



UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Corso di Laurea Triennale in Ingegneria Meccanica

**Applicazione di un metodo di deep learning per il riconoscimento delle fibre in
materiali compositi**

**Application of a deep learning method for the identification of fibers in
composite materials**

Relatore: Prof.ssa
Milena Martarelli

Tesi di Laurea di:
Federico Daminato

Correlatore: Prof.
Alessio Trolli

A.A. 2022/2023

Indice

Indice	1
Introduzione	3
Capitolo 1 Materiali compositi	5
1.1 Generalità sui materiali compositi	5
1.2 Struttura dei materiali compositi.....	5
1.2.1 Fibre nei materiali compositi	7
1.3 Proprietà dei materiali compositi	8
1.4 Esempi di applicazione dei materiali compositi	8
1.5 Importanza del riconoscimento delle fibre nei materiali compositi ...	9
Capitolo 2 Deep learning e segmentazione semantica	11
2.1 Deep Learning	11
2.1.1 Confronto tra Deep Learning e Machine learning.....	11
2.2 Reti neurali artificiali, la base del funzionamento del Deep Learning	14
2.2.1 Addestramento di un sistema di Deep Learning	16
2.3 Applicazioni del Deep Learning	19
2.4 Segmentazione semantica	21
2.4.1 Differenza rispetto alla rilevazione di oggetti.....	21
2.4.2 Funzionamento e architettura.....	22
2.4.3 Applicazioni	24
Capitolo 3 Applicazione ai materiali compositi	26

3.1 Ottenimento immagini e materiali analizzati.....	26
3.1.1 Funzionamento tomografia XCT	28
3.2 Tipo di rete utilizzata.....	29
3.2.1 Reti pre-addestrate e ResNet-18	31
3.3 Labeling.....	33
3.4 Training	36
3.5 Risultati allenamento.....	39
Capitolo 4 Conclusioni.....	43
Appendice 1.....	45
Appendice 2.....	46
Appendice 3.....	48
Appendice 4.....	49
Elenco delle figure	50
Bibliografia e sitografia	52

Introduzione

Il presente progetto di tesi nasce per illustrare il lavoro eseguito durante l'esperienza di tirocinio, svoltasi presso il Dipartimento di Ingegneria Industriale e Scienze Matematiche dell'Università Politecnica delle Marche. Lo scopo principale di questo lavoro è stato quello di applicare un approccio di *deep learning*¹ per l'identificazione delle fibre nei materiali compositi, tramite l'utilizzo della tecnica di *segmentazione semantica*². I materiali compositi costituiscono una vasta classe di materiali largamente impiegati in diversi settori industriali, grazie alle loro proprietà peculiari e alla leggerezza. Tuttavia, un aspetto critico per garantire la qualità e l'affidabilità del prodotto finale è la corretta identificazione e classificazione delle fibre che li compongono. In questo contesto, si rende necessario l'impiego della segmentazione semantica e del processo di Deep Learning al fine di migliorare la precisione e l'efficienza delle procedure di analisi e ispezione. L'utilizzo di tali tecnologie permette di ottenere risultati più accurati e affidabili, favorendo un migliore controllo della produzione e delle prestazioni dei materiali compositi. In conclusione, l'adozione della segmentazione semantica e del deep learning rappresenta un passo significativo verso il miglioramento dei processi industriali e la promozione dell'innovazione nei settori che fanno largo uso di materiali compositi.

Nel primo capitolo del presente elaborato vengono delineati i concetti chiave relativi ai materiali compositi. In particolare, si discute della loro struttura, delle loro proprietà intrinseche, delle applicazioni di rilevanza e dell'importanza connessa al riconoscimento accurato delle fibre che costituiscono tali materiali.

¹ Metodo di intelligenza artificiale (IA) che insegna ai computer a elaborare i dati in un modo che si ispira al cervello umano.

² Algoritmo di deep learning che associa un'etichetta o una categoria a ogni pixel di un'immagine.

Il secondo capitolo si concentra sull'approfondimento del Deep Learning e, più specificamente, sulla tecnica di segmentazione semantica. Nella prima sezione, si espone la distinzione fondamentale tra Deep Learning e Machine Learning, con una particolare attenzione alle reti neurali, che costituiscono il fulcro del Deep Learning, e al processo di addestramento delle stesse. Inoltre, si esaminano le molteplici applicazioni del Deep Learning. Nella seconda parte si procede all'analisi dettagliata della tecnica di segmentazione semantica. Vengono esposti i principi fondamentali del suo funzionamento, nonché le architetture tipiche utilizzate per implementarla in contesti pratici. Si offre inoltre un'ampia visione delle applicazioni in cui questa tecnica può risultare di grande utilità.

Il terzo capitolo costituisce una trattazione approfondita degli aspetti relativi al lavoro svolto nell'ambito di questo elaborato. Si inizia con la descrizione del metodo adottato per ottenere le immagini oggetto di studio, per poi focalizzarsi sull'architettura specifica della rete neurale utilizzata. Un'attenzione particolare è rivolta al processo di "labeling", una fase cruciale per l'addestramento della rete. Il capitolo si conclude con un'analisi dettagliata del codice utilizzato per il training della rete e la presentazione dei risultati ottenuti in questa fase di sperimentazione.

L'elaborato si conclude con un breve accenno alle possibili evoluzioni e applicazioni future dei metodi e delle tecniche esaminate nel corso dell'analisi condotta.

Capitolo 1

Materiali compositi

1.1 Generalità sui materiali compositi

I materiali compositi rappresentano un notevole avanzamento nel campo della scienza e della tecnologia dei materiali [1], fornendo soluzioni innovative e versatili per soddisfare le crescenti esigenze delle moderne applicazioni industriali e tecnologiche. Questa categoria di materiali si caratterizza per la capacità di combinare in modo sinergico le caratteristiche ottimali di più sostanze mediante l'utilizzo di tecnologie all'avanguardia, il che conferisce loro proprietà fisico-meccaniche nettamente superiori alle singole componenti costituenti, oltre che ai materiali tradizionali. La loro ampia applicabilità in vari settori, come l'aeronautica, l'automobilistica, l'industria navale, l'aerospaziale, la medicina e altro ancora, testimonia l'importanza di tali materiali nell'evoluzione tecnologica e nella risoluzione di complessi problemi ingegneristici. La continua ricerca e sviluppo in questo campo stanno aprendo nuove e promettenti opportunità per la progettazione e la creazione di materiali compositi sempre più avanzati e performanti.

1.2 Struttura dei materiali compositi

Un materiale composito è un materiale eterogeneo, cioè costituito da due o più fasi con proprietà fisiche differenti, il cui risultato è una combinazione che presenta prestazioni notevolmente superiori rispetto alle singole parti costituenti. Solitamente, le diverse fasi presenti in questa classe di materiali sono costituite da elementi diversi, come ad esempio nei compositi di fibra di

carbonio e resina epossidica. Tuttavia, esistono eccezioni in cui le diverse fasi sono costituite dallo stesso materiale [2].

Tipicamente una delle due fasi è di natura discontinua, caratterizzata da maggiore rigidità e resistenza, ed è comunemente denominata “rinforzo”. La seconda, invece, è di natura continua, meno rigida e meno resistente, ed è definita come “matrice”. Nella maggioranza dei casi, si utilizza una matrice polimerica, definendo così il materiale risultante come composito polimerico, in ragione della sua capacità di garantire una bassa densità e una leggerezza complessiva. Le matrici più comunemente impiegate comprendono le resine epossidiche, le resine poliuretaniche, le resine poliammidiche e le resine fenoliche. In alcuni casi, tra rinforzo e matrice, può essere presente un elemento aggiuntivo conosciuto come “interfase”. Le proprietà di un materiale composito derivano dalle caratteristiche dei suoi costituenti, nonché dalla geometria e dalla distribuzione delle fasi. In particolare, la geometria e l’orientamento del rinforzo influenzano l’*anisotropia*³ del materiale [1] [3].

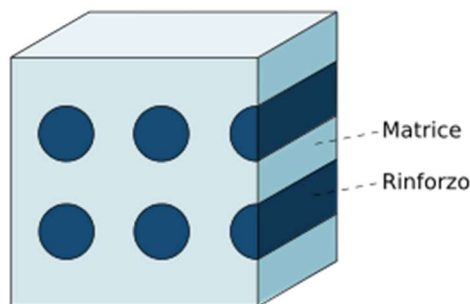


Figura 1.1: tipica struttura di un materiale composito [4].

