest'ultima prevede la scansione dei riferimenti bibliografici sia in entrata che in uscita degli studi primari identificati durante la fase iniziale di ricerca. Tale metodologia consente di individuare fonti di informazioni aggiuntive che potrebbero essere state trascurate. Attraverso questo approccio, si intende aumentare l'affidabilità dei risultati ottenuti nella successiva fase di analisi e valutazione.

3.1.1 Research Questions

L'obiettivo della systematic literature review è di individuare e classificare le tecniche presenti in letteratura capaci di ottimizzare la dimensione di un modello di Machine Learning. Per questa ragione, sono state formulate diverse research questions (**RQs**) mirate ad analizzare vari aspetti delle tecniche di ottimizzazione delle dimensioni dei modelli di machine learning.

La prima research question (**RQ1**) si focalizza sull'identificazione e la classificazione delle tecniche attualmente utilizzate per ridurre la dimensione dei modelli di Machine Learning. Sebbene siano state condotte precedenti ricerche sull'argomento, al momento non esiste una classificazione sistematica delle diverse tecniche in uso.

Q RQ1. *Quali sono le tecniche usate per ridurre la dimensione di un modello di Machine Learning?*

Oltre a identificare le tecniche, è importante comprendere in quali contesti queste tecniche sono applicate. Pertanto, la seconda research question (**RQ2**) si propone di indagare i task specifici per i quali vengono utilizzate le tecniche di minimizzazione della dimensione dei modelli.

Q RQ₂. *Per quali task sono usate tecniche di ridimensionamento di modelli di Machine Learning?*

Un altro aspetto riguarda la fase del ciclo di vita di sviluppo in cui queste tecniche vengono implementate. Infatti, la terza research question (**RQ3**) mira a classificare le tecniche in base alla loro applicazione nelle diverse fasi dello sviluppo, come il pre-training, in-training, o post-training.

Q RQ₃. *In quali fasi dello sviluppo (pre-training, in-training, post-training) le tecniche di ridimensionamento di modelli di Machine Learning sono implementate?*

Oltre all'ottimizzazione della dimensione del modello, le tecniche di ridimensionamento possono anche mirare a migliorare altri requisiti non-funzionali. La quarta research question (**RQ4**) esplora quali requisiti non-funzionali sono ottimizzati attraverso l'uso di queste tecniche.

Q RQ₄. *Quali requisiti non-funzionali le tecniche di ridimensionamento di modelli di Machine Learning mirano ad ottimizzare?*

Infine, per valutare l'efficacia di queste tecniche, è necessario analizzare i criteri e le metodologie di valutazione impiegate. La quinta research question (**RQ5**) è dedicata a esaminare i metodi utilizzati per valutare le tecniche di ridimensionamento nei vari studi.

Q RQ₅. *In che modo le tecniche di ridimensionamento di modelli di Machine Learning sono valutate?*

3.1.2 Definizione della Query di Ricerca

Uno dei principali passi di una systematic literature review è l'identificazione di termini di ricerca appropriati che possano aiutare il recupero dell'insieme completo di fonti. A questo scopo, è stata adottata la seguente strategia:

- Sono state inizialmente identificate le parole chiavi più rilevanti, che hanno composto la base per la ricerca;

- Sono stati individuati i possibili sinonimi o parole alternative per tutte le parole chiavi rilevanti;
- Sono stati utilizzati operatori booleani, quali AND e OR, per comporre la research question.

È stata formulata la seguente query di ricerca:

Stringa di ricerca: *TITLE (("machine learning model" OR "ML model" OR "deep learning model" OR "DL model") AND ("reduc*" OR "minimiz*" OR "compress*" OR "shrink*" OR "optimiz*" OR "diminish" OR "trim" OR "decreas*" OR "knowledge distillation" OR "quantiz*" OR "prun*") AND ("technique" OR "approach" OR "algorithm" OR "method"))*

La query di ricerca è stata costruita per garantire una copertura esaustiva degli argomenti rilevanti. In particolare, i sinonimi per ciascun concetto sono stati combinati utilizzando l'operatore logico OR, consentendo così una vasta raccolta di fonti. A ogni modo, i diversi concetti sono stati combinati con l'operatore AND, affinché la ricerca si concentri su fonti che discutono contemporaneamente modelli di Machine Learning e Deep Learning in relazione a tecniche, algoritmi, approcci o metodi e agli aspetti ottimizzazione dei modelli stessi. È importante notare che la ricerca è stata eseguita esclusivamente sui titoli delle pubblicazioni scientifiche, permettendo così di ottenere un insieme completo di fonti attinenti.

3.1.3 Database di ricerca

La ricerca delle fonti è stata condotta attraverso tre importanti biblioteche digitali: *Web of Science*¹, *Scopus*² e *IEEE Xplore*³. Queste biblioteche forniscono filtri appositamente progettati per recuperare i documenti in base alla query di ricerca definita. Questa selezione mirata delle biblioteche digitali ha permesso di concentrare gli sforzi di ricerca su piattaforme consolidate e affidabili, ottimizzando così il processo di individuazione e selezione delle fonti.

¹*Web of Science*: <https://www.webofscience.com/>

²*Scopus*: <https://www.scopus.com/>

³*IEEE Xplore*: <https://ieeexplore.ieee.org/>

3.1.4 Criteri di inclusione ed esclusione

I criteri di inclusione ed esclusione consentono di selezionare le risorse che rispondono alle research question di una systematic literature review [16]. Nel contesto di questo studio, i documenti recuperati dal processo di ricerca sono stati valutati in base ai seguenti criteri di esclusione e inclusione.

- *Criteri di inclusione.* Sono state incluse le risorse che discutevano di approcci, tecniche, metodi e algoritmi di ottimizzazione della dimensione di un modello di Machine Learning.
- *Criteri di esclusione* Sono state escluse le risorse che rispettano i seguenti vincoli:
 - Articoli non scritti in inglese;
 - Articoli duplicati;
 - Articoli brevi, con un numero di pagina inferiore a cinque;
 - Articoli la cui lettura integrale del documento non è disponibile;
 - Articoli non pubblicati o non sottoposti a peer review;
 - Articoli di workshop e sistematici;
 - Articoli fuori ambito

3.1.5 Metodo di Ricerca per condurre il processo di Snowballing

Una volta individuate le risorse nelle biblioteche digitali, è stata eseguita la tecnica dello snowballing [17]. Questa metodologia ha consentito di esaminare sistematicamente tutti i riferimenti bibliografici sia in entrata che in uscita dagli studi primari selezionati dal criterio di inclusione. Questi sono stati analizzati per identificare ulteriori risorse rilevanti per soddisfare la research question.

Nella pratica, il processo di snowballing si è articolato in due fasi: *forward snowballing* e *backward snowballing*. Il forward snowballing è stato condotto utilizzando *Google Scholar*⁴ per individuare gli articoli che hanno citato le risorse primarie selezionate, mentre il backward snowballing ha implicato l'esame dei riferimenti bibliografici

⁴*Google Scholar*: <https://scholar.google.com/>

presenti negli articoli originali. La combinazione di queste tecniche ha consentito di ottenere una visione completa delle risorse rilevanti e di espandere in modo significativo la base di studi inizialmente individuati. Il processo è stato iterativo e si è svolto in tre cicli di ricerca. Nel primo ciclo, sono stati presi in considerazione tutti i riferimenti degli studi primari selezionati, sia in entrata che in uscita. Successivamente, sono stati applicati i criteri di inclusione ed esclusione per selezionare gli studi pertinenti. Il secondo ciclo ha riguardato le risorse identificate nel primo ciclo, applicando nuovamente lo stesso processo di inclusione ed esclusione. Infine, il terzo ciclo ha completato l'analisi, raggiungendo una saturazione delle risorse raccolte, cioè il punto in cui non emergevano più nuove fonti rilevanti. Per facilitare l'intero processo, è stato utilizzato lo strumento Google Scholar, che ha reso più efficiente la gestione e l'analisi delle citazioni e dei riferimenti bibliografici.

3.1.6 Metodo di Ricerca per condurre la Valutazione della Qualità

Prima di procedere con l'estrazione dei dati necessari per rispondere alla research question, è stata eseguita un'attenta valutazione della qualità e della completezza delle risorse che hanno superato con successo il criterio di inclusione. Questo processo è stato fondamentale per scartare i documenti che non offrivano informazioni sufficientemente rilevanti da utilizzare nello studio. Solo le risorse che soddisfacevano criteri rigorosi di qualità sono state considerate per l'estrazione dei dati. L'implementazione del processo di valutazione della qualità è iniziata con la definizione di quattro domande qualitative volte a capire se un determinato studio fosse rilevante per rispondere alle research questions. Le domande di valutazione della qualità sono:

- **Q1:** L'articolo usa almeno un approccio per la riduzione della grandezza di un modello?
- **Q2:** L'articolo descrive i task utilizzati nel contesto dell'utilizzo di tecniche di ridimensionamento?
- **Q3:** L'articolo descrive in quale fase dello sviluppo sono state utilizzate le tecniche di ridimensionamento?

- **Q4:** L'articolo descrive le metriche di valutazione utilizzate per valutare le tecniche di ridimensionamento?

Nel processo di valutazione degli studi rispetto a ciascuna domanda qualitativa, è stato adottato un approccio *fuzzy linguistic* [18] che consiste nel valutare ciascuno studio primario tramite una punteggio continuo compreso tra 0 e 1, riflettendo il grado di soddisfacimento della specifica domanda. I punteggi sono stati assegnati secondo la seguente scala:

- 0 Non Soddisfatta
- 0.1 - 0.3 Poco Soddisfatta
- 0.4 - 0.6 Parzialmente Soddisfatta
- 0.7 - 0.9 Molto Soddisfatta
- 1 Soddisfatta

Al termine della valutazione condotta per ciascuna domanda, il punteggio totale per ciascuno studio è stato calcolato sommando i punteggi assegnati per ogni domanda qualitativa. Infatti:

- 0 Non Soddisfatta
- 0.1 - 1 Poco Soddisfatta
- 1.1 - 2 Parzialmente Soddisfatta
- 2.1 - 3 Molto Soddisfatta
- 3.1 - 4 Soddisfatta

Infine, gli studi che hanno ottenuto un punteggio complessivo superiore o uguale a 2 sono stati selezionati per essere analizzati nei dettagli ed estrarre informazioni.

3.1.7 Form di Estrazione Dati

Come ultimo passo della systematic literature review, è stato progettato un modulo di estrazione dati, definendo le informazioni da raccogliere. La Tabella 3.1 riassume i dati raccolti, riportando:

- la categoria a cui si riferisce il gruppo di attributi;
- lo scope in cui sono utilizzati i dati;
- la descrizione della categoria considerata;
- gli attributi specifici considerati.

Motivazione	Scope	Descrizione	Attributi Raccolti
Informazioni sul documento	Bibliometrici	Questa componente include informazioni generali sull'articolo e lo score assegnato	Titolo Autore Anno di pubblicazione URL Score
Tecniche di minimizzazione della size di un modello di machine learning	RQ1	Questa componente indica le tecniche di minimizzazione utilizzate	Tecniche utilizzate
Task	RQ2	Questa componente indica per quali task le tecniche di minimizzazione della size dei modelli di machine learning sono utilizzate	Task
Fasi di sviluppo	RQ3	Questa componente indica in che fase del processo di sviluppo le tecniche di minimizzazione della size dei modelli di machine learning sono implementate	Fase di implementazione
Requisiti non funzionali	RQ4	Questa componente indica quali requisiti non funzionali le tecniche di minimizzazione della size dei modelli di machine learning mirano ad ottimizzare	Requisiti non funzionali
Metriche di valutazione	RQ5	Questa componente indica in che modo le tecniche di minimizzazione della size dei modelli di machine learning sono valutate	Metriche

Tabella 3.1: Form Estrazione Dati

In primo luogo, sono stati identificati una serie di attributi che potessero fornire una descrizione statistica dei campioni, includendo elementi come il titolo, l'autore,

l'anno di pubblicazione, l'URL e il punteggio ottenuto nella fase di *quality assessment*. Oltre a queste informazioni, sono stati raccolti e archiviati dati relativi agli approcci di minimizzazione della dimensione dei modelli di machine learning, al task specifico, alla fase di implementazione dell'approccio, ai requisiti non funzionali e alle metriche di valutazione.

I requisiti non funzionali sono stati basati sullo standard *ISO 25010*⁵ e sono stati arricchiti con ulteriori requisiti relativi al consumo energetico e alla sostenibilità.

3.1.8 Esecuzione della Ricerca

La figura 3.1 riassume i risultati della systematic literature review. Inizialmente,

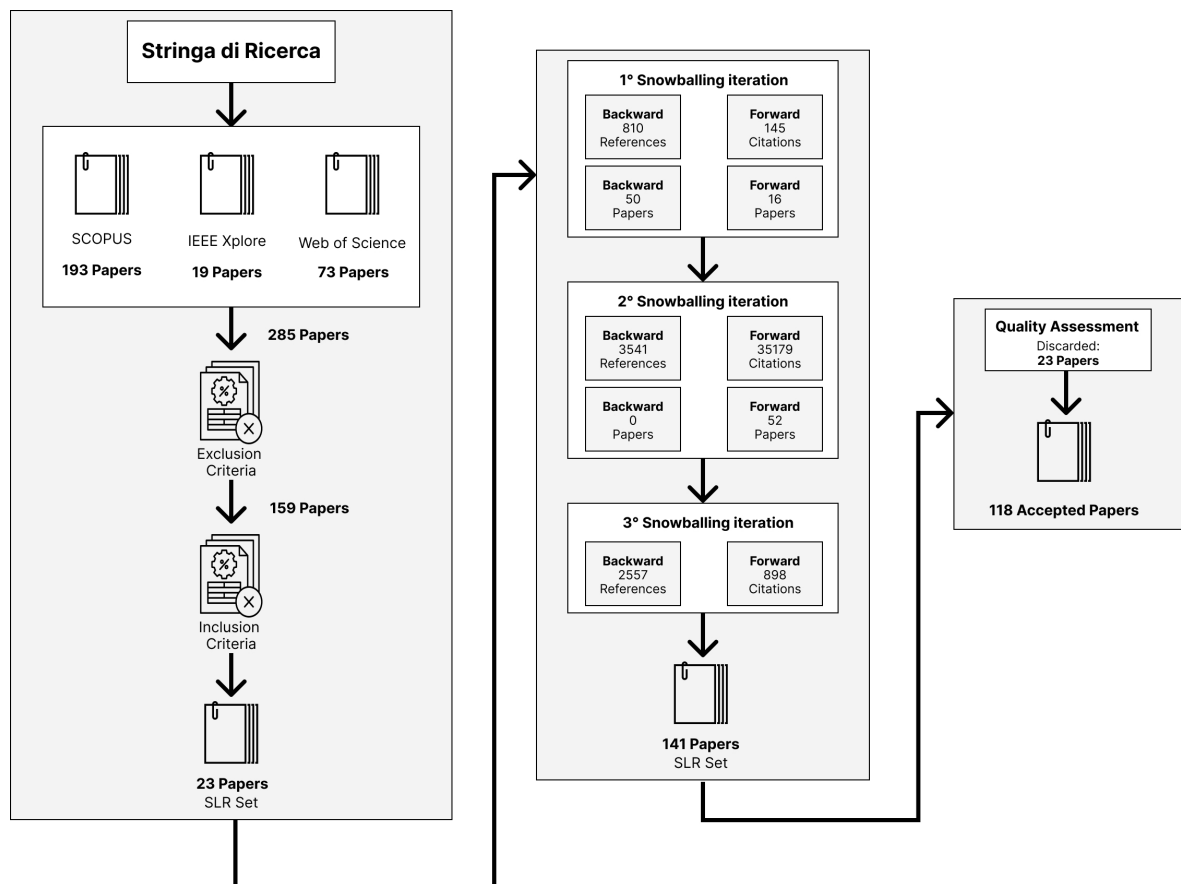


Figura 3.1: Processo per la Selezione degli articoli

sono stati identificati 285 articoli, suddivisi tra 193 provenienti da *Scopus*, 19 da *IEEE Xplore* e 73 da *Web of Science*. A questi articoli sono stati applicati vari criteri di

⁵ISO 25010: <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>

esclusione, come la rimozione dei duplicati (90 articoli), la rimozione degli articoli di workshop (7 articoli), l'eliminazione degli articoli non disponibili (24 articoli) e la rimozione di articoli con meno di 5 pagine (5 articoli). Dopo l'applicazione di questi criteri, il numero di articoli è stato ridotto a 159. Successivamente, sono stati applicati i criteri di inclusione, che richiedevano che gli articoli trattassero approcci, tecniche o metodologie legate al ridimensionamento di modelli di machine learning, riducendo ulteriormente il numero di articoli a 23. Poiché il numero di articoli risultava essere molto basso, è stata applicata la tecnica dello snowballing per espandere la ricerca. Utilizzando Google Scholar, il processo di snowballing è stato condotto in tre cicli distinti. Nel primo ciclo, sono stati esaminati 50 articoli con il metodo del backward snowballing e 16 articoli con il metodo del forward snowballing, per un totale di 66 articoli aggiunti. Nel secondo ciclo, sono stati esaminati 52 articoli tramite il forward snowballing. Il terzo ciclo non ha prodotto ulteriori articoli. Complessivamente, il processo di snowballing ha portato all'aggiunta di 118 articoli, portando il totale a 141 articoli. Dopo questa espansione, è stata eseguita una fase di quality assessment per valutare la rilevanza e la qualità degli articoli raccolti. A seguito di questa valutazione, 23 articoli sono stati esclusi, portando infine a un totale di 118 articoli selezionati per lo studio finale. Durante il processo di snowballing, sono stati esaminati complessivamente 43.130 articoli: 6.908 articoli tramite il backward snowballing e 36.222 articoli tramite il forward snowballing in vari livelli di approfondimento. In particolare, sono stati esaminati 955 articoli al primo livello, 38.720 al secondo livello e 3.455 al terzo livello di snowballing.

3.2 Analisi dei risultati

Di seguito sono mostrati i risultati ottenuti dalla fase di *data extraction* a partire dagli articoli selezionati nella fase di *quality assessment*.

3.2.1 Analisi Bibliometrica

La fase di quality assessment, come si può osservare dall'immagine 3.2, ha evidenziato che, tra i 118 articoli selezionati, la maggior parte ha ottenuto punteggi di qualità elevati. In particolare, l'80% degli articoli ha ricevuto una valutazione di qualità "Yes", indicando che questi studi hanno fornito informazioni dettagliate ed esplicite per affrontare le nostre research questions. Il restante 20% degli articoli ha ottenuto una valutazione di qualità "Mostly", dimostrando una discreta qualità e completezza delle informazioni fornite.