³ Proprietà per cui il valore di una grandezza fisica, in una sostanza o nello spazio, non è uguale in tutte le direzioni.

Le diverse fasi del composito hanno ruoli differenti a seconda del tipo e dalla messa in opera del materiale. Nel caso di materiali compositi con prestazioni moderate, il rinforzo, solitamente costituito da fibre corte o particelle, contribuisce ad aumentare la rigidità, ma il suo effetto di rafforzamento si limita a livello locale nel materiale. Al contrario, la matrice costituisce il componente principale che sostiene i carichi e definisce le proprietà meccaniche del materiale. Per quanto riguarda i materiali compositi ad elevate prestazioni strutturali, il rinforzo è costituito da fibre continue che formano lo scheletro del materiale, conferendo rigidezza e resistenza nella direzione delle fibre stesse. La matrice svolge il ruolo di proteggere e sostenere le fibre, trasferendo gli sforzi da una fibra all'altra. Nonostante le sue dimensioni ridotte, l'interfase gioca un ruolo significativo nel controllo del meccanismo di rottura, nella resistenza alla frattura e nel comportamento sforzi/deformazioni del materiale [1].

1.2.1 Fibre nei materiali compositi

Le fibre impiegate nei materiali compositi, grazie alle loro dimensioni limitate, presentano un livello di perfezione strutturale eccezionale. Questa particolare caratteristica, unitamente alle proprietà intrinseche dei materiali costituenti, conferisce alle fibre una notevole resistenza meccanica, un elevato modulo elastico e un basso peso specifico. Inoltre, tali fibre manifestano un comportamento elastico lineare fino a rottura.

Le tipologie di fibre più comunemente utilizzate comprendono quelle di vetro, carbonio, fibre organiche e minerali. Esse possono essere integrate nei compositi in diverse configurazioni: sotto forma di filamenti continui allineati parallelamente in un piano, oppure come fibre tagliate con orientamento casuale. In alternativa, possono essere intrecciate in una trama-ordito e disposte in un piano [1].

La selezione delle fibre e delle relative configurazioni esercita un'influenza significativa sulle proprietà meccaniche e strutturali del materiale composito, permettendo di sviluppare materiali con caratteristiche ottimali adatte a specifiche applicazioni industriali e tecnologiche.

1.3 Proprietà dei materiali compositi

Come precedentemente evidenziato, le proprietà di un materiale composito sono influenzate da diversi fattori, tra cui le caratteristiche dei costituenti, la geometria e la distribuzione delle fasi. È importante notare che i materiali compositi implicano costi maggiori rispetto ai materiali tradizionali, tuttavia, grazie ai numerosi vantaggi che offrono, sono stati sempre più adottati dalle aziende in diversi settori industriali. Tra i principali vantaggi si possono citare la leggerezza, la rigidità e la resistenza meccanica, la resistenza a trazione, la versatilità di progettazione e la riduzione del peso [3] [5].

1.4 Esempi di applicazione dei materiali compositi

Le peculiari caratteristiche dei materiali compositi hanno suscitato un ampio interesse e una crescente richiesta nei settori in cui fosse necessario ottenere un basso peso, elevate proprietà meccaniche e la capacità di realizzare forme complesse, aspetti difficilmente raggiungibili tramite altre tecnologie esistenti.

In particolare, le industrie aeronautica, navale e automobilistica fanno largo uso di materiali compositi per la costruzione di diverse componenti quali strutture alari (*figura 1.2*), fusoliere, carrelli, imbarcazioni, canoe, pannelli di carrozzeria, telai delle vetture "Formula 1" (*figura 1.2*), parti di motore e accessori vari. Nell'ambito dell'industria aerospaziale e militare, i materiali compositi sono adoperati per la realizzazione di componenti strutturali di

stazioni di lancio e veicoli spaziali, oltre che per la produzione di caschi e giubbotti antiproiettile. Nei contesti sportivi, i materiali compositi vengono utilizzati per la fabbricazione di sci, racchette da tennis, biciclette, canne da pesca e aste per il salto in alto. Nel settore medico, i materiali compositi trovano applicazione nella costruzione di protesi.

È interessante osservare come la natura stessa presenti esempi di materiali compositi, come il legno, il quale è composto principalmente da una matrice di cellulosa dispersa in lignina (il legno riveste un ruolo significativo come base per la produzione di altri materiali compositi, tra cui il compensato e il legno truciolare). Analogamente, le ossa del corpo umano sono costituite da una matrice di collagene rinforzata con apatite [2] [6].



Figura 1.2: fibra di carbonio nel telaio di una monoposto di Formula 1 e all'interno di una struttura alare [7] [8].

1.5 Importanza del riconoscimento delle fibre nei materiali compositi

L'importanza del riconoscimento delle fibre costituenti i materiali compositi risiede nella stretta correlazione tra le proprietà del materiale e la geometria e disposizione delle fibre al loro interno. L'accurata identificazione delle fibre permette di valutare il grado di allineamento e dispersione delle stesse, ottimizzando così le prestazioni complessive del composito. Inoltre, il processo

di identificazione e classificazione delle fibre riveste un ruolo critico nell'analisi dell'efficienza di fabbricazione e nella valutazione della qualità del prodotto finale, in quanto, eventuali anomalie o difetti riscontrati nelle fibre potrebbero compromettere l'integrità del composito e ridurne le prestazioni.

In conclusione, l'identificazione e il riconoscimento delle fibre nei materiali compositi rappresentano elementi fondamentali per garantire l'affidabilità, la sicurezza e le prestazioni ottimali del materiale stesso, in quanto contribuiscono a migliorare la precisione e l'efficienza delle procedure di analisi e di ispezione. Ciò costituisce una guida essenziale nel processo di progettazione e fabbricazione, promuovendo l'adozione di soluzioni avanzate e innovative. Il riconoscimento delle fibre avviene mediante un processo di deep learning, basato sulla tecnica della segmentazione semantica, come approfondiremo in seguito.

Capitolo 2

Deep learning e segmentazione semantica

2.1 Deep Learning

Il Deep Learning rappresenta un sottocampo del *Machine Learning*⁴ all'interno dell'intelligenza artificiale, caratterizzato dall'utilizzo di algoritmi basati sulla struttura e la funzione del cervello umano, noti come reti neurali artificiali. Tale approccio permette ai sistemi informatici di apprendere dai dati senza intervento umano. Si basa su modelli matematici organizzati in diversi strati di elaborazione, chiamati neuroni, che simulano il funzionamento delle cellule nervose del cervello. Questa tecnica di intelligenza artificiale richiede una maggiore quantità di tempo e risorse per l'addestramento e la generazione di risultati, ma dimostra notevole capacità nell'elaborazione di dati complessi e non strutturati, quali il linguaggio umano e i suoni. Attraverso l'applicazione del Deep Learning, è possibile ottenere una macchina in grado di classificare automaticamente i dati e organizzarli gerarchicamente, identificando quelli più rilevanti e utili per la risoluzione di specifici problemi, similmente a come opera la mente umana. Questo processo di apprendimento continuo consente di migliorare progressivamente le prestazioni del sistema nel tempo [9] [10].

2.1.1 Confronto tra Deep Learning e Machine learning

Machine Learning e Deep Learning sono due tecniche di intelligenza artificiale che permettono ai computer di apprendere da dati e risolvere

⁴ Sottinsieme dell'intelligenza artificiale (AI) che si occupa di creare sistemi che apprendono o migliorano le performance in base ai dati che utilizzano.

problemi complessi. Tuttavia, ci sono alcune differenze fondamentali tra i due approcci, che riguardano il tipo di dati, la struttura degli algoritmi, il grado di intervento umano, le prestazioni ottenute e il livello di scalabilità. Vediamo in dettaglio queste differenze.

Tipo di dati:

- il Machine Learning richiede dati strutturati e categorizzati, cioè dati che hanno una forma definita e che sono etichettati in base a delle classi predefinite;
- il Deep Learning può lavorare con dati non strutturati e non categorizzati, cioè dati che non hanno una forma definita e che non sono etichettati in anticipo. Questo significa che il Deep Learning può gestire dati più complessi e vari, come immagini, audio, testo e video [11].

Struttura degli algoritmi:

- il Machine Learning utilizza algoritmi semplici e lineari, che seguono una sequenza di passi logici per arrivare a una soluzione.
- il Deep Learning utilizza algoritmi complessi e non lineari, che si basano su reti neurali artificiali (modelli matematici ispirati al funzionamento del cervello umano, che sono composti da diversi strati di unità di calcolo chiamate neuroni. Ogni strato riceve in input i dati o i risultati dello strato precedente, li elabora secondo una funzione di attivazione e li trasmette allo strato successivo). Il Deep Learning può quindi apprendere relazioni più sofisticate tra i dati, ma richiede anche più potenza di calcolo e memoria [12] [13].

Grado di intervento umano:

- Il Machine Learning richiede più intervento umano per correggere e apprendere dagli errori. Infatti, il Machine Learning si basa

sull'apprendimento supervisionato, cioè il sistema deve ricevere un riscontro umano che gli indichi le classificazioni o le predizioni corrette. Inoltre, il Machine Learning richiede una fase di preelaborazione dei dati, in cui l'esperto deve selezionare le caratteristiche rilevanti e ridurre le ridondanze.

- Il Deep Learning, invece, richiede meno intervento umano per correggere e apprendere dagli errori. Infatti, il Deep Learning si basa sull'apprendimento non supervisionato, cioè il sistema è in grado di apprendere autonomamente dalle caratteristiche intrinseche dei dati. Inoltre, il Deep Learning non richiede una fase di preelaborazione dei dati, in quanto il sistema è in grado di estrarre automaticamente le caratteristiche rilevanti [11] [12].

Prestazioni ottenute:

Il Machine Learning ha prestazioni inferiori rispetto al Deep Learning quando si tratta di problemi complessi e dati non strutturati. Infatti, il Machine Learning tende a fare correlazioni semplici e lineari tra i dati, che possono essere insufficienti o errate in alcuni casi. Il Deep Learning ha prestazioni superiori rispetto al Machine Learning quando si tratta di problemi complessi e dati non strutturati. Infatti, il Deep Learning è in grado di fare correlazioni complesse e non lineari tra i dati, che possono essere più accurate e robuste in molti casi. Tuttavia, il Deep Learning richiede anche più dati per allenarsi e più tempo per apprendere [11] [13].

Livello di scalabilità:

La scalabilità del Deep Learning, favorita dall'incremento dei dati disponibili e dall'avanzamento degli algoritmi, costituisce una delle sue principali distinzioni rispetto al Machine Learning. Nel contesto dei sistemi di Deep Learning, la scalabilità spesso è agevolata dalla struttura profonda delle reti neurali,

consentendo una maggiore comprensione delle sfumature presenti nei dati all'aumentare della loro complessità. In questo caso, l'incremento di dati di addestramento si traduce in un miglioramento delle prestazioni della rete neurale. Al contrario, i sistemi di Machine Learning possono giungere a uno stadio di saturazione, nel quale l'ulteriore inserimento di dati di addestramento non comporta miglioramenti significativi delle prestazioni. Questo fenomeno può originarsi dal fatto che tali algoritmi dipendono dalle caratteristiche manualmente selezionate durante la fase di progettazione e possono pertanto raggiungere il loro limite di apprendimento [9].

In conclusione, il Machine Learning e il Deep Learning sono due tecniche di intelligenza artificiale che hanno diversi vantaggi e svantaggi a seconda del tipo di problema da risolvere. Il Machine Learning è più adatto per problemi semplici e dati strutturati, mentre il Deep Learning è più adatto per problemi complessi e dati non strutturati. Entrambe le tecniche sono in continua evoluzione e offrono molte opportunità per lo sviluppo di applicazioni innovative in vari settori.

2.2 Reti neurali artificiali, la base del funzionamento del Deep Learning

Le reti neurali artificiali sono modelli computazionali che si ispirano al funzionamento delle reti neurali biologiche riscontrabili nel cervello e nel sistema nervoso umano. Questi modelli sono composti da unità fondamentali denominate neuroni artificiali, i quali ricevono segnali in input (*ingresso*) provenienti da altre unità o da fonti esterne, sottoponendoli a processi di elaborazione al fine di generare nuovi segnali in output (*uscita*). I collegamenti tra i neuroni artificiali sono definiti da pesi sinaptici, i quali determinano il grado di importanza dei segnali in ingresso. L'elaborazione all'interno di

ciascun neurone artificiale è condotta attraverso una funzione di attivazione, la quale determina se il neurone debba trasmettere un segnale in output ad altre unità o meno. L'organizzazione dei neuroni artificiali avviene in strati, che si suddividono in tre categorie: strato di input, strato nascosto e strato di output. I dati da analizzare vengono accolti dallo strato di input, l'elaborazione si verifica negli strati nascosti e il risultato finale è prodotto dallo strato di output. Man mano che si procede attraverso gli strati, le funzionalità diventano via via più complesse rispetto agli strati precedenti, fino al raggiungimento del segnale di output.

La configurazione delle reti neurali può prevedere uno o più strati nascosti, a seconda della complessità dell'obiettivo da raggiungere. Di fatto, una rete neurale tradizionale presenta uno o due strati nascosti, mentre una rete di deep learning potrebbe includere centinaia di strati nascosti.

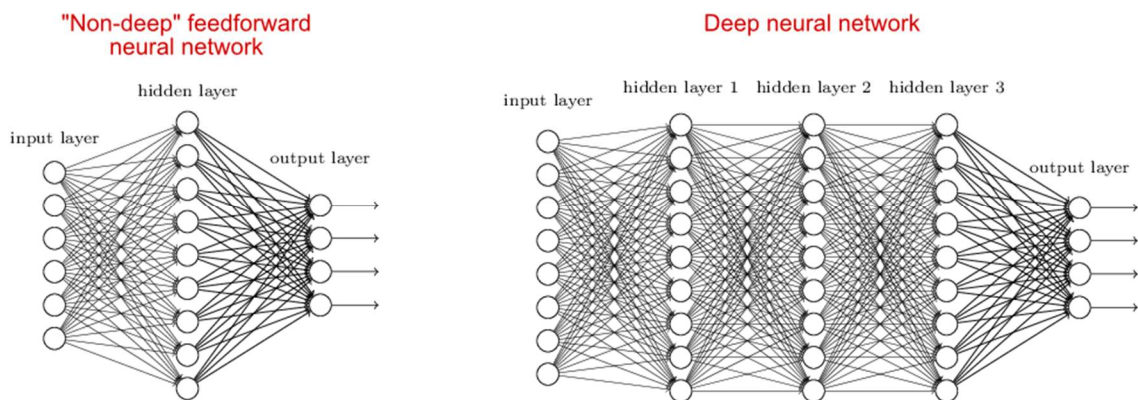


Figura 2.1: architettura di una rete neurale tradizionale (a sinistra) e di una rete neurale profonda (a destra) [9].

L'operatività delle reti neurali artificiali si basa sul processo di apprendimento, che mira a regolare i parametri della rete per minimizzare l'errore tra l'output desiderato e quello effettivamente prodotto, utilizzando un set di dati noto come dati di addestramento. In tal senso, sono disponibili diversi approcci di addestramento, tra i quali l'algoritmo di retro-propagazione dell'errore è uno dei più noti. Questo algoritmo comporta il calcolo dell'errore nei nodi di output e

la sua propagazione all'indietro verso l'input, al fine di aggiornare i pesi in modo da minimizzare l'errore complessivo.

Un esempio rilevante di rete neurale artificiale, utilizzato nel contesto del deep learning, è rappresentato dalla rete neurale convoluzionale (CNN). Questo tipo di rete è costituito da strati convoluzionali, che eseguono operazioni matematiche, chiamate convoluzioni, sui dati in ingresso per estrarre informazioni strutturali e morfologiche sottoforma di una mappa delle caratteristiche, e strati di pooling, che riducono il numero di parametri in input, mediante funzioni di aggregazione. Le reti convoluzionali dimostrano particolare efficacia nell'analisi di dati strutturati come le immagini, poiché sono in grado di catturare relazioni spaziali tra pixel e proprietà locali delle immagini.

Le reti neurali artificiali hanno dimostrato una vasta gamma di vantaggi e applicazioni in vari settori scientifici e tecnologici. Tra questi si annoverano la loro capacità di approssimare funzioni complesse non lineari, di generalizzare da esempi limitati, di adattarsi a contesti dinamici e incerti e di scoprire relazioni sottese nei dati. Nonostante tali meriti, le reti neurali artificiali presentano anche limitazioni e svantaggi, quali la richiesta di ingenti quantità di dati per l'apprendimento, la difficoltà nell'interpretare il loro funzionamento interno e l'assenza di garanzie formali sulla loro correttezza e affidabilità [9] [14] [15] [16].