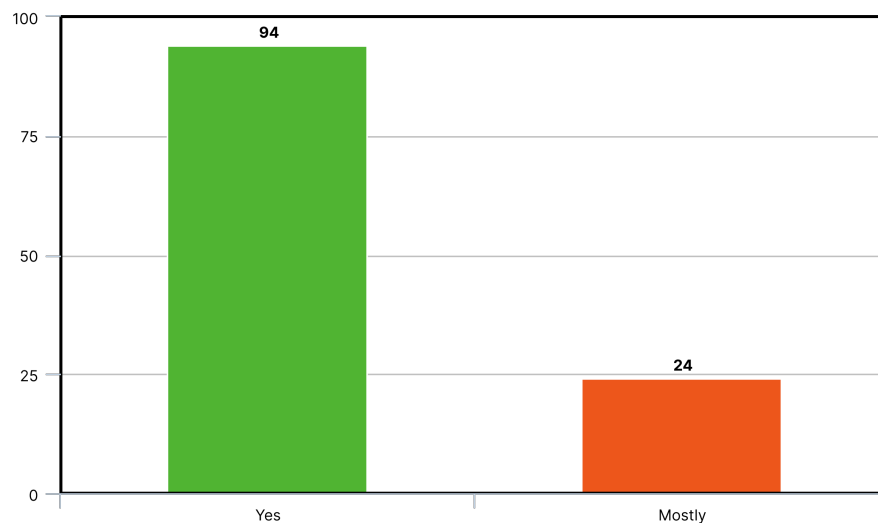


Figura 3.2: Qualità articoli selezionati

Questi risultati ci permettono di affermare che l'83% degli articoli selezionati, sia dalla ricerca iniziale sia attraverso il processo di snowballing, soddisfa adeguatamente i criteri di qualità richiesti per rispondere alle research questions. Inoltre, dei 118 articoli, 17 sono stati selezionati tramite la query di ricerca, mentre il restante è stato individuato attraverso la tecnica dello snowballing. Questo dimostra che questa tecnica ha portato all'inclusione di molti risultati che altrimenti sarebbero stati ignorati, ampliando il campione di studi per l'analisi.

Dall’analisi degli articoli emerge che già dal 1993 si è iniziato a discutere di tecniche di minimizzazione delle dimensioni dei modelli di machine learning. Tuttavia, come si può osservare dall’immagine 3.3, è a partire dal 2020 che questa necessità si è diffusa in maniera più significativa. Questo incremento recente riflette il crescente interesse della comunità scientifica verso l’ottimizzazione dei modelli di machine learning.

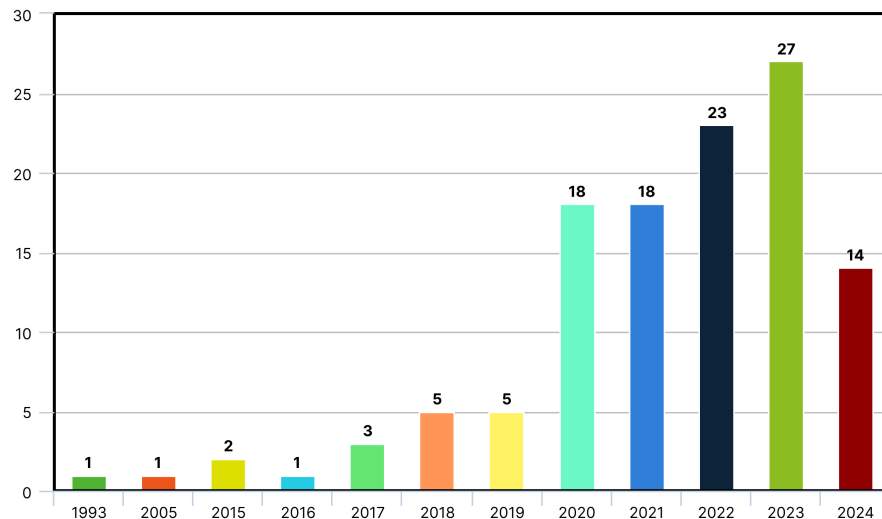


Figura 3.3: Anno pubblicazione articoli selezionati

3.2.2 RQ1: Quali sono le tecniche usate per ridurre la dimensione di un modello di Machine Learning?

Il primo obiettivo della systematic literature review è quello di individuare gli approcci e le tecniche di minimizzazione della dimensione di un modello di machine learning. Analizzando gli articoli selezionati, queste tecniche includono una varietà di approcci mirati a ridurre la complessità e le risorse computazionali, mantenendo al contempo buone prestazioni. Alcune di queste tecniche sono specificamente progettate per ridurre la dimensione del modello, mentre altre, pur non avendo come obiettivo principale la riduzione della dimensione, contribuiscono comunque all’ottimizzazione generale delle reti neurali e alla loro efficienza.

Gli studi considerati sono stati inizialmente analizzati per identificare le tecniche di minimizzazione della dimensione del modello utilizzate. Successivamente, tale tecniche sono state raggruppate per somiglianza. Ad esempio, tecniche come *pruning*

dei canali e *pruning dei kernel* sono stati raggruppati nella categoria *pruning*, poiché entrambe consistono nell'eliminazione di connessioni, neuroni o interi canali e layer poco rilevanti. Le tecniche individuate sono mostrate nelle Tabelle 3.2, 3.3 e 3.4.

Tecniche	Numero	Articoli	Significato
Pruning	59	[SLR1, SLR2, SLR3, SLR4, SLR5, SLR6], [SLR7, SLR8, SLR9, SLR10, SLR11, SLR12], [SLR13, SLR14, SLR15, SLR16, SLR17, SLR18] [SLR19, SLR20, SLR21, SLR22, SLR23, SLR24], [SLR25, SLR26, SLR27, SLR28, SLR29, SLR30], [SLR31, SLR32, SLR33, SLR34, SLR35, SLR36], [SLR37, SLR38, SLR39, SLR40, SLR41, SLR42], [SLR43, SLR44, SLR45, SLR46, SLR47, SLR48], [SLR49, SLR50, SLR51, SLR52, SLR53], [SLR54, SLR55, SLR56, SLR57, SLR58, SLR59]	Rimozione dei pesi meno significativi
Pruning dei canali e dei layer	1	[SLR60]	Rimozione di interi canali o layer
Pruning delle Mappe e delle Caratteristiche	1	[SLR61]	Eliminazione di feature maps meno rilevanti
Pruning del kernel	1	[SLR61]	Rimozione di kernel convoluzionali ridondanti
Pruning con sparsità strided intra-kernel	1	[SLR61]	Applica sparsità a livello intra-kernel, mantenendo alcuni pattern strided
Trainable Energy Aware Pruning	1	[SLR62]	Pruning che tiene conto del consumo energetico
Quantizzazione	41	[SLR1, SLR3, SLR63, SLR7, SLR64, SLR8, SLR9] [SLR10, SLR11, SLR13, SLR65, SLR66, SLR67] [SLR68, SLR69, SLR28, SLR29, SLR70, SLR30] [SLR71, SLR40, SLR72, SLR43, SLR73, SLR44] [SLR46, SLR74, SLR75, SLR76, SLR77, SLR78] [SLR79, SLR47, SLR80, SLR81, SLR82] [SLR52, SLR53, SLR83, SLR57, SLR59]	Riduce la precisione numerica dei pesi e delle attivazioni

Tabella 3.2: Tecniche di Minimizzazione della Dimensione

Tecniche	Numero	Articoli	Significato
AE-Qdrop per la quantizzazione	1	[SLR84]	Combina quantizzazione con dropout adattivo per ridurre l'errore di quantizzazione
Binarizzazione	3	[SLR85, SLR86, SLR40]	Converte i pesi e le attivazioni in valori binari
Knowledge Distillation	28	[SLR1, SLR3, SLR7, SLR87, SLR11, SLR13] [SLR88, SLR89, SLR90, SLR91, SLR40, SLR92] [SLR93, SLR73, SLR94, SLR46, SLR95, SLR96] [SLR97, SLR98, SLR99, SLR100, SLR47, SLR54] [SLR101, SLR102, SLR59, SLR21]	Processo in cui modello insegnante trasferisce conoscenza a uno studente
Self Distillation	1	[SLR103]	Il modello stesso funge da insegnante per migliorare le prestazioni senza richiedere un modello separato
Low rank approximation	3	[SLR11, SLR13, SLR46]	Decomposizione delle matrici di peso in fattori di rango inferiore
Fattorizzazione a basso rango	3	[SLR16, SLR18, SLR57]	Decomposizione delle matrici di peso in prodotti di matrici più piccole
Decomposizione Tucker-CP	1	[SLR104]	Metodo per decomporre un tensore in componenti a basso rango
Compressione Lossy	1	[SLR105]	Riduce la dimensione del modello sacrificando una parte dell'accuratezza
TinyNAS e TinyEngine	1	[SLR106]	Framework per creare modelli piccoli
Convoluzione leggera	4	[SLR16, SLR107, SLR108, SLR54]	Riduce il numero di operazioni convoluzionali

Tabella 3.3: Tecniche di Minimizzazione della Dimensione

Tecniche	Numero	Articoli	Significato
MobileNetV3	1	[SLR109]	Modello leggero ottimizzato
RNNPool	1	[SLR110]	Pooling ottimizzati per reti ricorrenti
LogNNet	2	[SLR117, SLR118]	Rete neurale con architettura
Algoritmi evolutivi	2	[SLR111, SLR112]	Tecniche che utilizzano principi di evoluzione biologica
Quantum Autoencoder	1	[SLR113]	Utilizza il calcolo quantistico per comprimere le informazioni
Group Teaching Optimization Algorithm	1	[SLR114]	Più modelli (insegnanti) influenzano un singolo modello (studente)
Random Projection Algorithm	1	[SLR115]	Proiezione casuale delle caratteristiche ad alta dimensione in uno spazio a dimensione inferiore
Sostituzione della memoria	1	[SLR116]	Tecnica che ottimizza l'utilizzo della memoria durante l'addestramento e l'inferenza

Tabella 3.4: Tecniche di Minimizzazione della Dimensione

Le tecniche individuate per la minimizzazione della dimensione dei modelli di machine learning possono essere suddivise in due categorie principali: quelle che mirano direttamente a ridurre la dimensione del modello e quelle che, pur avendo obiettivi diversi, possono contribuire indirettamente a questa riduzione. Tra le tecniche con l'obiettivo primario di minimizzare la dimensione del modello troviamo il *pruning*, la *quantizzazione*, la *knowledge distillation*, la *low rank approximation* e la *compressione lossy*. I *modelli light-weight* sono anch'essi progettati con l'obiettivo di

essere compatti, ottimizzando l'architettura per l'utilizzo su dispositivi con risorse limitate. Dall'altro lato, le tecniche come gli *algoritmi di ottimizzazione evolutiva*, il *quantum autoencoding*, il *group teaching optimization algorithm*, il *random projection algorithm* e la *strategia di sostituzione della memoria* non hanno come obiettivo principale la riduzione della dimensione, ma possono portare indirettamente a modelli più piccoli.

🔗 **Answer to RQ₁.** Le tecniche principali per ridurre la dimensione dei modelli di machine learning includono *pruning*, *quantizzazione*, *knowledge distillation*, *approssimazione a basso rango*, *compressione lossy* e *modelli light-weight*. Tecniche indirette comprendono *algoritmi evolutivi*, *quantum autoencoder*, *random projection* e *ottimizzazione della memoria*.

3.2.3 RQ2: Per quali task sono usate tecniche di ridimensionamento di modelli di Machine Learning?

La systematic literature review ha permesso di identificare un'ampia gamma di task di machine learning a cui sono applicate tecniche di ridimensionamento dei modelli. Durante il processo di analisi, per facilitare l'individuazione e la categorizzazione dei task, è stato fatto riferimento ai nomi dei task presenti nel catalogo di *Hugging Face*⁶, che ha contribuito a uniformare la terminologia e a evitare ambiguità. I risultati mostrano una prevalenza significativa nell'ambito della *Computer Vision*, con 94 studi focalizzati sull'interpretazione e l'analisi di immagini e video. Altri task rilevanti includono *Natural Language Processing* e *Signal Classification*, con 4 studi ciascuno. La ricerca ha inoltre evidenziato applicazioni specifiche, ognuna delle quali è stata identificata una sola volta. Questi includono il *Biosignal Processing*, la *rilevazione e la diagnosi dei guasti*, la *classificazione*, il *Time Series Prediction con elementi di Spatiotemporal Modeling*, la *Sound Classification*, il *Federated Learning*, e l'*Analisi di Big Data*. In aggiunta ai task, lo studio ha permesso di riportare anche la varietà di modelli di machine learning utilizzati nelle tecniche di ridimensionamento. Tra questi, il modello più frequentemente citato è *VGG16*, presente in 14 studi. Altri modelli ampiamente utilizzati includono *AlexNet* e *ResNet* (entrambi con 13 citazioni).

⁶*Hugging Face*: <https://huggingface.co/models>

Seguono *MobileNetV2* (9 studi), *ResNet50* e *MobileNet* (5 studi ciascuno). Modelli come *VGG* e *DenseNet* sono stati menzionati in 4 studi ciascuno. Modelli meno utilizzati, ma comunque rilevanti, includono le diverse versioni di *ResNet*, come *ResNet18*, *ResNet20*, e *ResNet32* (ciascuno con 3 studi), e altri come *ResNext* e le diverse varianti di *YOLO*, presenti in 2 studi ciascuno.

Infine, molti modelli sono stati identificati una sola volta nei vari studi, tra cui *VGG8*, *VGG19*, *HybridNet*, *InceptionResNetV2*, *EfficientNetB0*, *Shufflenet*, *U-net*, e molte altre architetture più specializzate, come *BERT-base*, *DistillBERT* e *GTOA-MLBOA*, ciascuna menzionata in un singolo studio.

🔗 **Answer to RQ₂.** Le tecniche di ridimensionamento dei modelli di machine learning sono principalmente applicate a task di *Computer Vision*, seguiti da *NLP* e *Signal Classification*. Altri task includono il *Biosignal Processing*, la *rilevazione e la diagnosi dei guasti*, la *classificazione*, il *Time Series Prediction con elementi di Spatiotemporal Modeling*, la *Sound Classification*, il *Federated Learning*, e l'*Analisi di Big Data*.

3.2.4 RQ3. In quali fasi dello sviluppo le tecniche di ridimensionamento di modelli di Machine Learning sono implementate?

Le tecniche di minimizzazione dei modelli di machine learning, come mostrato nella Tabella 3.5, possono essere applicate in diverse fasi del processo di addestramento, a seconda degli obiettivi dello studio o della natura stessa della tecnica. Analizzando gli studi giunti alla fase di estrazione dei dati, si osserva che tecniche come il *pruning* e la *quantizzazione*, sono spesso applicate durante la fase di addestramento, mentre risultano meno frequenti nel post-addestramento. In particolare, il *pruning* può essere integrato come passo iterativo all'interno del ciclo di addestramento, dove, dopo ogni epoca di addestramento, vengono eliminate le connessioni o i neuroni considerati meno rilevanti. Questo processo iterativo permette di mantenere elevata l'accuratezza del modello, riducendone al contempo la dimensione. Quando applicato dopo l'addestramento, il *pruning* riduce la struttura del modello una volta che è completamente addestrato, eliminando le parti che non contribuiscono significativamente alle predizioni. La *quantizzazione*, invece, può essere applicata sia durante l'addestramento che come passaggio finale, convertendo i pesi del modello in formati

Tecnica	Fase
Pruning	Addestramento Post-Addestramento
Quantizzazione	Addestramento Post-Addestramento
Knowledge Distillation	Post-Addestramento
Low rank approximation	Post-Addestramento
Compressione Lossy	Post-Addestramento
Modelli Light-Weight	Definizione dell'architettura
Algoritmi Evolutivi	Ottimizzazione
Quantum Autoencoder	Pre-Elaborazione
Group Teaching Optimization Algorithm	Ottimizzazione
Random Projection Algorithm	Pre-Elaborazione Post-Addestramento
Sostituzione della Memoria	Post-Addestramento

Tabella 3.5: Fasi Implementazione Tecniche

più compatti, come interi a bassa precisione. Altre tecniche, come la *knowledge distillation*, sono utilizzate nella fase di post-addestramento, dove un modello più grande e complesso viene usato per addestrare un modello più piccolo e leggero, trasferendo le conoscenze acquisite. Questo approccio permette di ottenere un modello molto più compatto che però mantiene significativamente salde le prestazioni. Le tecniche di *low rank approximation* e di *compressione lossy* vengono applicate come tecniche di ottimizzazione post-addestramento, mirando a decomporre matrici e a comprimere informazioni senza influire drasticamente sulle prestazioni del modello. I *modelli light-weight*, invece, sono progettati per essere compatti fin dall'inizio, con l'obiettivo di minimizzare la dimensione già nella fase di definizione dell'architettura, ben prima dell'inizio dell'addestramento. Questi modelli infatti nascono per essere utilizzati su dispositivi con risorse computazionali limitate, come smartphone o dispositivi IoT.

Gli algoritmi di *ottimizzazione evolutiva*, *quantum autoencoder* e *group teaching optimization* presentano un approccio differente rispetto alle tecniche precedentemente discusse, poiché il loro obiettivo principale non è la riduzione della dimensione del modello. Tuttavia, possono portare indirettamente a modelli più compatti e ottimizzati. Gli *algoritmi di ottimizzazione evolutiva* sono solitamente utilizzati nella fase di ottimizzazione del modello, dove si esplora uno spazio di soluzioni tramite la simulazioni di processi evolutivi, selezionando le migliori architetture o configurazioni di parametri che ottimizzano non solo le prestazioni del modello ma anche la sua efficienza in termini di dimensioni. Il *quantum autoencoder*, una tecnica emergente nel campo della computazione quantistica, viene utilizzato per ridurre la dimensionalità dei dati quantistici. Quest'ultimo viene applicato durante la fase di pre-elaborazione o compressione dei dati, prima dell'addestramento del modello quantistico, contribuendo così alla riduzione della dimensione del modello stesso quanto implementato. Il *group teaching optimization algorithm* è un approccio metaeuristico ispirato al processo di insegnamento-apprendimento di gruppo. Questo algoritmo può essere utilizzato in diverse fasi del ciclo di vita del modello, spesso nella fase di ottimizzazione, dove gruppi di soluzioni vengono iterativamente raffinati.

Anche altre tecniche come il *random projection algorithm* e la *strategia di sostituzione della memoria* sono impiegate principalmente come tecniche di pre-elaborazione od ottimizzazione post-addestramento. La prima riduce la dimensionalità dei dati in uno

spazio di dimensioni inferiori, mentre la seconda ottimizza l'uso della memoria durante l'esecuzione del modello, contribuendo così a una riduzione della dimensione complessiva del modello in termini di risorse richieste.

🔗 **Answer to RQ₃.** Le tecniche di ridimensionamento possono essere applicate durante l'addestramento (pruning, quantizzazione), dopo l'addestramento (pruning, quantizzazione, knowledge distillation, low rank approximation, compressione lossy, random project algorithm e sostituzione della memoria), nella definizione dell'architettura (modelli light-weight), nella fase di ottimizzazione (modelli evolutivi, group teaching optimization algorithm) e in fase di pre-elaborazione (quantum autoencoder, random project algorithm).

3.2.5 RQ4. Quali requisiti non-funzionali le tecniche di ridimensionamento di modelli di Machine Learning mirano ad ottimizzare?