2.2.1 Addestramento di un sistema di Deep Learning

L'addestramento dei sistemi di deep learning costituisce un elemento essenziale all'interno del processo di sviluppo di modelli capaci di apprendere ed estrarre rappresentazioni significative da dati di natura complessa. Questa procedura comporta l'ottimizzazione dei pesi all'interno delle reti neurali

profonde attraverso l'esposizione a dati di addestramento. Gli approcci principali in questa disciplina sono rappresentati dall'addestramento supervisionato e da quello non supervisionato.

Nell'addestramento supervisionato, i modelli di deep learning vengono presentati con dati di addestramento che comprendono sia le informazioni di input che le corrispondenti uscite (output) desiderate. L'obiettivo principale consiste nell'indurre il modello a imparare una relazione tra i dati di input e i rispettivi output, in modo tale da consentire la previsione accurata di risultati per nuovi dati. Il processo di addestramento si compone di una fase iniziale di propagazione in avanti, in cui la rete neurale elabora i dati di addestramento, generando previsioni (output). Queste previsioni vengono poi confrontate, nella fase successiva, con i valori di riferimento (target) associati ai dati di addestramento. L'errore risultante, nella fase di propagazione all'indietro, viene "retro-propagato" attraverso i vari strati della rete, comportando la modifica dei pesi sinaptici in modo da minimizzare l'errore. Gli aggiornamenti dei pesi sinaptici sono calcolati mediante algoritmi di ottimizzazione, che regolano l'entità e la direzione dei cambiamenti in base alla discrepanza tra le previsioni della rete e i valori di riferimento. Queste fasi di propagazione in avanti, calcolo dell'errore, propagazione all'indietro e ottimizzazione dei pesi vengono iterativamente ripetute attraverso numerosi cicli (epoche), coinvolgendo l'intero set di dati di addestramento. Tale iterazione progressiva permette alla rete di perfezionare gradualmente le sue capacità, riducendo l'errore nelle previsioni. Una volta completato l'addestramento, la rete viene valutata su dati precedentemente non incontrati, noti come dati di validazione e test. Questo passaggio consente di valutare l'efficacia della rete nel formulare previsioni precise su nuovi input. Esempi di contesti di addestramento supervisionato includono la classificazione di immagini, il riconoscimento vocale e l'analisi del sentiment.

L'addestramento non supervisionato si concentra invece sulla manipolazione di dati sprovvisti di etichette, ossia privi di informazioni riguardo alle uscite desiderate associate a ciascun esempio di input. In tale scenario, l'obiettivo fondamentale è identificare strutture intrinseche e modelli nascosti all'interno dei dati senza l'ausilio di etichette di output. Tra gli approcci nell'ambito dell'addestramento non supervisionato spiccano il clustering e la riduzione della dimensionalità. Nel clustering, il modello cerca di aggregare i dati in gruppi omogenei in base alle somiglianze delle loro caratteristiche. Invece, nella riduzione della dimensionalità, il modello si propone di proiettare i dati in uno spazio di minori dimensioni, conservando allo stesso tempo le informazioni salienti. Questi approcci trovano ampio impiego nell'analisi dei dati e nella scoperta di strutture nascoste [10] [17] [18] [19].

In sintesi, l'addestramento dei sistemi di deep learning può adottare sia approcci supervisionati che non supervisionati, a seconda della disponibilità di etichette e degli obiettivi dell'applicazione. Mentre l'addestramento supervisionato mira a far apprendere alla rete relazioni tra input e output noti, l'addestramento non supervisionato è focalizzato sulla rivelazione di strutture latenti nei dati. Entrambi questi approcci rivestono un ruolo cruciale nello sviluppo di modelli di deep learning sofisticati e nell'individuazione di informazioni utili all'interno di dati complessi. L'addestramento di tali sistemi richiede una considerevole quantità di dati, potenza computazionale, algoritmi efficienti e una scelta oculata dell'architettura della rete.

2.3 Applicazioni del Deep Learning

La tecnica del Deep Learning si è dimostrata estremamente efficace in una vasta gamma di campi applicativi, grazie alla sua capacità di estrarre automaticamente caratteristiche rilevanti dai dati e di adattarsi a modelli complessi. Di seguito vengono riportati alcuni dei principali ambiti di applicazione del deep learning, accompagnati da esempi specifici.

Visione artificiale: il deep learning è stato impiegato con successo nell'analisi di immagini e video. Esempi includono la rilevazione di oggetti, il riconoscimento facciale, la **segmentazione semantica** (che comporta, per esempio, l'assegnazione di etichette alle varie parti costituenti un'immagine) e la generazione di immagini.

Elaborazione del linguaggio naturale: il deep learning permette di analizzare e generare testi, con applicazioni come la traduzione automatica, il riconoscimento vocale, la sintesi vocale, la comprensione del testo, la generazione di testi e la conversazione.

Classificazione predittiva: l'uso del Deep Learning consente la previsione del comportamento o dell'esito di fenomeni, con impieghi significativi nell'ambito della diagnosi medica, delle previsioni meteorologiche e finanziario, con l'analisi delle tendenze di mercato, il rilevamento di frodi e la previsione dei prezzi degli asset.

Guida autonoma: nel settore automobilistico, il deep learning è fondamentale per lo sviluppo di sistemi di guida autonoma. Le reti neurali vengono addestrate per riconoscere segnali stradali, veicoli, pedoni e altre informazioni cruciali per una guida sicura.

Scienze ambientali: l'adozione del Deep Learning si dimostra di grande rilevanza nell'analisi di dati complessi provenienti da sensori ambientali, come

ad esempio l'interpretazione di immagini satellitari per il monitoraggio delle foreste, dei cambiamenti climatici e della qualità dell'aria e dell'acqua.

Cinema: in ambito cinematografico, il Deep Learning trova impiego nella creazione di effetti speciali realistici, come la sostituzione dei volti degli attori, la modifica delle espressioni facciali, la creazione di ambienti virtuali e la generazione di personaggi animati [9].

Tuttavia, nonostante le nuove frontiere che il Deep Learning continua a esplorare, è essenziale riconoscere che si tratta di un campo ancora in evoluzione che affronta alcune limitazioni. Tra queste, si annovera l'esigenza di un cospicuo volume di dati per giungere a risposte esaustive, la poca flessibilità (dato che le macchine stanno ancora apprendendo in modi molto limitati, il che può portare a errori. Le reti di deep learning necessitano di dati per risolvere un problema specifico. Se viene richiesto di eseguire un task al di fuori di tale ambito, è probabile che non riesca), e la mancanza di trasparenza (poiché comprendere il processo decisionale di una rete neurale può risultare arduo, rendendo complicato individuare eventuali distorsioni indesiderate e spiegare le predizioni).

Nonostante questi ostacoli, gli specialisti del settore stanno costantemente progredendo verso la realizzazione di modelli di Deep Learning estremamente accurati e capaci di apprendere autonomamente. Tale sviluppo contribuirà significativamente all'avanzamento di numerose discipline, consentendo soluzioni innovative ed automatizzate per la risoluzione di una vasta gamma di problematiche complesse [14]

2.4 Segmentazione semantica

La segmentazione semantica è un algoritmo di deep learning utile per l'analisi di immagini, che associa un'etichetta o una categoria ad ogni pixel di un'immagine. Viene utilizzata per riconoscere una serie di pixel che formano categorie distinte, in modo da ottenere una comprensione dettagliata del contenuto dell'immagine. Ad esempio, un veicolo a guida autonoma deve identificare veicoli, pedoni, segnali stradali, marciapiedi e altri elementi della strada [20]. La segmentazione semantica è un problema difficile da risolvere, in quanto richiede una comprensione globale dell'immagine e una capacità di distinguere tra oggetti simili e sovrapposti. Inoltre, deve essere in grado di gestire variazioni di scala, forma, posa, illuminazione e sfondo degli oggetti.

2.4.1 Differenza rispetto alla rilevazione di oggetti

La segmentazione semantica si differenzia dalla rilevazione di oggetti in quanto effettua una suddivisione dell'immagine pixel per pixel, assegnando etichette semantiche a ciascun elemento presente. Tale metodologia consente una precisa identificazione degli oggetti di forma irregolare, a differenza della rilevazione di oggetti dove è necessario circoscrivere gli stessi all'interno di un rettangolo di delimitazione [20].

La segmentazione semantica può essere considerata come una forma più dettagliata e precisa di rilevazione di oggetti, poiché consente di ottenere contorni accurati sia per gli oggetti che per le loro parti costituenti. Tuttavia, è essenziale notare che la segmentazione semantica manifesta alcune limitazioni in confronto alla rilevazione degli oggetti, tra le quali:

- complessità computazionale maggiore: la segmentazione semantica richiede una quantità superiore di risorse computazionali e tempo per processare

un'immagine rispetto alla rilevazione di oggetti. Questa maggiore complessità deriva dal fatto che ogni singolo pixel deve essere analizzato, contrariamente alla rilevazione di oggetti che si focalizza solo su specifiche regioni di interesse;

- difficoltà nell'ottenere dati annotati: la segmentazione semantica richiede dei dati etichettati a livello di pixel, che sono più onerosi e laboriosi da produrre rispetto ai dati etichettati a livello di regione;
- mancanza di distinzione tra istanze: la segmentazione semantica attribuisce la medesima etichetta a tutti i pixel appartenenti alla stessa classe, senza effettuare una distinzione tra diverse istanze dello stesso tipo di oggetto. Per ovviare a questo problema, esiste una variante della segmentazione semantica chiamata segmentazione per istanze. Quest'ultima, oltre a classificare i pixel, assegna un identificatore univoco a ciascuna istanza di oggetto individuata.

2.4.2 Funzionamento e architettura

Le moderne metodologie di segmentazione semantica spesso si basano su reti neurali profonde, in particolare sulle reti neurali convoluzionali (CNN). Queste reti apprendono automaticamente caratteristiche rilevanti dalle immagini attraverso il processo di addestramento su ampi dataset annotati. Un'architettura comune utilizzata per la segmentazione semantica è l'encoder-decoder, in cui l'encoder costituisce la parte della rete neurale che esegue il sotto-campionamento, mentre il decoder è la parte che attua il sovra-campionamento. Encoder e decoder collaborano per estrarre informazioni semantiche dall'input e generare una mappa di previsione dettagliata, agevolando una segmentazione accurata delle diverse classi presenti nell'immagine.

Il processo di sotto-campionamento si occupa di ridurre la dimensione dell'immagine in ingresso e di aumentare la profondità delle caratteristiche estratte. Questo avviene tramite strati di convoluzione, i quali applicano dei filtri ai pixel dell'immagine per estrarre le caratteristiche più rilevanti (come bordi, angoli, colori, texture ecc.), seguiti da strati di pooling, che riducono la risoluzione delle mappe delle caratteristiche (matrici bidimensionali contenenti i valori dei pixel dopo l'applicazione dei filtri) attraverso l'applicazione di funzioni di aggregazione su una regione locale dei pixel. La parte di encoder può essere basata su una rete neurale pre-addestrata su un grande dataset di immagini, come AlexNet, VGG-16 o GoogLeNet, per sfruttare le conoscenze già apprese da questi modelli.

Il processo di sovra-campionamento è responsabile di aumentare le dimensioni dell'immagine in uscita e di ridurre la profondità delle caratteristiche estratte. Ciò viene realizzato impiegando strati di deconvoluzione, che applicano dei filtri ai pixel delle mappe delle caratteristiche per creare nuove mappe con una risoluzione maggiore, seguiti da degli strati di unpooling, che ripristinano la risoluzione originale delle mappe delle caratteristiche utilizzando le informazioni memorizzate dagli strati di pooling corrispondenti. La parte decoder può essere basata su una rete neurale simmetrica alla parte encoder, oppure su una rete neurale più semplice che utilizza solo degli strati deconvoluzione [20].

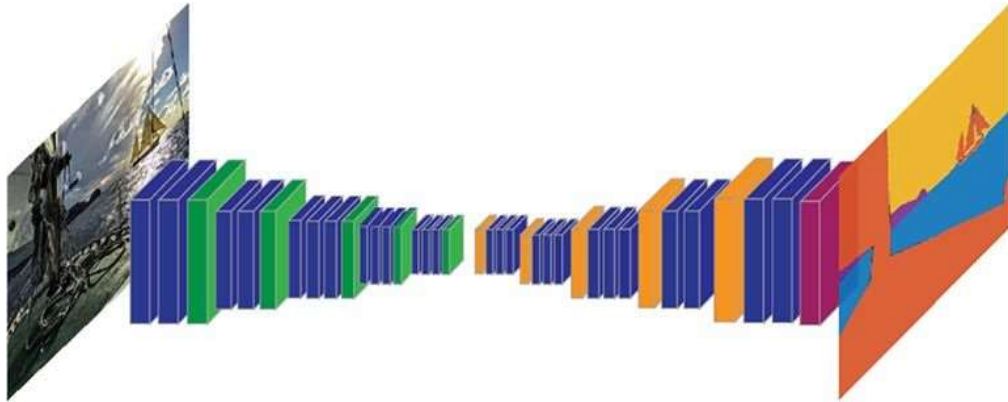


Figura 2.2: la CNN esegue funzioni relative all'immagine su ogni layer; quindi esegue il sottocampionamento dell'immagine utilizzando un livello di pooling (in verde). Questo processo viene ripetuto più volte per la prima metà della rete. L'output della prima metà di questo diagramma è seguito da una quantità uguale di layer di unpooling (in arancione) [20].

2.4.3 Applicazioni

Come evidenziato in precedenza, la segmentazione semantica è un processo che permette una maggiore precisione rispetto ad altre modalità di rilevazione di oggetti. Questa peculiarità conferisce all'approccio fondamentale importanza nelle sue applicazioni all'interno di settori in cui vengono richieste mappe immagine accurate, quali ad esempio:

- guida autonoma: nel contesto di veicoli a guida autonoma, la segmentazione semantica è necessaria per identificare strade, marciapiedi, pedoni, segnali stradali e ulteriori componenti, consentendo pertanto al veicolo di comprendere approfonditamente l'ambiente circostante e prendere decisioni appropriate;



Figura 2.3: esempio di applicazione della segmentazione semantica ad un veicolo a guida autonoma [20].

- diagnostica medica: nell'ambito della diagnostica medica, la segmentazione semantica trova impiego nell'individuazione e nell'isolamento di specifiche strutture anatomiche o regioni di rilevanza all'interno di immagini mediche, come ad esempio risonanze magnetiche o tomografie computerizzate;
- videosorveglianza: l'applicazione della segmentazione semantica può risultare particolarmente vantaggiosa nel campo della videosorveglianza. In questo scenario, l'approccio può essere adoperato per monitorare aree pubbliche, riconoscendo movimenti umani e veicolari e individuando comportamenti potenzialmente sospetti;
- agricoltura: nel contesto dell'agricoltura moderna, la segmentazione semantica rappresenta uno strumento d'analisi rilevante. Il suo utilizzo si concretizza nell'analisi delle coltivazioni al fine di identificare aree suscettibili di problematiche, tra cui patologie vegetali e carenze nutrizionali [20].

Capitolo 3

Applicazione ai materiali compositi

3.1 Materiali analizzati e ricostruzione tomografica

Per ottenere immagini utili all'individuazione e classificazione delle fibre all'interno dei materiali compositi, le tecniche non distruttive (NDT) si sono dimostrate lo strumento più adatto. Questo insieme di metodologie comprende esami, prove e rilievi volti all'analisi delle strutture materiali, senza alterarle o richiedere l'asportazione di campioni. Il loro obiettivo primario consiste nell'individuazione e nell'identificazione di difetti strutturali all'interno del materiale stesso [21].

Nel nostro contesto, abbiamo impiegato la tecnica di tomografia computerizzata a raggi X (XCT). L'adozione della tomografia computerizzata come metodologia di ispezione non distruttiva consente di ottenere immagini digitali tridimensionali di un oggetto, compresa la sua struttura interna. Queste immagini permettono la caratterizzazione interna dei materiali, inclusa la valutazione delle porosità, la rilevazione di difetti quali vuoti o crepe e l'esecuzione di controlli di qualità non distruttivi. In tal modo, la XCT, e in generale le NDT, si rivelano strumenti essenziali per la localizzazione e l'acquisizione di informazioni fondamentali in merito alla natura tridimensionale dei materiali, inclusa la loro struttura interna [22].

Nonostante nell'ultimo decennio si sia verificato un notevole incremento delle capacità della tomografia computerizzata a raggi X (XCT), alcune sfide rimangono presenti anche per questa tecnica. In particolare, il volume degli oggetti sottoposti a scansione è spesso di dimensioni limitate in confronto alle

architetture e ai componenti compositi. Nel caso di scansione di componenti di dimensioni maggiori, si può riscontrare un abbassamento del contrasto nelle immagini risultanti, il che rende più complessa la distinzione tra la resina e le fibre del materiale composito [23].