Nel contesto della systematic literature review condotta, un'attenzione particolare è stata riservata all'analisi dei requisiti non funzionali che ogni studio ha trattato. Questi requisiti sono stati analizzati in conformità allo standard *ISO 25010*⁷, che definisce un quadro di riferimento per valutare la qualità dei sistemi software dal punto di vista non funzionale. Lo standard identifica nove categorie di requisiti non funzionali e a queste sono state aggiunte due ulteriori categorie relative al consumo energetico e alla sostenibilità. In definitiva le categorie di requisiti non funzionali considerati sono: *Adeguatezza funzionale, Efficienza delle prestazioni, Compatibilità, Capacità di interazione, Affidabilità, Sicurezza, Manutenibilità, Flessibilità, Safety, Consumo energetico, Sostenibilità*. Dall'analisi degli studi, i requisiti non funzionali più frequentemente affrontati sono stati:

- *Efficienza delle prestazioni*: 118 articoli;
- *Affidabilità*: 77 articoli
- *Flessibilità*: 72 articoli

⁷ISO 25010: <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>

- *Consumo energetico*: 67 articoli
- *Manutenibilità*: 53 articoli
- *Compatibilità*: 38 articoli

L'*efficienza delle prestazioni*, con particolare riferimento al *comportamento temporale* e all'*utilizzo delle risorse*, è un aspetto cruciale per i modelli di machine learning, specialmente in applicazioni su dispositivi con risorse limitate. In effetti, nel contesto delle tecniche della riduzione della dimensione dei modelli di machine learning è un fattore cruciale. L'*affidabilità* è un altro requisito essenziale, soprattutto in contesti critici. Gli studi che hanno impiegato tecniche di *knowledge distillation* hanno evidenziato come queste possano contribuire a mantenere elevati livelli di tolleranza ai guasti e di recuperabilità.

🔗 **Answer to RQ₄.** Le tecniche di ridimensionamento dei modelli di machine learning mirano principalmente a ottimizzare i seguenti requisiti non funzionali: *efficienza delle prestazioni, affidabilità, flessibilità, consumo energetico, manutenibilità e compatibilità*.

3.2.6 RQ5. In che modo le tecniche di ridimensionamento di modelli di Machine Learning sono valutate?

Le tecniche di ridimensionamento dei modelli di Machine Learning sono valutate utilizzando una varietà di metriche che riflettono diversi aspetti delle prestazioni, dell'efficienza, e della sostenibilità dei modelli. Dall'analisi degli articoli, le metriche più comuni includono l'*accuratezza* del modello, il *tempo di inferenza*, l'*uso delle risorse computazionali* (CPU/GPU) e la *dimensione del modello*. L'*accuratezza* è spesso utilizzata come metrica primaria per valutare l'effetto del ridimensionamento sulla capacità predittiva del modello, con metriche secondarie come la *precisione*, il *recall*, l'*AUC-ROC* e la *matrice di confusione* che forniscono ulteriori dettagli sulle prestazioni.

Inoltre, molte tecniche sono valutate in termini di prestazioni computazionali, come il *tempo di inferenza* e il *throughput*. Ad esempio, l'*uso della memoria*, la *velocità operativa* e la *scalabilità* sono metriche importanti per determinare l'efficacia delle tecniche di ridimensionamento, specialmente su dispositivi con risorse limitate.

Infine, la sostenibilità è un aspetto emergente nelle valutazioni, con metriche che considerano il *consumo energetico* e l'*impatto ambientale*.

📌 **Answer to RQ₅.** Le tecniche di ridimensionamento dei modelli di machine learning sono valutate principalmente attraverso l'*accuratezza*, la *precisione*, il *recall*, l'*AUC-ROC*, il *tempo di inferenza*, il *throughput*, l'*uso delle risorse computazionali (CPU/GPU)*, e la *dimensione del modello*, con un'attenzione crescente verso il *consumo energetico* e la *sostenibilità*.

Studio di Benchmark: Computer Vision

Una volta individuati gli approcci adottati in letteratura per la riduzione della dimensione dei modelli di Machine Learning, è stato avviato uno studio di benchmark volto a esaminare come queste tecniche impattano sulle metriche e sui consumi energetici. Questo studio si focalizza in particolare sul task di *Computer Vision*, poiché è in questo ambito che si riscontrano le applicazioni più frequenti delle tecniche di riduzione delle dimensioni, come evidenziato dai risultati ottenuti dalla systematic literature review.

Lo studio si concentra in particolare sulle tecniche maggiormente utilizzate, quali la **quantizzazione**, il **pruning** e la **low rank approximation**, analizzando come le modifiche nella dimensione e nella complessità dei modelli influiscano su parametri quali l'accuratezza, la precisione, il recall, l'AUC-ROC, il tempo di inferenza, il throughput e il consumo energetico. L'obiettivo è fornire una panoramica completa dei trade-off associati all'ottimizzazione delle dimensioni dei modelli, contribuendo così alla definizione di linee guida per la scelta delle strategie più efficaci.

L'esperimento eseguito è reperibile al repository GitHub:

<https://github.com/antgioia/benchmarkTinyML>

4.1 Studio di Benchmark

Nel contesto di questa ricerca scientifica, lo studio di benchmark rappresenta una metodologia di ricerca fondamentale per raccogliere informazioni fondamentali sulle tecniche di ottimizzazione della dimensione dei modelli di machine learning. L'obiettivo del presente capitolo è presentare il processo e l'analisi dei risultati dello studio di benchmark condotto per analizzare il trade-off tra performance e sostenibilità. Lo studio è iniziato con la definizione degli oggetti di ricerca, i soggetti dello studio per poi procedere alla definizione delle variabili dipendenti e indipendenti. Una volta stabilite le ipotesi sperimentali, è iniziata la valutazione delle variabili indipendenti, iniziando con quelle relative alle performance e all'accuratezza, seguite da quelle concernente il consumo energetico. Dopo aver ottenuto i risultati, per ogni modello è stata eseguita un'analisi statistica sui dati relativa ai consumi, al fine di ottenere informazioni sulle distribuzioni dei dati. Una volta ottenute le informazioni per ogni tecnica e per ogni modello, queste sono state nuovamente analizzate a trarre conclusioni sul trade-off tra performance e prestazioni per ogni tecnica.

4.1.1 Research Question

Per proseguire questo studio, è stato importante stilare una research question che ha seguito tutto l'esperimento.

Q RQ₆. *In che modo le tecniche di ottimizzazione trovate impattano i modelli ML sugli attributi di qualità?*

Questa research question si propone di esplorare l'influenza delle tecniche di minimizzazione della dimensione dei modelli di machine learning maggiormente utilizzate in letteratura sulla qualità dei modelli di machine learning. In particolare sono analizzate le tecniche di quantizzazione, low rank approximation, pruning globale non strutturato, pruning non strutturato, pruning strutturato per canali e il pruning casuale non strutturato. L'indagine si concentrerà su vari attributi di qualità, tra cui l'accuratezza, la precisione, il recall, l'AUC-ROC, il tempo di inferenza, il throughput e il consumo energetico. L'obiettivo è comprendere come queste tecniche possano migliorare o compromettere le prestazioni complessive dei modelli, bilan-

ciando la necessità di modelli più leggeri e veloci, e con un occhio rivolto al consumo energetico.

4.2 Oggetti dello Studio

L'oggetto dello studio è il dataset utilizzato durante la sperimentazione. È stato scelto *Imagenet Mini*¹, una versione ridotta di *ImageNet* disponibile su Kaggle. Questo dataset contiene un sottoinsieme di immagini prese dall'originale ImageNet, mantenendo le stesse 1.000 categorie ma riducendo il numero complessivo di immagini a una dimensione gestibile per esperimenti più rapidi. Sebbene sia più piccolo rispetto al dataset completo di ImageNet, ImageNet Mini offre comunque un banco di prova rappresentativo e sfidante per modelli complessi e avanzati come AlexNet, ResNet e VGG16, che sono stati addestrati sulla versione completa. Il dataset conserva la diversità delle classi originali, permettendo ai modelli di dimostrare le loro capacità di generalizzazione su un'ampia gamma di categorie visive. Grazie alla ridotta dimensione, ImageNet Mini consente di effettuare esperimenti con tempi di calcolo più brevi, pur mantenendo la possibilità di confrontare le prestazioni ottenute con i risultati di studi che utilizzano il dataset completo.

4.3 Soggetti dello Studio

I soggetti dello studio sono i modelli impiegati durante l'esperimento, selezionati per la loro rilevanza e impatto nel campo della visione artificiale. Per raggiungere gli obiettivi prefissati, sono stati considerati i tre modelli più comunemente utilizzati nei documenti analizzati durante la systematic literature review (cap. 3.2.3): *AlexNet*²,

¹*Dataset* *ImageNet* *Mini:* <https://www.kaggle.com/datasets/figotini/imagenetmini-1000>

²*Modello* *AlexNet:* <https://pytorch.org/vision/main/models/generated/torchvision.models.alexnet.html>

*ResNet18*³, e *VGG16*⁴.

AlexNet è noto per il suo impatto pionieristico nella classificazione delle immagini, avendo contribuito in modo significativo alla diffusione delle reti neurali profonde. La sua architettura, caratterizzata da una profondità maggiore rispetto ai modelli precedenti, rimane un punto di riferimento importante per gli sviluppi successivi nel campo. *ResNet*, con la sua innovativa struttura a reti residuali, è ampiamente impiegato per la sua capacità di mantenere alte prestazioni anche su dataset complessi e di grandi dimensioni, risolvendo il problema del vanishing gradient nelle reti molto profonde. Infine, *VGG16* è riconosciuto per la sua semplicità e l'efficacia nell'implementazione di architetture di rete profonda, con un design che enfatizza l'uso di piccole convoluzioni. Questi modelli, grazie alla loro solidità e diffusione, rappresentano una base ideale per valutare le tecniche di ottimizzazione e miglioramento delle prestazioni.

4.4 Variabili dello Studio

Nel contesto dello studio di benchmark, le variabili indipendenti e dipendenti rivestono un ruolo cruciale nell'analisi degli effetti delle tecniche di riduzione delle dimensioni dei modelli. Queste variabili sono riportate nella Tabella 4.2.

Variabili Indipendenti. Le variabili indipendenti includono il *modello baseline*, che rappresenta la versione originale del modello senza alcuna ottimizzazione, e il *modello ottimizzato*, sul quale sono state applicate tecniche di ridimensionamento. Questo permette di confrontare direttamente le prestazioni del modello prima e dopo l'ottimizzazione. In particolare, lo studio si concentra sulle seguenti tecniche:

- *TB - Baseline*: Il modello originale senza riduzione delle dimensioni.
- *TQ - Quantizzazione Dinamica*: Riduce la precisione numerica delle operazioni matematiche all'interno del modello per diminuirne la complessità computazionale e la memoria necessaria.

³Modello *ResNet18*: <https://pytorch.org/vision/2.0/models/generated/torchvision.models.resnet18.html>

⁴Modello *VGG16*: <https://pytorch.org/vision/main/models/generated/torchvision.models.vgg16.html>

Nome	Scala	Operazione
<i>Variabili indipendenti</i>		
Modello di baseline	Nominale	Modelli senza tecniche di ottimizzazione
Modello ottimizzato	Nominale	Modelli ottimizzati con low rank approximation, quantizzazione, pruning casuale non strutturato, pruning globale non strutturato, pruning non strutturato, pruning strutturato per canali
<i>Variabili dipendenti</i>		
Accuratezza	Rapporto	$(TP + TN) / (TP + TN + FP + FN)$
Precisione	Rapporto	$(TP) / (TP + FP)$
Recall	Rapporto	$(TP) / (TP + FN)$
AUC-ROC	Rapporto	$TPR = (TP) / (TP + FN)$ e $FPR = (FP) / (FP + TN)$
Tempo di Inferenza	Rapporto	Tempo totale per elaborare l'intero dataset
Throughput	Rapporto	$(\text{Numero di inferenze eseguite}) / (\text{tempo per eseguire le inferenze})$
Dimensione del Modello	Rapporto	La dimensione del modello
Consumo Energetico	Rapporto	Power Usage X Training Time

Tabella 4.2: Variabili dello Studio

- *TL - low rank Approximation*: Riduce la complessità del modello decomponendo le matrici di pesi in forme più semplici, mantenendo solo le componenti principali.
- *TPCNS - Pruning Casuale Non Strutturato*: Rimuove casualmente i pesi del modello senza tenere conto della struttura delle reti neurali.
- *TPGNS - Pruning Globale Non Strutturato*: Rimuove i pesi con minore rilevanza a livello globale, cioè considerando l'intera rete.
- *TPNS - Pruning Non Strutturato*: Elimina i pesi meno importanti in base alla loro magnitudine, riducendo la complessità del modello senza cambiare la struttura della rete.

- *TPSPC - Pruning Strutturato Per Canali*: Rimuove interi canali o filtri nella rete, riducendo non solo il numero di parametri ma anche la complessità computazionale.

Queste tecniche sono state scelte poiché risultano essere le più utilizzate, come evidenziato dai risultati della systematic literature review. Un'altra tecnica frequentemente utilizzata è la *knowledge distillation*, ma poiché non si dispone di un hardware sufficientemente potente per addestrare un modello studente, si è deciso di non trattarla in questo studio. Inoltre, ci siamo focalizzati sulle tecniche che possono essere implementate utilizzando *PyTorch*, al fine di garantire l'omogeneità dei risultati e semplificare l'implementazione.

Variabili Dipendenti Le variabili dipendenti sono influenzate dalle modifiche apportate ai modelli e includono metriche di performance e di sostenibilità ambientale. Lo studio utilizza le metriche maggiormente adottate in questo contesto, come evidenziato dalla systematic literature review, con l'eccezione della matrice di confusione, esclusa a causa dell'elevato numero di classi nel dataset Imagenet. Le metriche di performance incluse sono:

- *Accuratezza*: Percentuale di previsioni corrette sul totale delle previsioni effettuate.
- *Precisione*: Percentuale di vere positività rispetto al totale delle previsioni positive effettuate
- *Recall*: Percentuale di vere positività rispetto al totale delle istanze positive reali
- *AUC-ROC*: L'Area Under the Curve (AUC) della Receiver Operating Characteristic (ROC) rappresenta il trade-off tra il tasso di veri positivi e il tasso di falsi positivi.
- *Tempo di Inferenza*: Tempo necessario per eseguire l'intero processo di valutazione del modello.
- *Throughput*: Quantità di inferenze che il sistema può completare in un secondo.
- *Dimensione del modello*: Spazio di archiviazione necessario per salvare il modello.

La metrica di sostenibilità utilizzata è il *consumo energetico* misurato in kWh campionata durante il periodo di valutazione del modello. Questa metrica è strettamente dipendente dall'hardware e quantifica le risorse necessarie per utilizzare il modello. Per quantificare questa metrica è utilizzato lo strumento *CodeCarbon*⁵ che sfrutta strumenti di misurazione dell'energia. Tale tool fornisce la sua entità centrale *TrackerEmission* che cattura dati critici sul consumo energetico e sulle risorse. Queste informazioni vengono utilizzate dal tool per generare un report sulla sostenibilità ambientale in formato *.csv*.

4.5 Ipotesi Sperimentali, Esecuzione e Analisi

Dopo aver definito i soggetti e gli oggetti dello studio, sono state definite le ipotesi di lavoro che hanno consentito l'esecuzione dello studio di benchmark e la successiva analisi dei dati. In questa sezione sono riportati questi aspetti, descrivendo in dettaglio la logica e i metodi di ricerca impiegati per affrontare gli obiettivi dello studio.

4.5.1 Ipotesi Sperimentali

Dovendo analizzare come le variabili dipendenti sono influenzate a seconda delle tecniche di minimizzazione, sono stati definiti i seguenti elementi sperimentali. Sia μ_{B_i} e μ_{B_j} i modelli costruiti utilizzando le variabili indipendenti B_i e B_j , rispettivamente, dove $B_i, B_j \in \{Baseline, QuantizzazioneDinamica, LowRankApproximation, Pruning...\}$; sia S l'insieme delle variabili dipendenti dello studio.

Sia s_0 una variabile dipendente considerata nello studio. L'ipotesi nulla è la seguente:

$$H_0^{B_i, B_j, S_0} : \mu_{B_i}^{S_0} = \mu_{B_j}^{S_0} \forall i \neq j$$

$$S_0 \in \{Accuratezza, Precision, Recall, AUC - ROC, TempodiInferenza...\}$$

$$S_0 \in \{Throughput, DimensionedelModello, ConsumoEnergetico\}$$

La corrispondente ipotesi alternativa è la seguente:

$$H_\alpha^{B_i, B_j, S_0} : \mu_{B_i}^{S_0} \neq \mu_{B_j}^{S_0} \forall i \neq j$$

⁵Code Carbon: <https://codecarbon.io/>

$$S_0 \in \{Accuratezza, Precision, Recall, AUC - ROC, TempodiInferenza...\}$$

$$S_0 \in \{Throughput, DimensionedelModello, ConsumoEnergetico\}$$

4.5.2 Esecuzione dell'Esperimento

Per la valutazione delle prime 7 variabili dipendenti, è stata eseguita un'unica valutazione su tutto il dataset Imagenet Mini. Invece, per la variabile dipendente del consumo energetico, al fine di ottenere risultati più affidabili, sono stati selezionati casualmente 1000 campioni dal dataset di valutazione e su questi sono stati effettuati 100 cicli di valutazione.

Sono stati condotti 21 esperimenti che coinvolgono i tre modelli addestrati utilizzando ciascuna variabile indipendente dello studio, ovvero un modello di base che non includeva nessuna tecnica di minimizzazione della dimensione dei modelli, e i modelli addestrati utilizzando le varie tecniche di minimizzazione della dimensione dei modelli. Gli esperimenti sono stati condotti su una macchina con sistema operativo Linux, dotata di una CPU AMD EPYC v4 con 16 core e 64 GB di RAM. L'ambiente software della macchina era configurato per supportare Python 3.10.12, PyTorch 2.4.1, scikit-learn 2.5.2 e CodeCarbon 2.6.0. Prima dell'avvio di ciascuna sessione del benchmark, è stata prestata particolare attenzione a garantire che non vi fossero processi di background inutili in esecuzione, al fine di stabilizzare l'ambiente e ottenere misurazioni affidabili e ripetibili. Questo accorgimento ha consentito di registrare accuratamente i dati relativi al consumo energetico mediante l'uso di CodeCarbon. I modelli creati sono stati salvati come file .pth in modo da consentire ulteriori analisi in futuro. I risultati delle valutazioni sulle 7 variabili dipendenti sono salvati in file .txt, mentre quelli sul consumo energetico in file .csv.

4.6 Analisi dei Dati

Come ultima fase dello studio di benchmark, sono stati analizzati i dati provenienti dagli esperimenti eseguiti. I file .txt contenenti le valutazioni delle variabili indipendenti relative all'accuratezza e alle performance sono stati esaminati singolarmente, poiché in totale erano solo 21.

I file .csv relativi al consumo energetico sono stati invece analizzati statisticamente. Ogni modello minimizzato tramite tecniche di riduzione delle dimensioni è stato confrontato statisticamente con il modello di base, privo di minimizzazione. Per ciascuna distribuzione di dati, è stato eseguito il test di normalità di Shapiro-Wilk con un livello di significatività $\alpha = 0,05$, al fine di determinare se i dati sul consumo energetico seguissero una distribuzione normale. L'analisi ha mostrato che non tutti i dati seguivano una distribuzione normale e a seconda dei casi sono stati trattati con il test T o con il test Mann-Whitney U. Successivamente, sono stati creati dei grafici utilizzando la libreria Matplotlib per analizzare visivamente le distribuzioni dei dati.