Lo studio è stato condotto su due provini di stesse dimensioni ma composti da materiali differenti. Entrambi i campioni sono costituiti da materiali compositi di tipo CM, caratterizzati da una matrice in resina rinforzata, rispettivamente, da fibre di vetro (GFRP) e lino (FRP) forniti dall'azienda HP Composites S.p.A company. La matrice polimerica utilizzata per entrambi i provini è la resina epossidica termoindurente. I campioni sono stati ottenuti ritagliando un laminato formato da cinque strati di materiale pre-impregnato, sia nel caso del GFRP che del FRP. Le dimensioni dei provini sono le seguenti: 100 mm x 10 mm x 2.8 mm per il GFRP e 100 mm x 10 mm x 2.7 mm per il FRP.

Le principali proprietà dei materiali sono riassunte nella seguente tabella:

	Fibre	Resina	Densità delle fibre	Densità della resina	Contenuto di fibra/resina
GFRP	E-glass	Epoxy ER450	2.58 g/cm^3	1.2 g/cm^3	70-30%
FRP	Flax	Epoxy ER450	1.45 g/cm^3	1.2 g/cm^3	50-50%

Tabella 1: caratteristiche tecniche provini analizzati.

3.1.1 Acquisizioni tomografiche

La tomografia XCT è una tecnica di scanning diagnostica che utilizza i raggi X per ottenere sezioni trasversali dell'oggetto scansionato. Il funzionamento del tomografo si basa sul principio dell'attenuazione dei raggi X, cioè la diminuzione della loro intensità quando attraversano la materia. Perciò la XCT permette di ricostruire la densità del provino scansionato, attraverso sezioni 2D perpendicolari all'asse del sistema di acquisizione [24].

La prima fase del funzionamento di un tomografo XCT consiste nella generazione dei raggi X. Tali raggi, emessi dalla sorgente, attraversano il campione d'interesse, posizionato su un piatto rotante tra la sorgente stessa e un rivelatore. Il rivelatore registra l'attenuazione subita dai raggi X nel loro attraversamento dell'oggetto. Le misurazioni dell'attenuazione dei raggi X da diverse angolazioni vengono quindi utilizzate per generare proiezioni bidimensionali dell'oggetto in esame. Successivamente, mediante l'impiego di algoritmi matematici sofisticati, le proiezioni bidimensionali vengono utilizzate al fine di creare una rappresentazione tridimensionale dell'oggetto in questione. Le immagini risultanti, relative alle sezioni perpendicolari all'asse del sistema di acquisizione, sono di natura digitale e consistono in una matrice quadrata di elementi noti come pixel. Ciascun pixel assume un valore specifico all'interno della scala di grigi, contribuendo a delineare con precisione le caratteristiche dell'oggetto sottoposto a tomografia XCT.

La *figura 3.1* mostra due immagini acquisite mediante la tecnica di tomografia XCT. È evidente come nell'immagine relativa alla fibra di vetro, *figura 3.1 (a)*, sia possibile effettuare una distinzione nitida tra le fibre orientate orizzontalmente, quelle orientate verticalmente e la resina circostante. D'altra parte, nell'immagine corrispondente alla fibra di lino, *figura 3.1 (b)*, risulta più complesso separare chiaramente le fibre dalla resina.

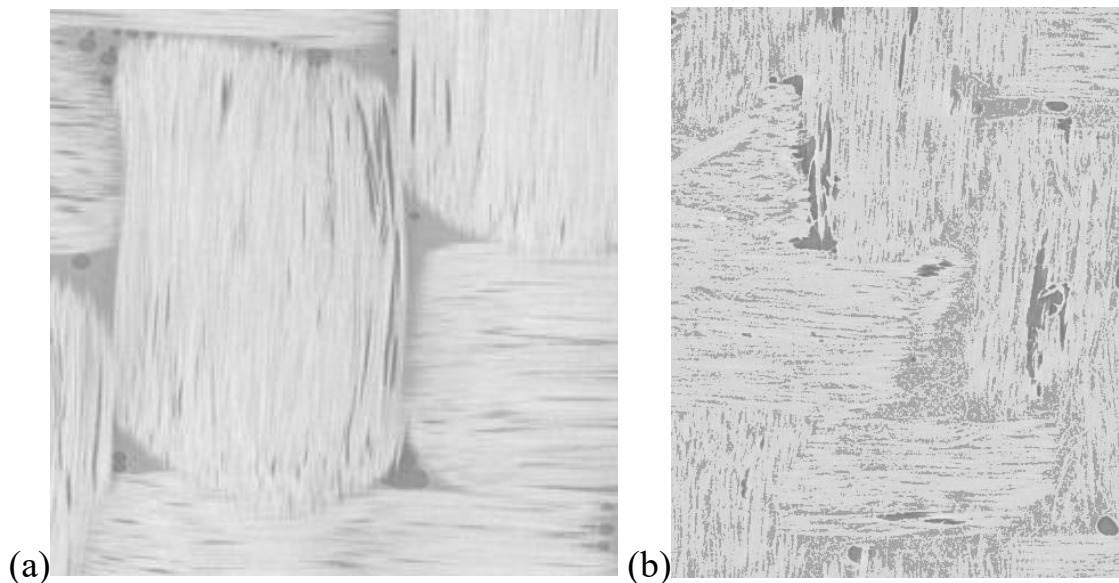


Figura 3.1: immagini in scala di grigi della fibra di vetro (a) e di lino (b).

3.2 Tipo di rete utilizzata

In questo studio, è stata impiegata una rete neurale convoluzionale profonda, nota come DCNN (Deep Convolutional Neural Network), denominata DeepLab v3+. Questa rete si basa su una rete pre-addestrata di tipo ResNet18. La DeepLab v3+ rappresenta un tipo di rete neurale convoluzionale concepita per la segmentazione semantica delle immagini. Tale architettura sfrutta un approccio encoder-decoder, incorpora convoluzioni dilatate e integra collegamenti skip (ovvero connessioni dirette tra i diversi strati della rete. Vengono utilizzati per integrare informazioni più dettagliate provenienti da livelli inferiori con informazioni di contesto più ampio da livelli superiori) per condurre il processo di segmentazione delle immagini. È opportuno sottolineare che, prima di poter impiegare la rete per effettuare la segmentazione semantica, è necessario procedere con il suo addestramento [25].

La DCNN fa affidamento su operazioni di convoluzione, che possono essere concepite come il movimento di un filtro o kernel sull'immagine di input o sulle

mappe delle caratteristiche generate da uno strato precedente. Tipicamente, in ogni strato di convoluzione, sono presenti più filtri, e man mano che la DCNN si approfondisce, si utilizzano un numero crescente di filtri. Questa pratica genera un maggior numero di caratteristiche, migliorando così la capacità di segmentare fasci di fibre con orientamenti diversi. L'utilizzo di più filtri in uno stesso strato permette di estrarre simultaneamente diverse caratteristiche.

Un aspetto rilevante è l'espansione dei campi visivi dei filtri convoluzionali all'aumentare della profondità della DCNN. Ciò è di notevole importanza poiché le caratteristiche estratte possono fornire informazioni multi-scala. Inoltre, la DCNN non solo estrae caratteristiche multi-scala del materiale, ma è anche in grado di combinare informazioni provenienti da diversi livelli attraverso la concatenazione delle mappe delle caratteristiche a diversi livelli, preservando così il contesto informativo. Queste caratteristiche delle DCNN le conferiscono un vantaggio significativo rispetto ai metodi tradizionali, migliorando la precisione e la versatilità dell'analisi.

In generale, una DCNN è costituita da diversi strati convoluzionali, attivazioni non lineari, normalizzazione per batch (lotti) e strati di pooling, con il livello convoluzionale che costituisce il cuore dell'architettura. I livelli convoluzionali eseguono operazioni di convoluzione sulle immagini di input per estrarre informazioni strutturali e morfologiche sotto forma di mappe delle caratteristiche. Le attivazioni non lineari mappano i valori al di sotto di una soglia predefinita su zero, mentre gli strati di pooling semplificano gli input sostituendoli con statistiche riassuntive del loro contesto. L'operazione di convoluzione può essere concepita come il movimento di un filtro o kernel sull'immagine di input, con la successiva moltiplicazione elemento per elemento del filtro con la regione sovrapposta dell'immagine di input.

Le mappe delle caratteristiche risultanti vengono ulteriormente elaborate da ciascuno strato della rete, culminando nella produzione delle mappe di segmentazione nello strato finale. La dimensione e il numero totale di filtri in uno strato convoluzionale sono predefiniti, ma gli elementi dei filtri, noti come pesi, vengono determinati o "appresi" durante il processo di addestramento. Tale processo è matematicamente un problema di ottimizzazione in cui i pesi della rete vengono regolati per minimizzare una funzione di perdita, utilizzando appositi algoritmi.