4.7 Analisi dei Risultati

In questa sezione sono riportate le informazioni quantitative dalla fase di analisi dei dati. In particolare saranno analizzati i cambiamenti delle performance e dei consumi energetici in relazione alle tecniche utilizzate. L'obiettivo di questa analisi è andare a scoprire come cambiano le metriche di accuratezza e di performance dopo l'applicazione delle tecniche di minimizzazione della dimensione dei modelli di machine learning. Di seguito saranno analizzati singolarmente i modelli prima e dopo l'ottimizzazione per poi andare a effettuare una comparazione delle tecniche. In definitiva, l'analisi delle variabili dipendenti ha mostrato che le tecniche utilizzate hanno un impatto, pertanto, l'ipotesi nulla $H_0^{B_i, B_j, S_0}$ è stata respinta.

4.7.1 Analisi del Modello Alexnet

Analisi delle Performance e dell'Accuratezza

Nell'analisi delle prestazioni del modello AlexNet prima e dopo l'applicazione delle varie tecniche di ottimizzazione, emergono chiaramente i trade-off tra accuratezza, efficienza e riduzione della complessità del modello. Di seguito sono mostrati i risultati in tabelle e grafici, e sarà svolta una valutazione sui trade-off per ogni tecnica.

Modello Baseline TB:

Il modello baseline (Tabella 4.4) raggiunge un'accuratezza del 72%, con una precisione molto simile (73%) e un recall pari all'accuratezza (72%). L'AUC-ROC è elevato, raggiungendo il 99%, segno che il modello ha ottime capacità di distinguere tra le classi. Tuttavia, il tempo di inferenza è il più lungo tra tutti i modelli analizzati (170 secondi), con un throughput di 204 immagini/secondo. La dimensione del modello, 233MB, riflette la complessità del modello standard.

Modello Low Rank Approximation TL:

La low rank approximation (Tabella 4.6) porta a una riduzione minima del tempo di inferenza (161 secondi) e a un leggero aumento del throughput (216 immagini/secondo). Tuttavia, si nota un calo significativo nelle prestazioni in termini di accuratezza (65%), precisione (67%) e recall (65%). Questo evidenzia che questa tecnica, pur accelerando leggermente il processo di inferenza, comporta un peggioramento nella capacità del modello di fare previsioni accurate.

Variabile	Valore
Accuratezza	72%
Precisione	73%
Recall	72%
AUC-ROC	99%
Tempo di Inferenza	170s
Throughput	204i/s
Dimensione del Modello	233MB

Tabella 4.4: Performance AlexNet TB

Variabile	Valore
Accuratezza	65%
Precisione	67%
Recall	65%
AUC-ROC	99%
Tempo di Inferenza	161s
Throughput	216i/s
Dimensione del Modello	233MB

Tabella 4.6: Performance AlexNet TL

Modello Quantizzazione Dinamica TQ:

Il modello ottenuto tramite la quantizzazione dinamica (Tabella 4.8) riduce notevolmente la dimensione del modello, passando da 233MB a soli 65MB. L'accuratezza, la precisione, la recall e l'AUC-ROC rimangono invariate rispetto al modello baseline. Questo dimostra che la quantizzazione comunque permette di ottenere una buona compressione senza compromettere le prestazioni. Anche il tempo di inferenza migliora a 168 secondi, così come il throughput con 207 immagini/secondo.

Modello Pruning Casuale Non Strutturato TPCNS:

Il pruning casuale non strutturato (Tabella 4.10) comporta un calo dell'accuratezza al 64%, con una precisione del 65% e un recall del 64%, mantenendo comunque un AUC-ROC elevato (99%). Il tempo di inferenza (161 secondi) e il throughput (216.18 immagini/secondo) sono tra i migliori osservati, dimostrando che il pruning non strutturato riduce la complessità computazionale a scapito di una diminuzione delle prestazioni.

Variabile	Valore
Accuratezza	72%
Precisione	73%
Recall	72%
AUC-ROC	99%
Tempo di Inferenza	168s
Throughput	207i/s
Dimensione del Modello	65MB

Tabella 4.8: Performance TQ Alex net

Variabile	Valore
Accuratezza	64%
Precisione	65%
Recall	64%
AUC-ROC	99%
Tempo di Inferenza	161s
Throughput	216i/s
Dimensione del Modello	233MB

Tabella 4.10: Performance TPCNS Alex-net

Modello Pruning Globale Non Strutturato TPGNS:

Il modello ottenuto tramite pruning globale non strutturato (Tabella 4.12), similmente al pruning casuale non strutturato, comporta un calo dell'accuratezza al 65%, precisione al 66% e una recall al 65%, pur mantenendo un AUC-ROC al 99%. Anche qui, si osserva un miglioramento nel tempo di inferenza (161 secondi) e nel throughput (215 immagini/secondo), rendendolo una scelta valida per ridurre le complessità del modello ma a scapito di una diminuzione delle prestazioni.

Modello Pruning non Strutturato TPNS:

Il pruning non strutturato (Tabella 4.14) mostra risultati in linea con le altre tecniche di pruning, con una perdita di accuratezza (64%), precisione (66%), recall (64%). L'AUC-ROC rimane alto 99%, indicando che, nonostante la riduzione di parametri, il modello conserva buone capacità di classificazione. Anche qui, si nota un miglioramento delle prestazioni computazionali, con un tempo di inferenza ridotto (160 secondi) e un throughput aumentato (217 immagini/secondo).

Variabile	Valore
Accuratezza	65%
Precisione	66%
Recall	65%
AUC-ROC	99%
Tempo di Inferenza	161s
Throughput	215i/s
Dimensione del Modello	233MB

Tabella 4.12: Performance TPGNS Alexnet

Variabile	Valore
Accuratezza	64%
Precisione	66%
Recall	64%
AUC-ROC	99%
Tempo di Inferenza	160s
Throughput	217i/s
Dimensione del Modello	233MB

Tabella 4.14: Performance TPNS Alexnet

Modello Pruning Strutturato Per Canali TPSPC:

Il pruning strutturato per canali (Tabella 4.16) è la tecnica che causa il maggior deterioramento nelle metriche di prestazione. L'accuratezza scende a 63%, con una precisione di 65% e un recall di 63%. L'AUC-ROC resta invariato al 99%. Il tempo di inferenza si riduce leggermente rispetto alla baseline (168 secondi), ma il throughput rimane più basso rispetto agli altri metodi di pruning (207 immagini/secondo). Questa tecnica, quindi, sembra avere un impatto più negativo sulla capacità del modello di fare previsioni accurate.

Variabile	Valore
Accuratezza	63%
Precisione	65%
Recall	63%
AUC-ROC	99%
Tempo di Inferenza	168s
Throughput	207i/s
Dimensione del Modello	233MB

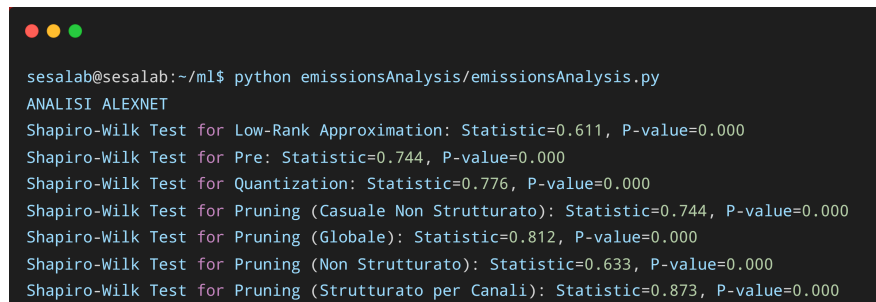
Tabella 4.16: Performance TPSPC Alexnet

In generale, la quantizzazione dinamica emerge come la tecnica più efficace per ridurre la dimensione del modello senza compromettere le prestazioni. Il pruning non strutturato, sia globale che casuale, comporta una lieve riduzione dell'accuratezza, ma consente una maggiore efficienza computazionale. Al contrario, il pruning strutturato per canali provoca un calo più marcato delle prestazioni, specialmente in termini di accuratezza. Infine, anche la low rank approximation si è dimostrata meno efficace, con un peggioramento significativo delle prestazioni rispetto alle altre tecniche.

Analisi dei Consumi Energetici

Per valutare l'impatto delle tecniche di minimizzazione delle dimensioni del modello sul consumo energetico di AlexNet, sono stati eseguiti test statistici per ciascuna tecnica di ottimizzazione. Inizialmente è stato eseguito il test di Shapiro-Wilk che è stato applicato per verificare la normalità delle distribuzioni dei dati, successivamente il test T è stato utilizzato per confrontare il consumo energetico di ciascuna tecnica con il modello prima dell'ottimizzazione.

I risultati del test di Shapiro-Wilk, mostrati nella figura 4.1 hanno mostrato che, per tutte le tecniche applicate, le distribuzioni del consumo energetico non seguono una distribuzione normale ($P\text{-value} < 0.05$ in tutti i casi). In particolare, le statistiche più basse sono state osservate per il pruning non strutturato (Statistic=0.633), seguito dalla low rank approximation (Statistic=0.611), evidenziando come queste due tecniche abbiano introdotto una maggiore variabilità nei dati energetici rispetto al modello pre-ottimizzato (Statistic=0.744).



```
sesalab@sesalab:~/ml$ python emissionsAnalysis/emissionsAnalysis.py
ANALISI ALEXNET
Shapiro-Wilk Test for Low-Rank Approximation: Statistic=0.611, P-value=0.000
Shapiro-Wilk Test for Pre: Statistic=0.744, P-value=0.000
Shapiro-Wilk Test for Quantization: Statistic=0.776, P-value=0.000
Shapiro-Wilk Test for Pruning (Casuale Non Strutturato): Statistic=0.744, P-value=0.000
Shapiro-Wilk Test for Pruning (Globale): Statistic=0.812, P-value=0.000
Shapiro-Wilk Test for Pruning (Non Strutturato): Statistic=0.633, P-value=0.000
Shapiro-Wilk Test for Pruning (Strutturato per Canali): Statistic=0.873, P-value=0.000
```

Figura 4.1: Test Shapiro-Wilk per Alexnet

I confronti diretti con il modello pre-ottimizzato attraverso il test T, come mostrato nella figura 4.2, hanno confermato differenze significative nel consumo energetico per tutte le tecniche. La low rank approximation e la quantizzazione hanno mostrato una differenza altamente significativa rispetto al modello pre-ottimizzato (T-statistic=1002.000, P-value=0.000 alla primo confronto e T-statistic=838.000, P-value=0.000 al secondo confronto).

Per quanto riguarda le varie forme di pruning, il pruning casuale non strutturato (T-statistic=817.000, P-value=0.000) ha introdotto una variabilità maggiore (Statistic=0.744). Il pruning strutturato per canali e il pruning non strutturato hanno ottenuto performance peggiori in termini di riduzione del consumo energetico, ri-

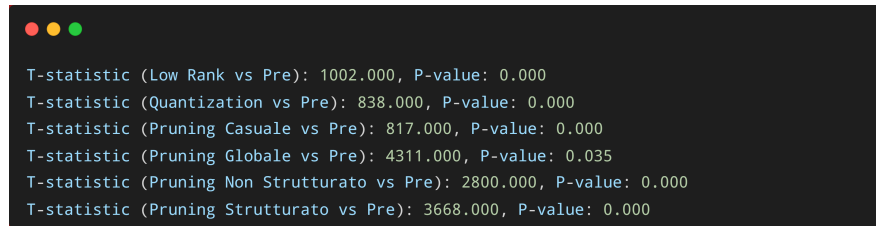


Figura 4.2: Test T per Alexnet

spettivamente $\text{Statistic}=0.873$, $\text{T-statistic}=3668.000$, $\text{P-value}=0.000$ per la prima tecnica e $\text{Statistic}=0.633$, $\text{T-statistic}=2800.000$, $\text{P-value}=0.000$ per la seconda. Infine, il pruning globale ha prodotto una riduzione meno pronunciata, con una P-value appena inferiore al 5% ($\text{T-statistic}=4311.000$, $\text{P-value}=0.035$), suggerendo che, sebbene efficace, potrebbe non essere la tecnica più efficiente in termini di riduzione energetica rispetto ad altre varianti.

I box plot aiutano l'elaborazione dei risultati precedentemente discussi. Nel grafico 4.3, si può osservare che il pruning casuale non strutturato (TPCNS) e il pruning non strutturato (TPNS) hanno ridotto in modo visibile il consumo energetico rispetto al modello originale (TB), con medie più basse e una minore dispersione. D'altro canto, il pruning globale non strutturato (TPGNS) e strutturato per canali (TPSPC) hanno evidenziato una riduzione meno marcata del consumo.

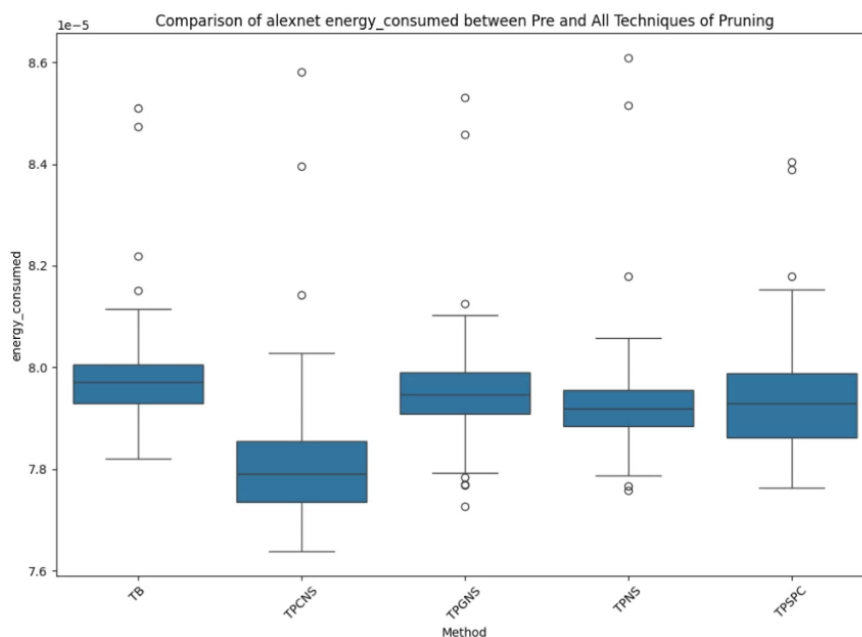


Figura 4.3: Grafico dei Consumi Energetici di AlexNet Pre e Pruning

Il grafico 4.4 mostra il confronto tra il modello baseline, la quantizzazione e la low rank approximation. Entrambe le tecniche abbassano significativamente il consumo energetico, con una distribuzione molto più compatta e con meno outlier rispetto al modello originale. La low rank approximation (TL) ha avuto una riduzione particolarmente notevole, con una media del consumo energetico inferiore rispetto a tutte le altre tecniche, rendendola una delle soluzioni più efficienti energeticamente.

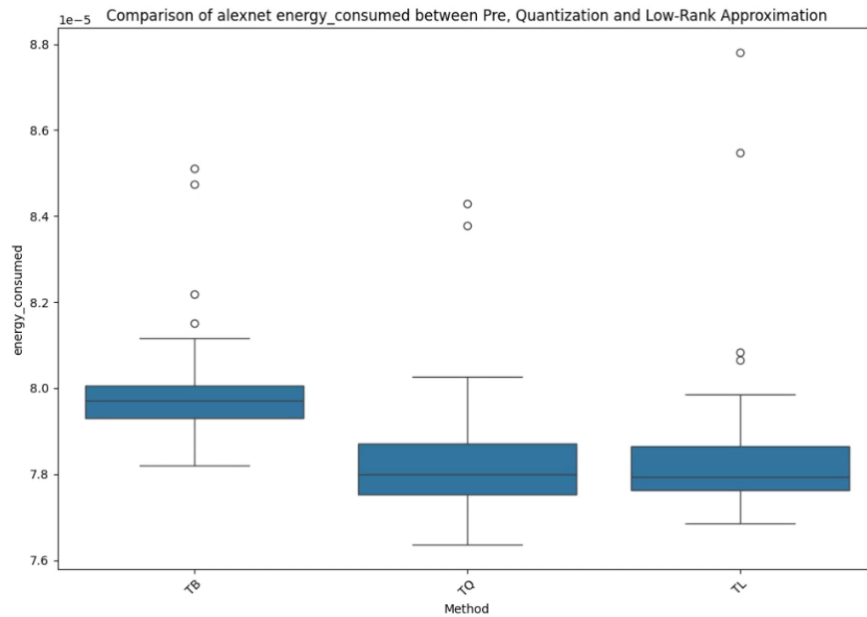


Figura 4.4: Grafico dei Consumi Energetici di AlexNet Pre, Quantizzazione e low rank Approximation

In sintesi, tutte le tecniche di minimizzazione applicate su AlexNet hanno portato a riduzioni significative del consumo energetico. Tuttavia, la low rank approximation e la quantizzazione sono risultate le più efficaci in termini di riduzione complessiva, mentre il pruning non strutturato si è dimostrato la tecnica di pruning più promettente. Questi risultati evidenziano i diversi trade-off tra le tecniche e le opportunità per un'ottimizzazione più efficiente dal punto di vista energetico.

4.7.2 Performance Sustainability Trade-Off per AlexNet

L'analisi del trade-off tra performance e sostenibilità energetica delle tecniche di ottimizzazione applicate su AlexNet rileva risultati interessanti, evidenziando come

alcune tecniche risultano più efficaci nella riduzione del consumo energetico, mentre altre riescano a mantenere meglio le prestazioni di accuratezza e performance.

Il modello AlexNet baseline pur avendo performance di accuratezza elevate e un buon tempo di inferenza, mostra un consumo energetico significativamente più elevato rispetto alle tecniche di ottimizzazione. Questo modello funge da punto di riferimento per valutare l'efficacia delle ottimizzazioni energetiche.

La low rank approximation emerge come la tecnica più efficace in termini di riduzione del consumo energetico. Nonostante una riduzione delle performance di accuratezza e precisione, riesce a mantenere un bilancio accettabile tra prestazioni e risparmio energetico. È interessante notare che questa tecnica introduce una maggiore variabilità nel consumo energetico, tuttavia grazie alla riduzione dei consumi si dimostra particolarmente vantaggioso dal punto di vista della sostenibilità.

La quantizzazione si dimostra estremamente vantaggiosa dal punto di vista energetico. Ciò avviene senza una riduzione sostanziale delle metriche di accuratezza e di performance. Inoltre, un punto focale, è la riduzione della dimensione del modello, rendendola una tecnica per questo modello forte dal punto di vista energetico, di memoria e di capacità di archiviazione. Quindi la quantizzazione rappresenta una delle migliori tecniche per mantenere prestazioni elevate riducendo al contempo il consumo energetico.

Per quanto riguarda le tecniche di pruning, il pruning casuale non strutturato mostra una riduzione significativa del consumo energetico, con una media di consumo più bassa rispetto al modello baseline. Tuttavia, questa tecnica riduce le performance. Sebbene sia energeticamente vantaggioso, non rappresenta un compromesso significativo in termini di performance. Il pruning globale non strutturato, pur portando una riduzione del consumo energetico meno pronunciata, riesce a mantenere delle performance un po' migliori rispetto ad altre tecniche di pruning. Questo fa sì che rappresenti un compromesso più bilanciato tra riduzione energetica e performance, anche se non è la tecnica più efficace in termini di ottimizzazione energetica pura. Il pruning non strutturato e strutturato per canali hanno ottenuto entrambi delle performance più basse in termini di riduzione del consumo energetico e le loro performance non sono particolarmente rilevanti.

In sintesi, il trade-off tra performance e sostenibilità energetica per AlexNet varia

notevolmente. La quantizzazione si distingue come tecniche più efficace per ridurre il consumo energetico, mantenendo metriche di performance relativamente alte. La low rank approximation mostra buoni risultati in termini di consumo energetico, ma meno vantaggiosi sull'accuratezza. Il pruning casuale non strutturato riduce notevolmente il consumo energetico, ma a costo di una variabilità maggiore e prestazioni ridotte. Le altre tre tecniche del pruning invece, non mostrano risultati importanti nelle riduzioni del consumo energetico e anche in termini di accuratezza e performance.

4.7.3 Analisi del Modello Resnet18

Analisi delle Performance e dell'Accuratezza

L'analisi delle prestazioni del modello ResNet18 prima e dopo l'applicazione di tecniche di ottimizzazione evidenzia differenza significative in termini di accuratezza, tempi di inferenza e throughput, a seconda della strategia utilizzata. Di seguito sono riportati i risultati in tabelle.

Modello Baseline TB:

Il modello ResNet18 (Tabella 4.18) originale ha un accuratezza del 78%, con una precisione del 79% e un recall allineato con l'accuratezza (78%). L'AUC-ROC è molto elevato (99%), tuttavia, il tempo di inferenza è notevole con 319 secondi e un throughput relativamente basso pari a 108 immagini/secondo, riflettendo la complessità del modello. La dimensione del modello, pari a 45MB, è relativamente contenuta rispetto ad altre reti più grandi.

Modello Low Rank Approximation TL:

La low rank approximation (Tabella 4.20) non produce alcuna variazione nelle metriche di prestazione rispetto al modello baseline. Tuttavia, si osserva un lieve peggioramento nel tempo di inferenza, che passa a 325 secondi, con un throughput ridotto a 107 immagini/secondo. Questo implica che la tecnica, pur non compromet-

tendo l'accuratezza, non porta miglioramenti in termini di efficienza computazionale e addirittura peggiora leggermente la velocità.

Variabile	Valore
Accuratezza	78%
Precisione	79%
Recall	78%
AUC-ROC	99%
Tempo di Inferenza	319s
Throughput	108i/s
Dimensione del Modello	45MB

Tabella 4.18: Performance TB ResNet18

Variabile	Valore
Accuratezza	78%
Precisione	79%
Recall	78%
AUC-ROC	99%
Tempo di Inferenza	325s
Throughput	107i/s
Dimensione del Modello	45MB

Tabella 4.20: Performance TL ResNet18

Modello Quantizzazione TQ:

La quantizzazione (Tabella 4.22) si conferma come tecnica efficace per ridurre la dimensione del modello (da 45MB a 43MB), mantenendo invariati i risultati di accuratezza, precisione, recall e AUC-ROC. Tuttavia, il tempo di inferenza aumenta leggermente (321 secondi) rispetto al modello originale, e il throughput rimane invariato (108 immagini/secondo). Ciò dimostra che la quantizzazione è utile principalmente per ridurre lo spazio di memoria senza impattare negativamente le prestazioni.

Modello Pruning Casuale Non Strutturato TPCNS:

Il pruning casuale non strutturato (Tabella 4.24) comporta una riduzione significativa dell'accuratezza, che scende al 71%. Anche la precisione diminuisce a 74%, e il recall è allineato all'accuratezza. Sebbene l'AUC-ROC rimanga elevato (99%), il tempo di inferenza peggiora (333 secondi) e il throughput si riduce a 104 immagini/secondo. Questi risultati indicano che il pruning casuale non strutturato

può ridurre la complessità del modello ma al costo di una consistente perdita di accuratezza e una diminuzione dell'efficienza computazionale.

Variabile	Valore
Accuratezza	78%
Precisione	79%
Recall	78%
AUC-ROC	99%
Tempo di Inferenza	321s
Throughput	1087i/s
Dimensione del Modello	43MB

Tabella 4.22: Performance TQ ResNet18

Variabile	Valore
Accuratezza	71%
Precisione	74%
Recall	71%
AUC-ROC	99%
Tempo di Inferenza	333s
Throughput	104i/s
Dimensione del Modello	45MB

Tabella 4.24: Performance TPCNS ResNet18

Modello Pruning Globale Non Strutturato TPGNS:

Il pruning globale (Tabella 4.26) mantiene l'accuratezza (78%), la precisione (79%) e la recall (78%) inalterata rispetto al modello originale. Tuttavia, si nota un aumento significativo nel tempo di inferenza (350 secondi), accompagnato da un calo del throughput a 99 immagini/secondo. Questo peggioramento delle performance computazionali indica che il pruning globale, pur mantenendo una buona accuratezza, ha un impatto negativo sull'efficienza computazionale.

Modello Pruning Non Strutturato TPNS:

Il pruning non strutturato (Tabella 4.28) mostra un impatto meno pronunciato rispetto al pruning casuale. L'accuratezza si riduce solo leggermente, scendendo a 77%, con precisione al 78% e recall al 77% in linea con le aspettative. L'AUC-ROC rimane identico a quello del modello originale (99%) mentre il tempo di inferenza è uguale a quello della low rank approximation (325 secondi), con un throughput

leggermente ridotto (107 immagini/secondo). Questo tipo di pruning sembra un compromesso accettabile tra accuratezza e riduzione della complessità.

Variabile	Valore
Accuratezza	78%
Precisione	79%
Recall	78%
AUC-ROC	99%
Tempo di Inferenza	350s
Throughput	99i/s
Dimensione del Modello	45MB

Tabella 4.26: Performance TPGNS Re-
sNet18

Variabile	Valore
Accuratezza	77%
Precisione	78%
Recall	77%
AUC-ROC	99%
Tempo di Inferenza	325s
Throughput	107i/s
Dimensione del Modello	45MB

Tabella 4.28: Performance TPNS Re-
sNet18

Modello Pruning Strutturato Per Canali TPSPC:

Il pruning strutturato per canali (Tabella 4.30) causa un calo più pronunciato dell'accuratezza (73%), precisione (77%) e recall (73%), rispetto al modello baseline. Anche se l'AUC-ROC rimane identico (99%), il tempo di inferenza (328 secondi) è maggiore rispetto alla baseline, e il throughput scende a 106 immagini/secondo. Questa tecnica risulta quindi meno efficace, soprattutto in termini di accuratezza, pur non apportando vantaggi significativi nei tempi di inferenza.

Variabile	Valore
Accuratezza	73%
Precisione	77%
Recall	73%
AUC-ROC	99%
Tempo di Inferenza	328s
Throughput	106i/s
Dimensione del Modello	45MB

Tabella 4.30: Performance TPSPC ResNet18

In sintesi, la quantizzazione rappresenta la tecnica più vantaggiosa per ridurre le dimensioni del modello senza influire negativamente sulle prestazioni. La low rank approximation, invece, ha risultati molto simili al modello baseline. Il pruning casuale non strutturato ha un impatto negativo sulle prestazioni in termini di accuratezza e tempo di inferenza, mentre il pruning globale non strutturato offre una buona accuratezza ma peggiora drasticamente l'efficienza computazionale. Il pruning non strutturato si posiziona come un'opzione intermedia, mentre il pruning strutturato per canali è la tecnica che produce la maggiore riduzione di accuratezza, con un impatto limitato sull'efficienza computazionale.

Analisi dei Consumi Energetici

L'analisi del consumo energetico per ResNet18 ha prodotto risultati significativi. I risultati del test di Shapiro-Wilk, riportati nella figura 4.5, hanno mostrato che, per tutte le tecniche applicate, le distribuzioni dei consumi non seguono una distribuzione normale in quanto il P-value è inferiore a 0.05 in tutti i casi. Il pruning strutturato per canali ha evidenziato la deviazione più marcata della normalità (Statistic=0.497), seguito dal pruning non strutturato (Statistic=0.796), segnalando che queste tecniche introducono una variabilità maggiore nei dati rispetto al modello di baseline (Statistic=0.925).

```

sesalab@sesalab:~/ml$ python emissionsAnalysis/emissionsAnalysis.py
ANALISI RESNET18
Shapiro-Wilk Test for Low-Rank Approximation: Statistic=0.748, P-value=0.000
Shapiro-Wilk Test for Pre: Statistic=0.925, P-value=0.000
Shapiro-Wilk Test for Quantization: Statistic=0.823, P-value=0.000
Shapiro-Wilk Test for Pruning (Casuale Non Strutturato): Statistic=0.919, P-value=0.000
Shapiro-Wilk Test for Pruning (Globale): Statistic=0.826, P-value=0.000
Shapiro-Wilk Test for Pruning (Non Strutturato): Statistic=0.796, P-value=0.000
Shapiro-Wilk Test for Pruning (Strutturato per Canali): Statistic=0.497, P-value=0.000

```

Figura 4.5: Test Shapiro-Wilk per ResNet18

I confronti eseguiti con il test T, illustrati nella figura 4.6, hanno confermato differenze significative nel consumo energetico per quasi tutte le tecniche rispetto alla baseline. In particolare, la low rank approximation (TL) e il pruning casuale non strutturato (TPCNS) hanno prodotto i valori più elevati di T-statistic, rispettivamente 8716.000 e 8724.000, entrambi con P-value=0.000, evidenziando una marcata differenza rispetto al modello baseline. A seguire, il pruning non strutturato (TPNS) ha prodotto dei valori pari a 6172.00 per la T-statistic e P-value=0.021, mentre il pruning strutturato epr canali (TPSPC) ha prodooto dei valori pari a 5424.000 per la T-statistic e P-value==0.599. Infine migliorano i valori per quanto riguarda la quantizzazione con T-statistic=1774.000 e P-value=0.000.

```

T-statistic (Low Rank vs Pre): 8716.000, P-value: 0.000
T-statistic (Quantization vs Pre): 1774.000, P-value: 0.000
T-statistic (Pruning Casuale vs Pre): 8724.000, P-value: 0.000
T-statistic (Pruning Globale vs Pre): 7126.000, P-value: 0.000
T-statistic (Pruning Non Strutturato vs Pre): 6172.000, P-value: 0.021
T-statistic (Pruning Strutturato vs Pre): 5424.000, P-value: 0.599

```

Figura 4.6: Test T per ResNet18

Dal grafico 4.7, si nota che tutte le tecniche di minimizzazione del pruning comportano un maggiore consumo di energia con molti valori outlier, tranne il pruning strutturato per canali (TPSPC), che mostra un leggerissimo miglioramento.

Un netto abbassamento dei consumi, come mostrato nel grafico 4.8, può essere raggiunto attraverso la quantizzazione. D'altra parte, la low rank approximation non offre miglioramenti nel consumo energetico.

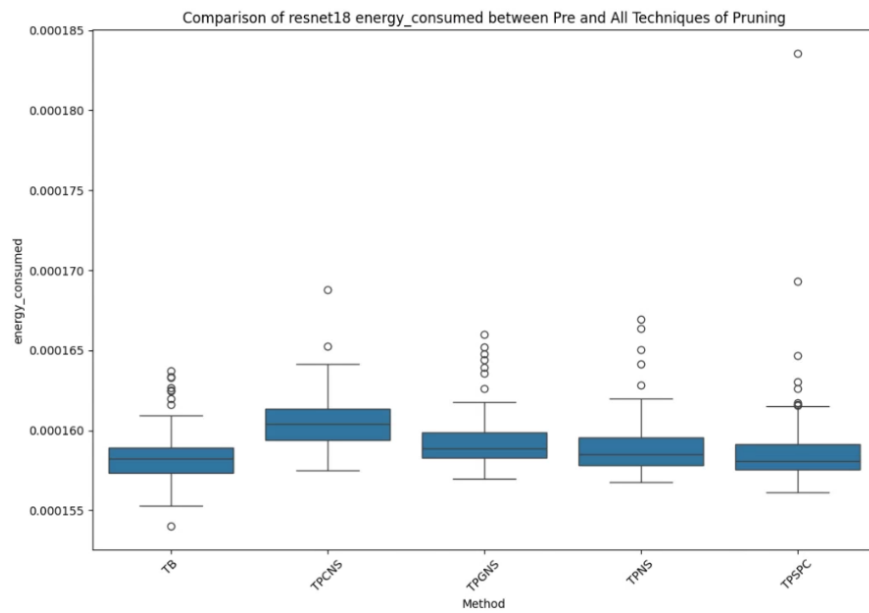


Figura 4.7: Grafico dei Consumi Energetici di ResNet18 Pre e Pruning

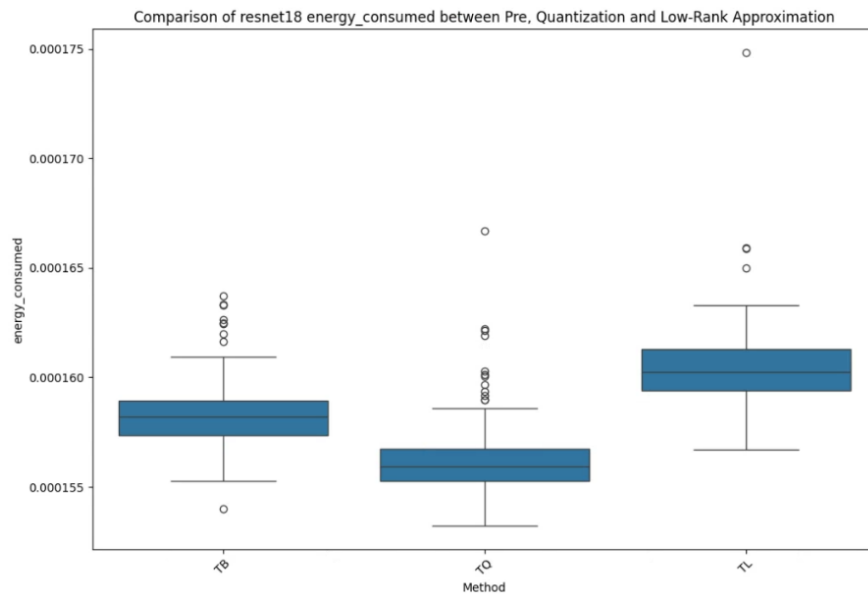


Figura 4.8: Grafico dei Consumi Energetici di ResNet18 Pre, Quantizzazione e low rank Approximation

4.7.4 Performance Sustainability Trade-Off per ResNet18

L'analisi dei trade-off tra performance e sostenibilità energetica per ResNet18 evidenzia come le diverse tecniche di ottimizzazione abbiano un impatto variabile, sia in termini di consumo energetico sia di prestazioni.

Il modello ResNet18 baseline funge da base per valutare l'efficacia delle tec-

niche di ottimizzazione. Esso presenta un buon bilanciamento tra accuratezza e tempo di inferenza, ma come evidenziato dall'analisi dei consumi energetici, richiede ottimizzazioni per ridurre l'impatto energetico.

La low rank approximation ha prodotto risultati contrastanti. Da un lato, mantiene le stesse metriche di accuratezza, precisione e recall del modello originale, e anche il throughput è comparabile. Tuttavia, dal punto di vista del consumo energetico, i risultati non sono incoraggianti. Infatti non rappresenta una soluzione sostenibile dal punto di vista energetico per ResNet18, nonostante mantenga buone prestazioni in termini di accuratezza e precisione.

La quantizzazione, invece, si distingue come una delle tecniche più efficaci nel ridurre il consumo energetico rispetto al modello baseline. Questo, combinato con il fatto che le performance di accuratezza e precisione rimangono praticamente invariate, rende la quantizzazione una soluzione altamente efficace per migliorare la sostenibilità del modello.

Il pruning casuale non strutturato ha mostrato un aumento significativo sul consumo energetico. Mentre, Dal punto di vista delle prestazioni, mostra un calo sulle metriche di accuratezza. Questo rende questa tecnica meno desiderabile se l'obiettivo è mantenere i consumi energetici bassi e alte le performance.

Il pruning strutturato per canali si distingue come la tecnica che ha mostrato un lieve miglioramento energetico più significativo rispetto ad altre tecniche di pruning. Tuttavia, il calo delle performance è comunque presente, indicando un compromesso tra la riduzione del consumo energetico e la qualità delle prestazioni.

Il pruning non strutturato e quello globale hanno mostrato un aumento poco marcato del consumo energetici, seppur mantengono prestazioni simili al modello baseline. Questo le rende due tecniche meno invitanti a causa dei consumi.

In conclusione, l'analisi dei trade-off tra performance e sostenibilità energetica per ResNet18 mostra che non tutte le tecniche di ottimizzazione portano a risultati simili. Le tecniche di quantizzazione e pruning strutturato per canali si distinguono come le più promettenti per ridurre il consumo energetico mantenendo un buon livello di accuratezza e precisione. La low rank approximation, pur mantenendo alte prestazioni, non offre miglioramenti energetici significativi e può introdurre una maggiore variabilità nei consumi. Questi risultati sottolineano l'importanza di

scegliere la tecnica di ottimizzazione più adatta in base agli obiettivi specifici: se l'accuratezza è prioritaria, il pruning globale o la quantizzazione sono le scelte migliori; se si desidera ridurre drasticamente il consumo energetico, la quantizzazione offre un bilanciamento ottimale tra sostenibilità e performance.

4.7.5 Analisi del Modello VGG16

Analisi delle Performance e dell'Accuratezza

L'analisi delle prestazioni del modello VGG16 prima e dopo l'ottimizzazione evidenzia notevoli differenze tra le metriche di accuratezza, tempi di inferenza e dimensione del modello. VGG16, con la sua elevata complessità, evidenzia in modo chiaro come le tecniche di compressione e pruning influiscano sulla sua capacità di generalizzazione.

Modello Baseline TB:

Il modello originale VGG16 (Tabella 4.32) ottiene un'accuratezza del 83%, una precisione dell'84% e una recall pari all'84%. L'AUC-ROC è pari al 99%, tuttavia il tempo di inferenza è estremamente elevato (1309 secondi), con un throughput molto basso di 27 immagini/secondo. La dimensione del modello, pari a 527.79MB è significativamente più grande rispetto ad altri modelli analizzati come ResNet18 e AlexNet.

Modello Low Rank Approximation TL:

La low rank approximation (Tabella 4.34) porta a una riduzione delle prestazioni del modello rispetto alla baseline in quanto l'accuratezza scende all'80%, la precisione all'81% e il recall all'80%, pur mantenendo un AUC-ROC pari al 99%. Il tempo di inferenza diminuisce leggermente a 1260 secondi con un throughput migliorato (28 immagini/secondo). Questa tecnica introduce una lieve riduzione dell'accuratezza del modello, rendendola meno efficace rispetto alla quantizzazione.

Variabile	Valore
Accuratezza	83%
Precisione	84%
Recall	83%
AUC-ROC	99%
Tempo di Inferenza	1309s
Throughput	27i/s
Dimensione del Modello	528MB

Tabella 4.32: Performance TB VGG16

Variabile	Valore
Accuratezza	80%
Precisione	81%
Recall	80%
AUC-ROC	99%
Tempo di Inferenza	1260s
Throughput	28i/s
Dimensione del Modello	528MB

Tabella 4.34: Performance TL VGG16***Modello Quantizzazione TQ:***

La quantizzazione (Tabella 4.36) riduce significativamente, con il 66%, la riduzione della dimensione del modello passando da 528MB a 174MB. Nonostante questa grande compressione, l'accuratezza (83%), la precisione (84%) e il recall (83%) rimangono invariati rispetto al modello originale, così come l'AUC-ROC (99%). Anche il tempo di inferenza migliora leggermente (1276 secondi), con un throughput aumentato leggermente a 27 immagini/secondo. La quantizzazione dimostra nuovamente la sua capacità a ridurre le dimensioni del modello mantenendo intatte le prestazioni di classificazione.