L'utilizzo di approcci di deep learning richiede un notevole impegno nella fase di etichettatura manuale delle immagini e nella preparazione dei dati. Tuttavia, una volta addestrata la rete, l'elaborazione dei dati diventa quasi istantanea, consentendo un notevole aumento dell'efficienza nel processo di progettazione virtuale [26].

3.2.1 Reti pre-addestrate e ResNet-18

Nel contesto della classificazione di immagini è possibile prendere una rete neurale pre-addestrata che ha precedentemente imparato ad estrarre caratteristiche potenti e informative dalle immagini e utilizzarla come punto di partenza per apprendere una nuova attività.

La maggior parte delle reti neurali pre-addestrate viene addestrata su un sottoinsieme del database ImageNet, ovvero un database di immagini organizzato secondo la gerarchia WordNet (ampio database lessicale dell'inglese. Sostantivi, verbi, aggettivi e avverbi sono raggruppati in insiemi di sinonimi cognitivi (synset), ciascuno dei quali esprime un concetto distinto. I synset sono collegati tra loro mediante relazioni concettuali-semantiche e

lessicali), in cui ogni nodo della gerarchia viene rappresentato da centinaia o migliaia di immagini.

È possibile utilizzare le reti neurali precedentemente addestrate per le seguenti attività:

- classificazione: è possibile applicare direttamente reti neurali pre-addestrate per risolvere problemi di classificazione delle immagini;
- estrazione di feature: È fattibile utilizzare una rete neurale pre-addestrata per estrarre caratteristiche rilevanti dalle immagini, sfruttando le attivazioni dei livelli intermedi come feature;
- transfer learning: si può procedere all'acquisizione dei livelli di una rete neurale addestrata su un vasto insieme di dati e successivamente perfezionare la rete su un nuovo insieme di dati specifici per un nuovo compito.

Nel processo decisionale relativo alla scelta della rete neurale da applicare a un determinato problema, è fondamentale considerare diverse caratteristiche delle reti neurali pre-addestrate. Tra queste caratteristiche, le più rilevanti comprendono la precisione (capacità della rete di effettuare previsioni accurate), velocità (velocità di elaborazione delle immagini) e dimensione della rete (la complessità della rete neurale influisce sulle risorse computazionali richieste e sulla dimensione del modello). In generale, la scelta di una rete neurale costituisce un compromesso tra queste caratteristiche [27].

Nel nostro contesto, la Deep Convolutional Neural Network (DCNN) si basa sull'utilizzo della rete convoluzionale pre-addestrata denominata ResNet-18. La ResNet-18 è una rete convoluzionale che presenta una profondità composta da 18 livelli distinti. Questa rete è stata precedentemente addestrata su un vasto

dataset composto da oltre un milione di immagini provenienti dal database ImageNet e ha dimostrato la capacità di classificare immagini in ben mille categorie diverse. Pertanto, la rete ha appreso rappresentazioni ricche di caratteristiche per un'ampia varietà di immagini.

La funzione principale di questa rete consiste nel prendere un'immagine in ingresso e restituire in uscita un'etichetta che identifica gli oggetti presenti nell'immagine, insieme alla relativa probabilità che tali oggetti appartengano a ciascuna delle categorie precedentemente apprese [28].

3.3 Labeling

Con il termine "labeling" si intende quell'insieme di operazioni che consistono nell'assegnare delle etichette o categorie a dei dati non strutturati, come immagini, audio, video o testo, al fine di renderli utilizzabili da algoritmi di intelligenza artificiale, machine learning o deep learning. Questo processo viene ampiamente utilizzato in diversi contesti, tra cui l'organizzazione dei dati, la categorizzazione degli oggetti, la classificazione dei testi, l'analisi dei dati e l'apprendimento automatico.

Nel contesto dello sviluppo di modelli di machine learning o deep learning, l'etichettatura dei dati rappresenta una fase cruciale nella preelaborazione dei dati. Questa fase comporta l'identificazione dei dati non strutturati e l'attribuzione di una o più etichette per definire il contesto associato a tali dati. Ciò consente al modello di machine learning o di deep learning di fare previsioni accurate e coerenti [29].

Prima di procedere con l'operazione di etichettatura, attraverso un codice MATLAB (vedi *Appendice 1*), è stata eseguita una fase di preelaborazione delle immagini ottenute mediante la tecnica di tomografia XCT. Questa

preelaborazione ha incluso il ritaglio di una porzione delle immagini e l'ottimizzazione del contrasto al fine di migliorare la precisione del processo di segmentazione manuale, e di conseguenza, dell'addestramento della rete neurale. Le immagini così ottenute (visualizzabili nella *figura 3.2 (a)*), sono state sottoposte al processo di etichettatura tramite l'utilizzo di Label Studio.

Label Studio è uno strumento open source dedicato all'etichettatura dei dati, che consente di svolgere una vasta gamma di operazioni, supportando numerosi formati di dati. Inoltre, è possibile integrare Label Studio con modelli di machine learning o deep learning al fine di effettuare previsioni sulle etichette o di condurre un apprendimento continuo [30].

Nel nostro specifico contesto, abbiamo utilizzato la tecnica di "segmentazione semantica con maschere" presente all'interno di Label Studio. Questa modalità ci ha consentito di classificare le diverse categorie di etichette presenti nelle immagini identificando "matrix" per la resina, "warp" per le fibre orientate in direzione orizzontale e "weft" per le fibre orientate in direzione verticale. In seguito, come si può osservare nella *figura 3.2 (b)*, abbiamo effettuato una segmentazione manuale, procedendo quindi a colorare manualmente le diverse aree delle immagini.

Queste immagini segmentate manualmente costituiscono la cosiddetta "verità fondamentale" utilizzata per l'addestramento e la valutazione delle immagini segmentate dalla rete neurale. Nel nostro studio sono state analizzate 100 immagini per le fibre di vetro e 100 per le fibre di lino.

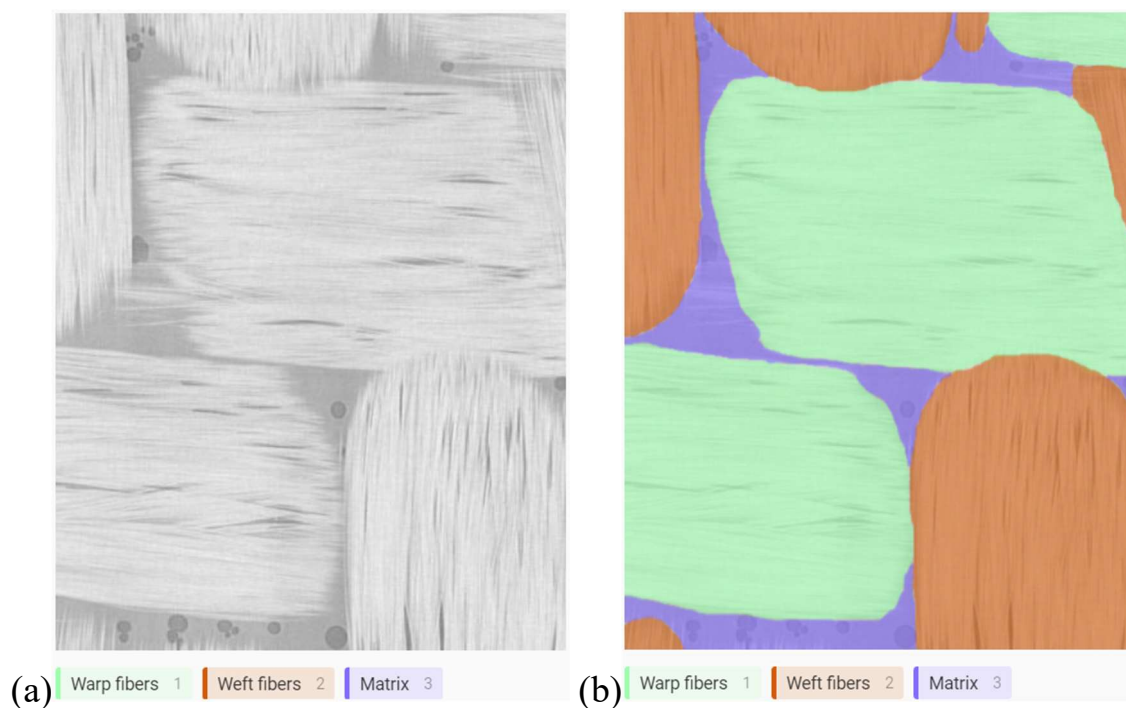


Figura 3.2: (a) fibra di vetro in scala di grigi (ottenuta tramite tomografia XCT). (b) fibra di vetro in cui è stata effettuata la segmentazione manuale nelle tre diverse categorie presenti: verde per le fibre orizzontali, rosso per le fibre verticali e blu per la resina.