Modello Pruning Casuale Non Strutturato:

Il pruning casuale non strutturato (Tabella 4.38) causa una drastica riduzione delle prestazioni di accuratezza (54%), precisione (64%) e recall (54%). L'AUC-ROC invece resta sempre al 99%. Anche se il tempo di inferenza (1267 secondi) e il throughput (27 immagini/secondo) migliorano leggermente rispetto al modello di partenza, le prestazioni di classificazione vengono gravemente compromesse. Questo evidenzia come il pruning casuale non strutturato possa ridurre la complessità del modello ma con un impatto molto negativo sulle prestazioni di generalizzazione.

Variabile	Valore
Accuratezza	83%
Precisione	84%
Recall	83%
AUC-ROC	99%
Tempo di Inferenza	1276s
Throughput	27i/s
Dimensione del Modello	174MB

Tabella 4.36: Performance TQ VGG16

Variabile	Valore
Accuratezza	54%
Precisione	64%
Recall	54%
AUC-ROC	99%
Tempo di Inferenza	1267s
Throughput	27i/s
Dimensione del Modello	528MB

Tabella 4.38: Performance TPCNS VGG16***Modello Pruning Globale Non Strutturato:***

Il pruning non strutturato (Tabella 4.40) produce una riduzione dell'accuratezza (80%), precisione (80%) e un recall dell'80%. L'AUC-ROC rimane lo stesso al 99%, mentre il tempo di inferenza diminuisce leggermente rispetto al modello originale (1274 secondi) e il throughput rimane stabile (27 immagini al secondo). Questa tecnica mantiene buone prestazioni senza compromettere troppo l'efficienza computazionale.

Modello Pruning Non Strutturato:

Il pruning non strutturato (Tabella 4.42) produce una riduzione dell'accuratezza (79%), precisione (80%) e recall (79%), simili al pruning globale, e un AUC-ROC invariato (99%). Il tempo di inferenza rimane stabile (1267 secondi), così come il throughput pari a 27 immagini/secondo. Questa tecnica sembra fornire una via di mezzo tra accuratezza e complessità computazionale, riducendo leggermente le prestazioni senza perdere molto in efficienza.

Variabile	Valore
Accuratezza	80%
Precisione	80%
Recall	79%
AUC-ROC	99%
Tempo di Inferenza	1274s
Throughput	27i/s
Dimensione del Modello	528MB

Tabella 4.40: Performance TPGNS
VGG16

Variabile	Valore
Accuratezza	79%
Precisione	80%
Recall	79%
AUC-ROC	99%
Tempo di Inferenza	1267s
Throughput	27i/s
Dimensione del Modello	528MB

Tabella 4.42: Performance TPNS VGG16

Modello Pruning Strutturato Per Canali:

Il pruning strutturato per canali (Tabella 4.44) si rivela una delle tecniche meno efficaci per VGG16. L'accuratezza scende drasticamente a 53%, con precisione e recall che si abbassano rispettivamente a 64% e 53%. L'AUC-ROC resta sempre al 99%, anche se il tempo di inferenza è leggermente migliore rispetto al modello baseline (1271 secondi), il throughput (27 immagini/secondo) resta uguale. Questo metodo compromette fortemente le prestazioni del modello, suggerendo che il pruning strutturato per canali non è adatto a modelli complessi come VGG16.

Variabile	Valore
Accuratezza	53%
Precisione	64%
Recall	53%
AUC-ROC	99%
Tempo di Inferenza	1271s
Throughput	27i/s
Dimensione del Modello	528MB

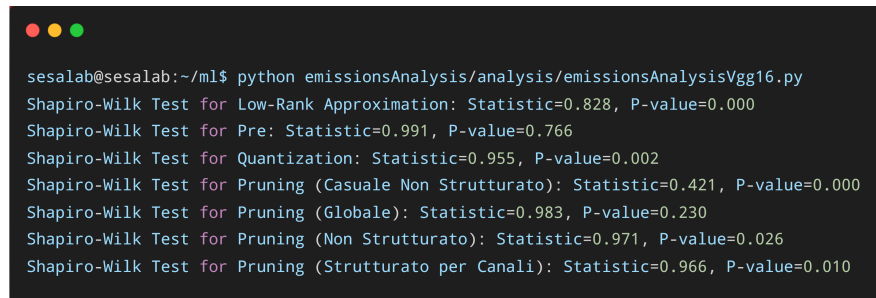
Tabella 4.44: Performance TPSPC VGG16

Nel complesso, la quantizzazione si conferma nuovamente la tecnica migliore per ridurre la dimensione del modello senza intaccare in maniera significativa le prestazioni del modello, mentre la low rank approximation offre un miglioramento sull'efficienza computazionale a costo di un calo delle metriche di accuratezza. Il pruning globale non strutturato e il pruning non strutturato si comportano in modo simile, con un compromesso accettabile tra accuratezza e riduzione della complessità del modello. Al contrario il pruning strutturato per canali e quello casuale non strutturato causano una grave perdita delle prestazioni.

Analisi dei Consumi Energetici

L'analisi dei consumi energetici del modello VGG16, eseguita attraverso test statistici, ha fornito risultati significativi riguardo l'impatto delle tecniche di minimizzazione della dimensione del modello. Per prima cosa è stato applicato il test di Shapiro-Wilk per valutare la normalità dei dati, i cui risultati sono mostrati nella figura 4.9. Il test ha rilevato che i consumi energetici prima dell'applicazione di qualsiasi tecnica di minimizzazione seguono una distribuzione normale (Statistic=0.991, P-value=0.766), suggerendo che le misurazioni pre-minimizzazione sono adeguate per analisi parametriche. Anche il test eseguito sui dati relativi al consumo del modello ottimizzato con pruning globale non strutturato (TPGNS) presentano una

normalità (T-statistic=0.983 e P-value=0.230). Al contrario, le analisi su low rank approximation hanno dimostrato una distribuzione non normale (T-statistic=0.828, P-value=0.000), indicando che l'utilizzo di questa tecnica altera in modo sostanziale la distribuzione dei consumi. Similmente, i risultati relativi alla quantizzazione (TQ) mostrano una certa non normalità (Statistic=0.955, P-value=0.002), suggerendo un effetto marcato sui consumi energetici. Per quanto riguarda le tecniche di pruning, le analisi hanno mostrato che il pruning casuale non strutturato (TPCNS) ha una distribuzione fortemente non normale (Statistic=0.421, P-value=0.000), mentre le altre forme di pruning, come quello non strutturato (Statistic=0.971, P-value=0.026) e strutturato per canali (TPSPC) (Statistic=0.966, P-value=0.010), hanno anch'esse mostrato una deviazione dalla normalità, seppur meno pronunciata.



```
sesalab@sesalab:~/ml$ python emissionsAnalysis/analysis/emissionsAnalysisVgg16.py
Shapiro-Wilk Test for Low-Rank Approximation: Statistic=0.828, P-value=0.000
Shapiro-Wilk Test for Pre: Statistic=0.991, P-value=0.766
Shapiro-Wilk Test for Quantization: Statistic=0.955, P-value=0.002
Shapiro-Wilk Test for Pruning (Casuale Non Strutturato): Statistic=0.421, P-value=0.000
Shapiro-Wilk Test for Pruning (Globale): Statistic=0.983, P-value=0.230
Shapiro-Wilk Test for Pruning (Non Strutturato): Statistic=0.971, P-value=0.026
Shapiro-Wilk Test for Pruning (Strutturato per Canali): Statistic=0.966, P-value=0.010
```

Figura 4.9: Test Shapiro-Wilk per VGG16

Per eseguire il confronto tra le varie tecniche è stata utilizzata la tecnica di Mann-Whitney, a eccezione del test T per l'analisi del pruning globale non strutturato a causa della sua normalità. Nei dettagli in figura 4.10, il confronto tra la low rank approximation e i dati pre-minimizzazione ha mostrato un punteggio di 726.000 con un P-value di 0.000, indicando una differenza statisticamente significativa. Anche la quantizzazione ha prodotto risultati simili, con un punteggio di 3154.000 e P-value di 0.000. D'altra parte, il pruning casuale ha mostrato un P-value di 0.669, suggerendo che non c'era una differenza significativa rispetto ai dati pre-minimizzazione. Al contrario, il pruning globale ha presentato un T-statistic di 5.426 e un P-value di 0.000, rivelando un'efficace riduzione dei consumi. Il pruning non strutturato ha mostrato un punteggio di 4052.000 con un P-value di 0.006, mentre il pruning strutturato per canali ha evidenziato il punteggio più alto di 7163.000 e un P-value di 0.000, evidenziando la sua efficacia nel ridurre i consumi energetici.

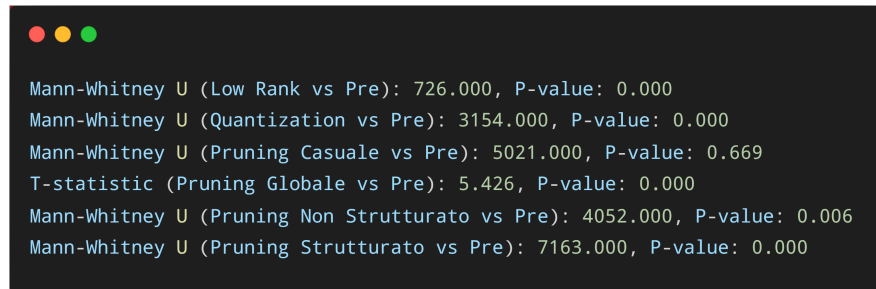


Figura 4.10: Test T e Test di Mann-Whitney U per VGG16

Attraverso i box plot si possono analizzare i risultati graficamente. Dal grafico 4.11, si può osservare che tutte le tecniche di pruning presentano pressoché tutte lo stesso consumo energetico con leggeri peggioramenti causati dai modelli ottimizzati con pruning globale non strutturato e pruning strutturato per canali. Mentre un miglioramento, seppur leggero ma con leggeri outlier è mostrato dal modello ottimizzato con pruning non strutturato. Il pruning casuale non strutturato ha consumi simili a quello baseline.

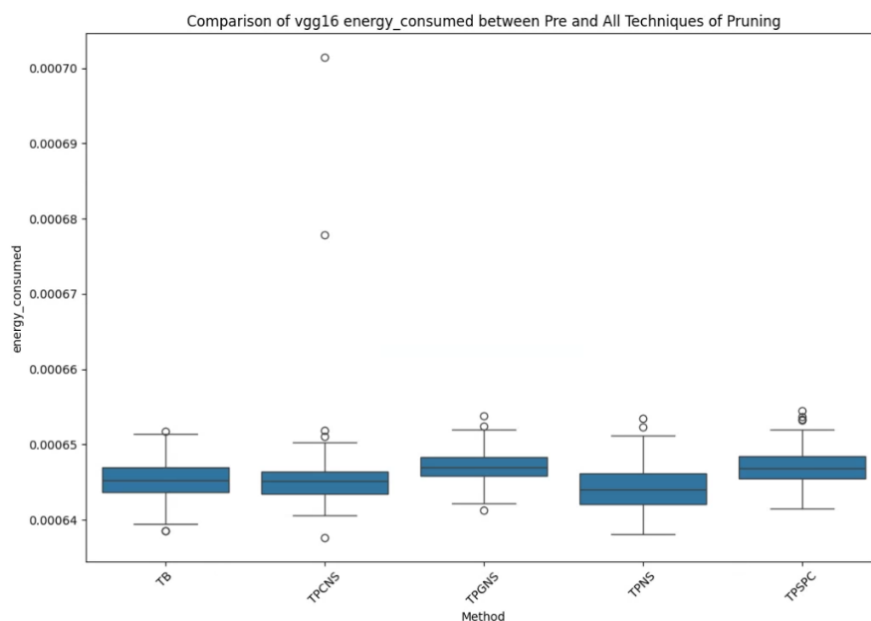


Figura 4.11: Grafico dei Consumi Energetici di VGG16 Pre e Pruning

Osservando il grafico 4.12 si può osservare una chiara tendenza decrescente nell'energia consumata passando dal modello baseline (TB) a quello quantizzato (TQ), fino a quello ottimizzato con low rank approximation (TL). Quest'ultimo mostra il consumo energetico più basso ma anche la maggiore variabilità.

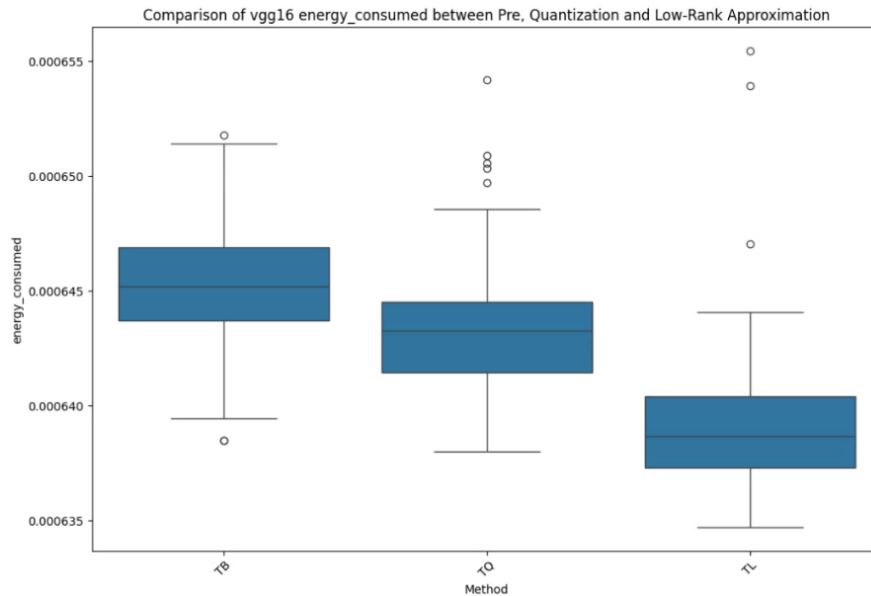


Figura 4.12: Grafico dei Consumi Energetici di VGG16 Pre, Quantizzazione e low rank Approximation

4.7.6 Performance Sustainability Trade-Off per VGG16

L'analisi dei trade-off tra performance e sostenibilità per il modello VGG16 offre spunti significativi su come diverse tecniche di ottimizzazione influenzino i consumi energetici e le prestazioni del modello.

Il modello baseline funge da base per il confronto delle tecniche. Esso presenta tempi di inferenza più lunghi rispetto a modelli più piccoli come ResNet18 e AlexNet, ma comunque mantiene buone performance complessive.

La low rank approximation ha dimostrato di avere il più basso consumo energetico tra tutte le tecniche di ottimizzazione applicate. Tuttavia, il consumo energetico ridotto si accompagna a una maggiore variabilità nei dati indicando come l'approccio potrebbe introdurre instabilità. In termini di performance, ha mantenuto le stesse metriche rendendola una delle tecniche più bilanciate per chi cerca di ridurre il consumo energetico senza sacrificare l'accuratezza.

La quantizzazione si dimostra una tecnica con una significativa riduzione dei consumi energetici, ma inferiori a quelli della low rank approximation. Dal punto di vista delle prestazioni, la quantizzazione mantiene delle metriche simili a quelle originali. Importante è anche l'aspetto della riduzione della dimensione del modello.

Per quanto riguarda le tecniche del pruning, il modello ottenuto con la tecnica

4.8 – RQ6: In che modo le tecniche di ottimizzazione trovate impattano i modelli ML sugli attributi di qualità?

di pruning globale non strutturato presenta un consumo energetico più elevato di quello baseline. Anche le performance sono leggermente più basse del modello di partenza.

Il pruning strutturato per canali ha mostrato un aumento dei consumi e una degradazione delle prestazioni di performance per cui non è una tecnica da preferire.

Per lo stesso motivo, anche il pruning casuale non strutturato ha un'elevata degradazione delle performance nonostante i consumi energetici restano pressoché invariati dal modello baseline.

Per quanto riguarda il pruning non strutturato, esso ha prodotto una leggerissima riduzione dei consumi energetici e delle performance.

In conclusione, l'analisi dei trade off tra performance e sostenibilità per VGG16 evidenzia alcune tecniche che si distinguono per la loro efficacia nel ridurre i consumi energetici senza compromettere in modo significativo le performance. Nuovamente la quantizzazione si rivela un'ottima tecnica che diminuisce i consumi energetici, senza peccare di molto sulle performance dei modelli. Invece la low rank approximation si mostra come una buona tecnica che diminuisce di poco le performance, ma che abbassa i consumi energetici. Le tecniche di pruning sono da evitare a causa della perdita delle performance o degli aumenti dei consumi energetici, a eccezione del pruning non strutturato.

4.8 RQ6: In che modo le tecniche di ottimizzazione trovate impattano i modelli ML sugli attributi di qualità?

L'analisi complessiva delle tecniche di minimizzazione applicate su tre modelli popolari evidenzia un chiaro trade-off tra performance e sostenibilità energetica:

- **Low Rank Approximation:**
 - *Performance:* In generale tende a mantenere le prestazioni di accuratezza, di precisione e di recall vicine al modello originale, specialmente su ResNet18 e VGG16. Tuttavia, in alcuni casi, come per Alexet, si osserva

una riduzione delle prestazioni, anche se rimangono accettabili. Anche il tempo di inferenza e di throughput mostra miglioramenti. Le dimensioni del modello sono le stesse di quello di baseline.

- *Sostenibilità*: Dal punto di vista energetico, questa tecnica produce risultati misti. Per AlexNet e VGG16, riduce significativamente i consumi energetici, ma introduce anche una maggiore variabilità nel comportamento energetico. Al contrario, su ResNet18, l'impatto sui consumi energetici è meno favorevole.
- *Conclusioni*: La low rank approximation è una scelta bilanciata per ridurre i consumi energetici senza compromettere troppo le performance, ma la variabilità nei risultati energetici può rappresentare un rischio.

- **Quantizzazione Dinamica:**

- *Performance*: Emerge come una delle tecniche più equilibrate in quanto mantiene alte le metriche di accuratezza, precisione, recall, tempo di inferenza e throughput, con prestazioni vicine al modello baseline. La dimensione dei modelli scende drasticamente consentendo l'applicazione dei modelli in dispositivi con limitato spazio di archiviazione.
- *Sostenibilità*: Questa tecnica si dimostra particolarmente efficace nella riduzione del consumo energetico.
- *Conclusioni*: La quantizzazione è una delle migliori tecniche per chi cerca di ottimizzare il consumo energetico senza sacrificare le prestazioni, ottenendo un modello più leggero.

- **Pruning**: L'impatto del pruning varia notevolmente in base alla tecnica applicata. In generale, queste tecniche si dimostrano meno efficaci in termini di sostenibilità energetica e performance rispetto alla low rank approximation e alla quantizzazione, ma ci sono alcune eccezioni:

- **Pruning Globale Non Strutturato:**

- * *Performance*: Mantiene prestazioni accettabili. Diminuiscono di circa il 5% le performance sulle metriche di accuratezza, precisione e recall sul

modello AlexNet, mentre sugli altri modelli tutte le metriche incluso il tempo di inferenza e il throughput restano pressoché simili al modello di partenza. Le dimensioni del modello sono le stesse di quello di baseline.

- * *Sostenibilità*: Il consumo energetico su VGG16 e ResNet18 risulta più elevato rispetto al modello baseline, mentre su AlexNet offre una leggera riduzione dei consumi.
- * *Conclusioni*: L'impatto dei consumi di questa tecnica non è sempre prevedibile, anche se mantiene buone performance sulle altre metriche.

– **Pruning Strutturato Per Canali:**

- * *Performance*: Risulta un calo marcato delle performance soprattutto sui modelli AlexNet e VGG16, con riduzioni significative dell'accuratezza, precisione e recall. I tempi di inferenza e il throughput invece, presentano un leggerissimo miglioramento rispetto al modello baseline. Le dimensioni del modello sono le stesse di quello di baseline.
- * *Sostenibilità*: Offre una scarsa riduzione del consumo energetico che resta pressoché simile al modello baseline.
- * *Conclusioni*: Tecnica poco utile a causa delle riduzioni delle performance. Poco applicabile se si ha necessità di diminuire i consumi energetici.

– **Pruning Non Strutturato**

- * *Performance*: Mantiene prestazioni relativamente buone sulle metriche di accuratezza, precisione, recall e AUC-ROC, specialmente su VGG16, ma su ResNet18 e AlexNet si osservano cali più significativi. I tempi di inferenza e il throughput restano invariati rispetto al modello di partenza. Le dimensioni del modello sono le stesse di quello di baseline.
- * *Sostenibilità*: I miglioramenti nei consumi energetici sono meno evidenti.

4.8 – RQ6: In che modo le tecniche di ottimizzazione trovate impattano i modelli ML sugli attributi di qualità?

- * *Conclusioni*: Una tecnica da considerare solo quando altre tecniche di pruning od ottimizzazione non sono applicabili, poiché l'efficienza energetica è limitata. Le performance e le accuratezze risultano buone.

– **Pruning Casuale Non Strutturato:**

- * *Performance*: Le performance sono generalmente scarse, con una significativa riduzione delle metriche di accuratezza, precisione e recall su tutti i modelli. Invece i tempi di inferenza e il throughput presentano lievi differenze rispetto al modello di partenza. Le dimensioni del modello sono identiche a quello baseline.
- * *Sostenibilità*: Anche dal punto di vista energetico, i benefici sono limitati e in alcuni casi, come su ResNet18, i consumi energetici aumentano.
- * *Conclusioni*: Questa tecnica è una delle meno raccomandate, poiché compromette sia le performance sia l'efficienza energetica.

🔗 **Answer to RQ₆.** In sintesi, il trade-off generale tra performance e sostenibilità energetica delle tecniche di minimizzazione nei modelli AlexNet, ResNet18 e VGG16 può essere così riassunto:

- **Quantizzazione** è la tecnica più vantaggiosa dal punto di vista energetico e rappresenta la scelta migliore per chi desidera mantenere elevate le prestazioni riducendo i consumi energetici. Infatti le metriche di accuratezza, precisione, recall, tempo di inferenza e throughput sono molto simili ai modelli baseline. Inoltre, è l'unica che direttamente riduce le dimensioni dei modelli.
- **Low Rank Approximation** è una buona scelta quando si desidera una riduzione del consumo energetico senza sacrificare troppo le prestazioni, ma va considerata la variabilità energetica che può introdurre.
- **Pruning** presenta risultati misti a seconda delle tecniche. Le tecniche di pruning globali e strutturate per canali offrono risultati variabili a seconda del modello di baseline, mentre risultati meno soddisfacenti sono riportati dal pruning casuale non strutturato. La tecnica del pruning non strutturato mostra dei peggioramenti in termini di accuratezza, precisione e recall, mentre i consumi energetici restano invariati.

Minacce alla Validità

In questo capitolo sono esposte le principali minacce alla validità dello studio condotto e sono elencati e discussi i principali fattori che potrebbero influenzare la validità dei risultati insieme alle precauzioni prese.

5.1 Minacce alla Validità Interna

Le minacce alla validità interna sono causate da un qualsiasi fattore che può distogliere i risultati dello studio, che potrebbe rendere incerta la relazione causale tra le variabili. Un esempio di minaccia alla validità interna è la selezione del dataset in quanto se non fosse stato rappresentativo o avrebbe contenuto errori avrebbe potuto compromettere le conclusioni sui trade-off tra prestazioni e sostenibilità. Per mitigare questa minaccia è stato selezionato il dataset Imagenet in quanto tutti e tre i modelli posti sotto esame sono stati addestrati con esso.

5.2 Minacce alla Validità Esterna

La validità esterna si riferisce alla possibilità di generalizzare i risultati dello studio ad altri contesti. Nel caso di questo studio, le minacce potrebbero derivare

dall'utilizzo di specifici modelli (AlexNet, ResNet18, VGG16), che potrebbero non rappresentare lo spettro intero di modelli di machine learning. In aggiunta, anche l'utilizzo di un singolo dataset potrebbe non essere sufficiente a generalizzare i risultati ad altre applicazioni.

5.3 Minacce alla Validità della Selezione degli Studi

Le minacce alla validità della selezione degli studi riguarda il processo di scelta degli studi da includere in una systematic literature review. La selezione degli articoli potrebbe aver escluso studi rilevanti a causa dei criteri di esclusione, della query di ricerca o della scelta di database specifici. Tuttavia, questa minaccia è stata mitigata attraverso l'uso del processo di snowballing, che ha permesso di ampliare la ricerca includendo ulteriori studi rilevanti. In particolare sono stati applicati metodi di forward e backward snowballing che hanno portato all'analisi di ulteriori 43.130 articoli.

5.4 Minacce alla Validità di Costrutto

La validità di costrutto riguarda la corretta misurazione dei concetti teorici alla base dello studio. Una minaccia è l'interpretazione delle tecniche di ottimizzazione che potrebbero essere implementate o valutate in modo diverso rispetto ad altri studi.

5.5 Minacce alla Validità della Conclusione

La minaccia alla conclusione riguarda la solidità dei risultati ottenuti. Un rischio potrebbe essere l'analisi statistica dei dati se non vengono utilizzati test appropriati. Inoltre, se non si considerano in maniera adeguata gli effetti delle diverse tecniche di minimizzazione, si rischia di trarre conclusioni sbagliate sull'impatto delle prestazioni e sostenibilità.

CAPITOLO 6

Conclusioni

Questo capitolo riassume i risultati ottenuti nel corso del lavoro di ricerca, evidenziando il contributo dello studio alla letteratura scientifica e proponendo sviluppi futuri. L'obiettivo primario di questo studio è stato valutare il trade-off ottimale tra performance, accuratezza e sostenibilità energetica nei modelli di machine learning. A tal proposito, sono stati risposti due obiettivi di ricerca: l'individuazione degli approcci e tecniche di riduzione delle dimensioni dei modelli di machine learning, tramite una systematic literature review; la realizzazione di uno studio di benchmark sui modelli individuati, per valutare come l'uso di tecniche di riduzione delle dimensioni influenzino l'accuratezza, le prestazioni e il consumo energetico, identificando il trade-off tra performance e sostenibilità.

La systematic literature review ha fornito importanti informazioni e i risultati più rilevanti possono essere riassunti come segue:

- RQ1: Le tecniche principali per ridurre la dimensione dei modelli di machine learning includono pruning, quantizzazione, knowledge distillation, low rank approximation, compressione lossy e modelli light-weight. Tecniche indirette comprendono algoritmi evolutivi, quantum autoencoder, quantum autoencoder, random projection e sostituzione della memoria.

- RQ2: Le tecniche di ridimensionamento dei modelli di machine learning sono principalmente applicate a task di Computer Vision, seguite da NLP e Signal Classification. Altri task includono il Biosignal Processing, la rilevazione e la diagnosi dei guasti, la classificazione, il Time Series Prediction con elementi di Spatiotemporal Modeling, la Sound Classification, il Federated Learning, e l'Analisi di Big Data.
- RQ3: Le tecniche di ridimensionamento possono essere applicate durante l'addestramento (pruning, quantizzazione), dopo l'addestramento (pruning, quantizzazione, knowledge distillation, low rank approximation, compressione lossy, random project algorithm e sostituzione della memoria), nella definizione dell'architettura (modelli light-weight), nella fase di ottimizzazione (modelli evolutivi, group teaching optimization algorithm) e in fase di pre-elaborazione (quantum autoencoder, random project algorithm).
- RQ4: Le tecniche di ridimensionamento dei modelli di machine learning mirano principalmente a ottimizzare i seguenti requisiti non funzionali: efficienza delle prestazioni, affidabilità, flessibilità, consumo energetico, manutenibilità e compatibilità.
- RQ5: Le tecniche di ridimensionamento dei modelli di machine learning sono valutate principalmente attraverso l'*accuratezza*, la *precisione*, il *recall*, l'*AUC-ROC*, il *tempo di inferenza*, il *throughput*, l'*uso delle risorse computazionali (CPU/GPU)*, e la *dimensione del modello*, con un'attenzione crescente verso il *consumo energetico* e la *sostenibilità*.

Lo studio di benchmark ha permesso di valutare l'impatto delle tecniche di ottimizzazione, quali pruning, low rank approximation e quantizzazione, sugli attributi di qualità. I risultati possono essere riassunti così:

- La quantizzazione si è dimostrata la più vantaggiosa sia in termini di riduzione dei consumi energetici che di prestazioni. I modelli risultano significativamente più leggeri rispetto alla baseline, mantenendo metriche di accuratezza, precisione, recall, tempo di inferenza e throughput molto simili a quelle dei modelli originali.

- La low rank approximation è una buona scelta quando si desidera una riduzione del consumo energetico senza sacrificare troppo le prestazioni.
- Il pruning presenta risultati misti a seconda delle tecniche. Il pruning globale e strutturato per canali ha mostrato risultati misti, mentre il pruning casuale non strutturato ha dato i peggiori risultati. La tecnica del pruning non strutturato mostra peggioramenti in termini di accuratezza e performance, mentre i consumi energetici restano invariati.

In conclusione, questo lavoro ha fornito una visione chiara del trade-off tra prestazioni e sostenibilità energetica delle tecniche di ridimensionamento dei modelli di machine learning. L'uso della quantizzazione ha dimostrato di essere particolarmente efficace in questo. Tuttavia, altri approcci richiedono l'utilizzo di altre tecniche, quali la quantizzazione, per ridurre la dimensione del modello dopo l'ottimizzazione.

Per quanto riguarda gli sviluppi futuri, sarebbe utile esplorare tecniche avanzate che combinano diversi approcci per identificare le migliori soluzioni di riduzione delle dimensioni. Inoltre, potrebbe essere interessante valutare l'impatto di queste tecniche a nuovi modelli e task, come quelli dell'elaborazione del linguaggio naturale e il Signal Processing.

Bibliografia

- [1] John Butlin. Our common future. by world commission on environment and development.(london, oxford university press, 1987, pp. 383 5.95.), 1989. (Citato a pagina 6)
- [2] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluís-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021. (Citato a pagina 6)
- [3] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green ai. *Communications of the ACM*, 63(12):54–63, 2020. (Citato a pagina 6)
- [4] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13693–13696, 2020. (Citato a pagina 6)
- [5] Stefanos Georgiou, Maria Kechagia, Tushar Sharma, Federica Sarro, and Ying Zou. Green ai: Do deep learning frameworks have different costs? In *Proceedings of the 44th International Conference on Software Engineering*, pages 1082–1094, 2022. (Citato a pagina 7)

-
- [6] Roberto Verdecchia, June Sallou, and Luís Cruz. A systematic review of green ai. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(4):e1507, 2023. (Citato a pagina 7)
- [7] Lizhi Liao, Heng Li, Weiyi Shang, and Lei Ma. An empirical study of the impact of hyperparameter tuning and model optimization on the performance properties of deep neural networks. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 31(3):1–40, 2022. (Citato a pagina 7)
- [8] Raluca Maria Hampau, Maurits Kaptein, Robin Van Emden, Thomas Rost, and Ivano Malavolta. An empirical study on the performance and energy consumption of ai containerization strategies for computer-vision tasks on the edge. In *Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering*, pages 50–59, 2022. (Citato a pagina 7)
- [9] Simin Chen, Mirazul Haque, Cong Liu, and Wei Yang. Deeppperform: An efficient approach for performance testing of resource-constrained neural networks. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pages 1–13, 2022. (Citato a pagina 7)
- [10] Abbas Ourmazd. Science in the age of machine learning. *Nature Reviews Physics*, 2(7):342–343, 2020. (Citato a pagina 7)
- [11] Simone Disabato and Manuel Roveri. Incremental on-device tiny machine learning. In *Proceedings of the 2nd International workshop on challenges in artificial intelligence and machine learning for internet of things*, pages 7–13, 2020. (Citato a pagina 8)
- [12] Pete Warden and Daniel Situnayake. *Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers*. O’Reilly Media, 2019. (Citato a pagina 8)
- [13] Riku Immonen, Timo Hämäläinen, et al. Tiny machine learning for resource-constrained microcontrollers. *Journal of Sensors*, 2022, 2022. (Citato a pagina 9)

- [14] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. (Citato a pagina 10)
- [15] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018. (Citato a pagina 11)
- [16] Barbara Kitchenham, O Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering—a systematic literature review. *Information and software technology*, 51(1):7–15, 2009. (Citato alle pagine 13 e 16)
- [17] Claes Wohlin. Second-generation systematic literature studies using snowballing. In *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*, pages 1–6, 2016. (Citato alle pagine 13 e 16)
- [18] Asad Ali and Carmine Gravino. A systematic literature review of software effort prediction using machine learning methods. *Journal of software: evolution and process*, 31(10):e2211, 2019. (Citato a pagina 18)

Systematic Literature Review

- [SLR1] Ming Zhao, Meng Li, Sheng-Lung Peng, and Jie Li. A novel deep learning model compression algorithm. *Electronics*, 11(7):1066, 2022.
- [SLR2] Xinyu Hao, Zhang Xia, Mengxi Jiang, Qiubo Ye, and Guangsong Yang. Radio signal modulation recognition method based on deep learning model pruning. *Applied Sciences*, 12(19):9894, 2022.
- [SLR3] Hyungkeuk Lee, NamKyung Lee, and Sungjin Lee. A method of deep learning model optimization for image classification on edge device. *Sensors*, 22(19):7344, 2022.
- [SLR4] Muhammad Sami Ullah, Muhammad Attique Khan, Anum Masood, Olfa Mzoughi, Oumaima Saidani, and Nazik Alturki. Brain tumor classification from mri scans: a framework of hybrid deep learning model with bayesian optimization and quantum theory-based marine predator algorithm. *Frontiers in Oncology*, 14:1335740, 2024.
- [SLR5] Yuan Gao, Shohei Miyata, and Yasunori Akashi. How to improve the application potential of deep learning model in hvac fault diagnosis: Based on pruning and interpretable deep learning method. *Applied Energy*, 348:121591, 2023.

- [SLR6] Zhuangzhi Chen, Zhangwei Wang, Xuzhang Gao, Jinchao Zhou, Dongwei Xu, Shilian Zheng, Qi Xuan, and Xiaoni Yang. Channel pruning method for signal modulation recognition deep learning models. *IEEE Transactions on Cognitive Communications and Networking*, 2023.
- [SLR7] José Vitor Santos Silva, Leonardo Matos Matos, Flávio Santos, Héllisson Oliveira Magalhães Cerqueira, Hendrik Macedo, Bruno Otávio Piedade Prado, Gilton José Ferreira da Silva, and Kalil Araújo Bispo. Combining deep learning model compression techniques. *IEEE Latin America Transactions*, 20(3):458–464, 2021.
- [SLR8] Martinson Ofori, Omar El-Gayar, Austin O’Brien, and Cherie Noteboom. A deep learning model compression and ensemble approach for weed detection. 2022.
- [SLR9] Thivindu Paranayapa, Piumini Ranasinghe, Dakshina Ranmal, Dulani Mee-deniya, and Charith Perera. A comparative study of preprocessing and model compression techniques in deep learning for forest sound classification. *Sensors*, 24(4):1149, 2024.
- [SLR10] Weiguo Shen, Wei Wang, Jiawei Zhu, Huaji Zhou, and Shunling Wang. Pruning-and quantization-based compression algorithm for number of mixed signals identification network. *Electronics*, 12(7):1694, 2023.
- [SLR11] Hubert Msuya, Baraka J Maiseli, et al. Deep learning model compression techniques: Advances, opportunities, and perspective. *Tanzania Journal of Engineering and Technology*, 42(2):65–83, 2023.
- [SLR12] Qi Li, Hengyi Li, and Lin Meng. Deep learning architecture improvement based on dynamic pruning and layer fusion. *Electronics*, 12(5):1208, 2023.
- [SLR13] Abid Hussain, Heng-Chao Li, Danish Ali, Muqadar Ali, Fakhar Abbas, and Mehboob Hussain. An optimized deep supervised hashing model for fast image retrieval. *Image and Vision Computing*, 133:104668, 2023.

- [SLR14] Ming Zhao, Min Hu, Meng Li, Sheng-Lung Peng, and Junbo Tan. A novel fusion pruning algorithm based on information entropy stratification and iot application. *Electronics*, 11(8):1212, 2022.
- [SLR15] Ting-Wu Chin, Ruizhou Ding, Cha Zhang, and Diana Marculescu. Towards efficient model compression via learned global ranking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1518–1528, 2020.
- [SLR16] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. *IEEE Signal Processing Magazine*, 35(1):126–136, 2018.
- [SLR17] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4340–4349, 2019.
- [SLR18] Mingbao Lin, Rongrong Ji, Yan Wang, Yichen Zhang, Baochang Zhang, Yonghong Tian, and Ling Shao. Hrank: Filter pruning using high-rank feature map. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1529–1538, 2020.
- [SLR19] Eunho Lee and Youngbae Hwang. Layer-wise network compression using gaussian mixture model. *Electronics*, 10(1):72, 2021.
- [SLR20] Zherui Zhang and Ya Tu. A pruning neural network for automatic modulation classification. In *2021 8th International Conference on Dependable Systems and Their Applications (DSA)*, pages 189–194. IEEE, 2021.
- [SLR21] Xiaolian Liu, Shaopeng Gong, Xiangxu Hua, Taotao Chen, and Chunjiang Zhao. Research on temperature detection method of liquor distilling pot feeding operation based on a compressed algorithm. *Scientific Reports*, 14(1):13292, 2024.
- [SLR22] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the*

- IEEE/CVF conference on computer vision and pattern recognition, pages 11264–11272, 2019.
- [SLR23] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.
- [SLR24] Pengyi Zhang, Yunxin Zhong, and Xiaoqiong Li. Slimyolov3: Narrower, faster and better for real-time uav applications. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.
- [SLR25] Dihua Wu, Shuaichao Lv, Mei Jiang, and Huaibo Song. Using channel pruning-based yolo v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. *Computers and Electronics in Agriculture*, 178:105742, 2020.
- [SLR26] Dandan Wang and Dongjian He. Channel pruned yolo v5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. *Biosystems Engineering*, 210:271–281, 2021.
- [SLR27] Nhut Huynh and Kim-Doang Nguyen. Real-time droplet detection for agricultural spraying systems: A deep learning approach. *Machine Learning and Knowledge Extraction*, 6(1):259–282, 2024.
- [SLR28] Woomin Jun, Minjun Son, Jisang Yoo, and Sungjin Lee. Optimal configuration of multi-task learning for autonomous driving. *Sensors*, 23(24):9729, 2023.
- [SLR29] Woomin Jun, Jisang Yoo, and Sungjin Lee. Synthetic data enhancement and network compression technology of monocular depth estimation for real-time autonomous driving system. *Sensors*, 24(13):4205, 2024.
- [SLR30] Cheng Fan, Ruikun Chen, Jinhan Mo, and Longhui Liao. Personalized federated learning for cross-building energy knowledge sharing: Cost-effective strategies and model architectures. *Applied Energy*, 362:123016, 2024.

- [SLR31] Kit Yan Chan, Ka Fai Cedric Yiu, Shan Guo, and Huimin Jiang. A roulette wheel-based pruning method to simplify cumbersome deep neural networks. *Neural Computing and Applications*, pages 1–19, 2024.
- [SLR32] Takashi Yokota, Kazunori Kojima, Shi-wook Lee, and Yoshiaki Itoh. Reduction of speech data posteriorgrams by compressing maximum-likelihood state sequences in query by example. In *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 649–653. IEEE, 2020.
- [SLR33] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461:370–403, 2021.
- [SLR34] Russell Reed. Pruning algorithms-a survey. *IEEE transactions on Neural Networks*, 4(5):740–747, 1993.
- [SLR35] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020.
- [SLR36] Miguel A Carreira-Perpinán and Yerlan Idelbayev. “learning-compression” algorithms for neural net pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8532–8541, 2018.
- [SLR37] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pages 5058–5066, 2017.
- [SLR38] Liyang Liu, Shilong Zhang, Zhanghui Kuang, Aojun Zhou, Jing-Hao Xue, Xinjiang Wang, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Group fisher pruning for practical network compression. In *International Conference on Machine Learning*, pages 7021–7032. PMLR, 2021.
- [SLR39] Shangqian Gao, Feihu Huang, Weidong Cai, and Heng Huang. Network pruning via performance maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9270–9280, 2021.

- [SLR40] Olutosin Ajibola Ademola, Mairo Leier, and Eduard Petlenkov. Evaluation of deep neural network compression methods for edge devices using weighted score-based ranking scheme. *Sensors*, 21(22):7529, 2021.
- [SLR41] Thuan Q Huynh and Rudy Setiono. Effective neural network pruning using cross-validation. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 972–977. IEEE, 2005.
- [SLR42] Jing Chang and Jin Sha. Prune deep neural networks with the modified $l_{\{1/2\}}$ penalty. *IEEE Access*, 7:2273–2280, 2018.
- [SLR43] Arnauld Nzegha Fountsop, Jean Louis Ebongue Kedieng Fendji, and Marcellin Atemkeng. Deep learning models compression for agricultural plants. *Applied Sciences*, 10(19):6866, 2020.
- [SLR44] Ming Zhao, Xindi Tong, Weixian Wu, Zhen Wang, Bingxue Zhou, and Xiaodan Huang. A novel deep-learning model compression based on filter-stripe group pruning and its iot application. *Sensors*, 22(15):5623, 2022.
- [SLR45] Meng Li, Ming Zhao, Tie Luo, Yimin Yang, and Sheng-Lung Peng. A compact parallel pruning scheme for deep learning model and its mobile instrument deployment. *Mathematics*, 10(12):2126, 2022.
- [SLR46] Seema Bhalgaonkar, Mousami Munot, et al. Model compression of deep neural network architectures for visual pattern recognition: Current status and future directions. *Computers and Electrical Engineering*, 116:109180, 2024.
- [SLR47] Deepak Ghimire, Dayoung Kil, and Seong-heum Kim. A survey on efficient convolutional neural networks and hardware acceleration. *Electronics*, 11(6):945, 2022.
- [SLR48] Donghyeon Lee, Eunho Lee, and Youngbae Hwang. Pruning from scratch via shared pruning module and nuclear norm-based regularization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1393–1402, 2024.

- [SLR49] Donghyeon Lee, Eunho Lee, and Youngbae Hwang. Lossless reconstruction of convolutional neural network for channel-based network pruning. *Sensors*, 23(4):2102, 2023.
- [SLR50] Yi-Cheng Lo, Cheng-Lin Hsieh, and An-Yeu Andy Wu. Constraints-aware trainable pruning with system optimization for the on-demand offloading edge-cloud collaborative system. In *2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2023.
- [SLR51] Bingzhao Zhu, Masoud Farivar, and Mahsa Shoaran. Resot: Resource-efficient oblique trees for neural signal classification. *IEEE Transactions on Biomedical Circuits and Systems*, 14(4):692–704, 2020.
- [SLR52] Bharath Srinivas Prabakaran, Asima Akhtar, Semeen Rehman, Osman Hassan, and Muhammad Shafique. Bionetexplorer: Architecture-space exploration of biosignal processing deep neural networks for wearables. *IEEE Internet of Things Journal*, 8(17):13251–13265, 2021.
- [SLR53] Sebastian Müksch, Theo Olausson, John Wilhelm, and Pavlos Andreadis. Benchmarking the accuracy of algorithms for memory-constrained image classification. In *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 400–404. IEEE, 2020.
- [SLR54] Hyungkeuk Lee, NamKyung Lee, and Sungjin Lee. A method of deep learning model optimization for image classification on edge device. *Sensors*, 22(19):7344, 2022.
- [SLR55] Huidong Liu, Fang Du, Lijuan Song, and Zhenhua Yu. Block-wisely supervised network pruning with knowledge distillation and markov chain monte carlo. *Applied Sciences*, 12(21):10952, 2022.
- [SLR56] Woosuk Kwon, Sehoon Kim, Michael W Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. A fast post-training pruning framework for transformers. *Advances in Neural Information Processing Systems*, 35:24101–24116, 2022.

- [SLR57] Ali Alqahtani, Xianghua Xie, and Mark W Jones. Literature review of deep network compression. In *Informatics*, volume 8, page 77. MDPI, 2021.
- [SLR58] Shangqian Gao, Zeyu Zhang, Yanfu Zhang, Feihu Huang, and Heng Huang. Structural alignment for network pruning through partial regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17402–17412, 2023.
- [SLR59] Bratislav Predić, Uroš Vukić, Muzafer Saračević, Darjan Karabašević, and Dragiša Stanujkić. The possibility of combining and implementing deep neural network compression methods. *Axioms*, 11(5):229, 2022.
- [SLR60] Yatao Li, Leiying He, Jianneng Chen, Jun Lyu, Chuanyu Wu, et al. High-efficiency tea shoot detection method via a compressed deep learning model. *International Journal of Agricultural and Biological Engineering*, 15(3):159–166, 2022.
- [SLR61] Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3):1–18, 2017.
- [SLR62] Cheng-Yang Chang, Yu-Chuan Chuang, Kuang-Chao Chou, and An-Yeu Wu. T-eap: Trainable energy-aware pruning for nvm-based computing-in-memory architecture. In *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 78–81. IEEE, 2022.
- [SLR63] Xin Liu, Yue Zhang, Zenghai Wang, and Jie Yang. Research on deep learning model and optimization algorithm in edge computing. In *2023 5th International Conference on Applied Machine Learning (ICAML)*, pages 242–246. IEEE, 2023.
- [SLR64] Olutosin Ajibola Ademola, Petlenkov Eduard, and Leier Mairo. Ensemble of tensor train decomposition and quantization methods for deep learning model compression. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2022.

- [SLR65] Feng Zhu, Ruihao Gong, Fengwei Yu, Xianglong Liu, Yanfei Wang, Zhelong Li, Xiuqi Yang, and Junjie Yan. Towards unified int8 training for convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1969–1979, 2020.
- [SLR66] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018.
- [SLR67] Robert Stewart, Andrew Nowlan, Pascal Bacchus, Quentin Ducasse, and Ekaterina Komendantskaya. Optimising hardware accelerated neural networks with quantisation and a knowledge distillation evolutionary algorithm. *Electronics*, 10(4):396, 2021.
- [SLR68] Phuoc Pham and Jaeyong Chung. Improving model capacity of quantized networks with conditional computation. *Electronics*, 10(8):886, 2021.
- [SLR69] Zhe Han, Jingfei Jiang, Linbo Qiao, Yong Dou, Jinwei Xu, and Zhigang Kan. Accelerating event detection with dgcnn and fpgas. *Electronics*, 9(10):1666, 2020.
- [SLR70] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proceedings of the IEEE*, 108(4):485–532, 2020.
- [SLR71] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International conference on machine learning*, pages 1737–1746. PMLR, 2015.
- [SLR72] Xin Long, XiangRong Zeng, Zongcheng Ben, Dianle Zhou, and Maojun Zhang. A novel low-bit quantization strategy for compressing deep neural networks. *Computational Intelligence and Neuroscience*, 2020(1):7839064, 2020.

- [SLR73] Xuexiang Li, Hansheng Yang, Cong Yang, and Weixing Zhang. Efficient medical knowledge graph embedding: Leveraging adaptive hierarchical transformers and model compression. *Electronics*, 12(10):2315, 2023.
- [SLR74] Lorenz Kummer, Kevin Sidak, Tabea Reichmann, and Wilfried Gansterer. Adaptive precision training (adapt): A dynamic quantized training approach for dnns. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 559–567. SIAM, 2023.
- [SLR75] Cristian Sestito, Stefania Perri, and Robert Stewart. Accuracy evaluation of transposed convolution-based quantized neural networks. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.
- [SLR76] Cristian Sestito, Stefania Perri, and Robert Stewart. Design-space exploration of quantized transposed convolutional neural networks for fpga-based systems-on-chip. In *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDDCom/CyberSciTech)*, pages 1–6. IEEE, 2022.
- [SLR77] Josphat Chege Njuguna, Aysun Taşyapı Çelebi, and Anıl Çelebi. Implementation and optimization of lenet-5 model for handwritten digits recognition on fpgas using brevitas and finn. In *2023 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 1–5. IEEE, 2023.
- [SLR78] Mustafa Tasci, Ayhan Istanbulu, Selahattin Kosunalp, Teodor Iliev, Ivaylo Stoyanov, and Ivan Beloev. An efficient classification of rice variety with quantized neural networks. *Electronics*, 12(10):2285, 2023.
- [SLR79] Shien Zhu, Luan HK Duong, and Weichen Liu. Tab: Unified and optimized ternary, binary, and mixed-precision neural network inference on the edge. *ACM Transactions on Embedded Computing Systems (TECS)*, 21(5):1–26, 2022.

- [SLR80] Pierre-Emmanuel Novac, Ghouthi Boukli Hacene, Alain Pegatoquet, Benoit Miramond, and Vincent Gripon. Quantization and deployment of deep neural networks on microcontrollers. *Sensors*, 21(9):2984, 2021.
- [SLR81] Qian Huang and Zhimin Tang. High-performance and lightweight ai model for robot vacuum cleaners with low bitwidth strong non-uniform quantization. *AI*, 4(3):531–550, 2023.
- [SLR82] Zhenhong Sun, Ce Ge, Junyan Wang, Ming Lin, Heseng Chen, Hao Li, and Xiuyu Sun. Entropy-driven mixed-precision quantization for deep network design. *Advances in Neural Information Processing Systems*, 35:21508–21520, 2022.
- [SLR83] Fouad Sakr, Riccardo Berta, Joseph Doyle, Hamoud Younes, Alessandro De Gloria, and Francesco Bellotti. Memory efficient binary convolutional neural networks on microcontrollers. In *2022 IEEE International Conference on Edge Computing and Communications (EDGE)*, pages 169–177. IEEE, 2022.
- [SLR84] Jixing Li, Gang Chen, Min Jin, Wenyu Mao, and Huaxiang Lu. Ae-qdrop: Towards accurate and efficient low-bit post-training quantization for a convolutional neural network. *Electronics*, 13(3):644, 2024.
- [SLR85] Huifang Li, Guangzheng Hu, Jianqiang Li, and Mengchu Zhou. Intelligent fault diagnosis for large-scale rotating machines using binarized deep neural networks and random forests. *IEEE Transactions on Automation Science and Engineering*, 19(2):1109–1119, 2021.
- [SLR86] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. *Advances in neural information processing systems*, 29, 2016.
- [SLR87] Aminu Musa, Mohammed Hassan, Mohamed Hamada, and Farouq Aliyu. Low-power deep learning model for plant disease detection for smart-hydroponics using knowledge distillation techniques. *Journal of Low Power Electronics and Applications*, 12(2):24, 2022.

- [SLR88] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge distillation with adversarial samples supporting decision boundary. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3771–3778, 2019.
- [SLR89] Tianhong Li, Jianguo Li, Zhuang Liu, and Changshui Zhang. Few sample knowledge distillation for efficient network compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14639–14647, 2020.
- [SLR90] Hui Wang, Hanbin Zhao, Xi Li, and Xu Tan. Progressive blockwise knowledge distillation for neural network acceleration. In *IJCAI*, pages 2769–2775, 2018.
- [SLR91] Yuan Gao, Shohei Miyata, and Yasunori Akashi. Automated fault detection and diagnosis of chiller water plants based on convolutional neural network and knowledge distillation. *Building and Environment*, 245:110885, 2023.
- [SLR92] Zhihui Li, Pengfei Xu, Xiaojun Chang, Luyao Yang, Yuanyuan Zhang, Lina Yao, and Xiaojiang Chen. When object detection meets knowledge distillation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10555–10579, 2023.
- [SLR93] Aminu Musa, Mohamed Hamada, and Mohammed Hassan. A theoretical framework towards building a lightweight model for pothole detection using knowledge distillation approach. In *SHS Web of Conferences*, volume 139, page 03002. EDP Sciences, 2022.
- [SLR94] Yixia Chen, Mingwei Lin, Zhu He, Kemal Polat, Adi Alhudhaif, and Fayadh Alenezi. Consistency-and dependence-guided knowledge distillation for object detection in remote sensing images. *Expert Systems with Applications*, 229:120519, 2023.
- [SLR95] Lingjun Zhao, Jingyu Song, and Katherine A Skinner. Crkd: Enhanced camera-radar object detection with cross-modality knowledge distillation.

- In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15470–15480, 2024.
- [SLR96] Dejie Yang and Yang Liu. Active object detection with knowledge aggregation and distillation from large models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16624–16633, 2024.
- [SLR97] Qinyuan Huang, Kun Yang, Yuzhen Zhu, Long Chen, and Lijia Cao. Knowledge distillation for enhancing a lightweight magnet tile target detection model: Leveraging spatial attention and multi-scale output features. *Electronics*, 12(22):4589, 2023.
- [SLR98] Zhiqi Shen, Kaiquan Cai, Quan Fang, and Xiaoyan Luo. Air traffic flow prediction with spatiotemporal knowledge distillation network. *Journal of Advanced Transportation*, 2024(1):4349402, 2024.
- [SLR99] Edwin Goh, Isaac R Ward, Grace Vincent, Kai Pak, Jingdao Chen, and Brian Wilson. Self-supervised distillation for computer vision onboard planetary robots. In *2023 IEEE Aerospace Conference*, pages 1–11. IEEE, 2023.
- [SLR100] Grace M Vincent, Isaac R Ward, Charles Moore, Jingdao Chen, Kai Pak, Alice Yepremyan, Brian Wilson, and Edwin Y Goh. Clover: Contrastive learning for onboard vision-enabled robotics. *Journal of Spacecraft and Rockets*, 61(3):728–740, 2024.
- [SLR101] Longrong Yang, Xianpan Zhou, Xuewei Li, Liang Qiao, Zheyang Li, Ziwei Yang, Gaoang Wang, and Xi Li. Bridging cross-task protocol inconsistency for distillation in dense object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17175–17184, 2023.
- [SLR102] Mengya Gao, Yujun Wang, and Liang Wan. Residual error based knowledge distillation. *Neurocomputing*, 433:154–161, 2021.
- [SLR103] Yuang Liu, Wei Zhang, and Jun Wang. Adaptive multi-teacher multi-level knowledge distillation. *Neurocomputing*, 415:106–113, 2020.

- [SLR104] Dechun Song, Peiyong Zhang, and Feiteng Li. Speeding up deep convolutional neural networks based on tucker-cp decomposition. In *Proceedings of the 2020 5th International Conference on Machine Learning Technologies*, pages 56–61, 2020.
- [SLR105] Se-Min Lim and Sang-Woo Jun. Mobilenets can be lossily compressed: Neural network compression for embedded accelerators. *Electronics*, 11(6):858, 2022.
- [SLR106] Ji Lin, Wei-Ming Chen, Yujun Lin, Chuang Gan, Song Han, et al. Mcunet: Tiny deep learning on iot devices. *Advances in neural information processing systems*, 33:11711–11722, 2020.
- [SLR107] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pages 2736–2744, 2017.
- [SLR108] Albert Gural and Boris Murmann. Memory-optimal direct convolutions for maximizing classification accuracy in embedded applications. In *ICML*, pages 2515–2524, 2019.
- [SLR109] Zhi Zhang, Yongzong Lu, Yiqiu Zhao, Qingmin Pan, Kuang Jin, Gang Xu, and Yongguang Hu. Ts-yolo: an all-day and lightweight tea canopy shoots detection model. *Agronomy*, 13(5):1411, 2023.
- [SLR110] Oindrila Saha, Aditya Kusupati, Harsha Vardhan Simhadri, Manik Varma, and Prateek Jain. Rnnpool: Efficient non-linear pooling for ram constrained inference. *Advances in Neural Information Processing Systems*, 33:20473–20484, 2020.
- [SLR111] Amit Sharma, Suneet Kumar Gupta, Divya Kumari, Surya Teja Reddy Dwarampudi, Gundam Bhanu Prakash Reddy, and Dibyanarayan Hazra. Compressed deep learning model for detecting covid-19 disease: A genetic algorithm based approach. In *2023 Second International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, pages 121–125. IEEE, 2023.

- [SLR112] Arcadi Llanza, Fekhr Eddine Keddous, Nadiya Shvai, and Amir Nakib. Deep learning models compression based on evolutionary algorithms and digital fractional differentiation. In *2023 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–9. IEEE, 2023.
- [SLR113] R Ramesh and S Sathiamoorthy. Blood vessel segmentation and classification for diabetic retinopathy grading using dandelion optimization algorithm with deep learning model. *International Journal of Intelligent Engineering & Systems*, 16(5), 2023.
- [SLR114] G Prabakaran, K Venkatesh, Suma Christal Mary Sundararajan, G Shobana, S Srimathi, et al. Group teaching optimization algorithm with machine learning model for big data analytics. In *2023 International Conference on Sustainable Communication Networks and Application (ICSCNA)*, pages 147–152. IEEE, 2023.
- [SLR115] Morteza Heidari, Sivaramakrishnan Lakshmivarahan, Seyedehnafiseh Mirniaharikandehei, Gopichandh Danala, Sai Kiran R Maryada, Hong Liu, and Bin Zheng. Applying a random projection algorithm to optimize machine learning model for breast lesion classification. *IEEE Transactions on Biomedical Engineering*, 68(9):2764–2775, 2021.
- [SLR116] Fouad Sakr, Francesco Bellotti, Riccardo Berta, Alessandro De Gloria, and Joseph Doyle. Memory-efficient cmsis-nn with replacement strategy. In *2021 8th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 299–303. IEEE, 2021.
- [SLR117] Andrei Velichko. Neural network for low-memory iot devices and mnist image recognition using kernels based on logistic map. *Electronics*, 9(9):1432, 2020. (Citato a pagina 26)
- [SLR118] Andrei Velichko. A method for medical data analysis using the lognnet for clinical decision support systems and edge computing in healthcare. *Sensors*, 21(18):6209, 2021. (Citato a pagina 26)