Dopo la fase di segmentazione manuale, le immagini sono state esportate in forma di dati binari, suddividendole in tre categorie distinte: una per le fibre orientate orizzontalmente, una per quelle orientate verticalmente e una per la resina. La *figura 3.3* presenta le immagini ottenute in seguito all'applicazione di un codice MATLAB (vedi *Appendice 2*). Tale procedura è stata ripetuta per ogni immagine ed utilizzata per correggere eventuali errori derivanti dalla segmentazione manuale, come ad esempio la presenza di buchi causati da pixel non colorati.

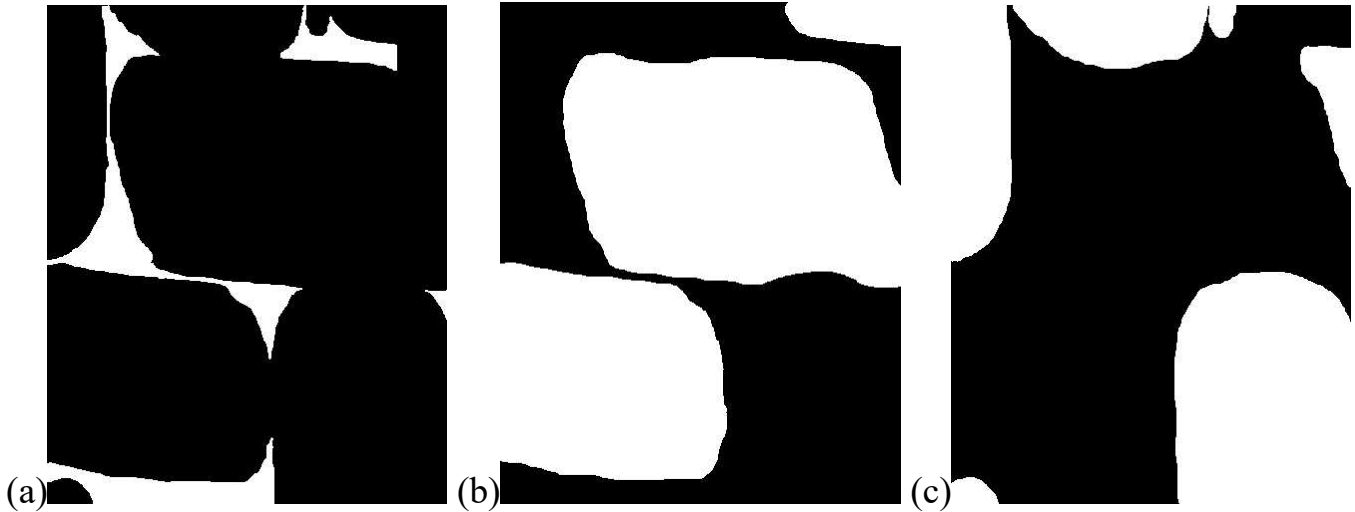


Figura 3.3: immagine in binario della fibra di vetro. (a) resina, (b) fibre orientate orizzontalmente, (c) fibre orientate verticalmente.

3.4 Training

Le immagini ottenute dopo le operazioni di etichettatura manuale e l'applicazione del codice descritto nell'*appendice 2*, come quelle visibili nella *figura 3.3*, sono state impiegate nel processo di addestramento della rete neurale. Tale processo di addestramento è stato condotto mediante un codice MATLAB, il quale è documentato nell'*appendice 3*.

Il codice per l'addestramento della rete neurale per la classificazione delle immagini è composto da diverse fasi. Di seguito, verrà spiegata ogni parte in dettaglio:

1. Creazione del Datastore delle Immagini in scala di grigi:

il codice inizia creando un datastore di immagini in scala di grigi utilizzando la funzione "imagedatastore". Questo datastore è destinato a contenere le immagini che verranno utilizzate per l'addestramento. Le immagini sono in scala di grigi, come quelle mostrate nella *Figura 3.1*.

2. Creazione del pixel label datastore:

successivamente, viene creato un secondo datastore chiamato "Pixel Label Datastore" utilizzando la funzione "pixelLabelDatastore". Questo datastore contiene immagini binarie con le relative etichette. Ogni pixel in ciascuna immagine è associato a un'etichetta che indica a quale classe appartiene. Nel nostro contesto, abbiamo due classi: "warp" e "background". La funzione "pixelLabelDatastore" richiede i seguenti input: location, ovvero una lista di cartelle o file di immagini specificati come un vettore o una matrice; classNames, una lista delle classi di etichette, che nel nostro caso sono "warp" e "background"; pixelLabelIDs, ID utilizzati per mappare le etichette delle classi ai ClassNames [31].

3. Creazione della Rete Neurale DeepLab v3+ basata su ResNet-18:

è stata utilizzata la funzione "deeplabv3plusLayer" per creare una rete neurale DeepLab v3+ basata su ResNet-18. Questa funzione restituisce un livello di DeepLab v3+ con parametri specificati come le dimensioni delle immagini, il numero di classi e la rete di base [32].

4. Combinazione dei datastore:

la funzione "combine" viene utilizzata per combinare i due datastore creati in precedenza. Questo passaggio è essenziale per unire i dati delle immagini con quelli delle etichette dei pixel per l'addestramento.

5. Configurazione delle opzioni di addestramento:

è stata utilizzata la funzione "trainingOptions" per definire le configurazioni di addestramento della rete neurale. In particolare, è stato specificato l'algoritmo di ottimizzazione denominato "SGDM" (stochastic gradient descent with momentum), il quale ha come obiettivo principale quello di accelerare la convergenza del processo di addestramento, e quindi far sì che la rete neurale

raggiunga una soluzione migliore e più rapidamente rispetto alla discesa del gradiente standard. L'algoritmo di discesa del gradiente standard aggiorna i parametri della rete (pesi) per minimizzare la funzione di perdita, spostandosi di poco ad ogni iterazione nella direzione del gradiente negativo della perdita. Nell'algoritmo di discesa del gradiente standard, il gradiente della funzione di perdita viene valutato usando l'intero set di addestramento e l'algoritmo usa subito l'intero set di dati. Invece, ad ogni iterazione, l'algoritmo di discesa stocastica del gradiente valuta il gradiente e aggiorna i parametri utilizzando un sottoinsieme dei dati di addestramento. Ad ogni iterazione viene usato un sottoinsieme diverso, chiamato mini-batch. Un passaggio completo dell'algoritmo di addestramento lungo l'intero set di addestramento usando mini-batch è un'epoca. l'algoritmo di discesa stocastica del gradiente può oscillare lungo il percorso di discesa più ripida verso il valore ottimale. Aggiungere un termine di momento (algoritmo di discesa stocastica del gradiente con momento) all'aggiornamento dei parametri è un modo per ridurre questa oscillazione. Le dimensioni del mini-batch e il numero massimo di epoche possono essere specificati usando rispettivamente le opzioni di addestramento "MiniBatchSize" e "MaxEpochs". Il parametro "MiniBatchSize" specifica le dimensioni del mini-batch da utilizzare per ogni iterazione di addestramento. In altre parole, quante immagini vengono utilizzate contemporaneamente per calcolare il gradiente e aggiornare i pesi. Il parametro "MaxEpochs" rappresenta il numero massimo di epoche che verranno eseguite durante l'intero processo di addestramento. Un'epoca rappresenta un passaggio completo attraverso l'intero set di addestramento utilizzando il mini-batch. Nel nostro caso abbiamo impostato una dimensione del mini-batch pari a 8 e un numero massimo di epoche pari a 10 (il che significa che l'addestramento verrà eseguito per un massimo di 10 volte attraverso l'intero set di dati) [33].

6. Addestramento della rete neurale:

Infine, la funzione "trainNetwork" viene utilizzata per addestrare la rete neurale. I suoi input includono la combinazione dei due datastore creati in precedenza (chiamati "images"), i layer della rete neurale precedentemente creati (chiamati "layers"), e le opzioni di addestramento definite in precedenza (chiamate "options"). Questo passaggio addestra la rete neurale per la classificazione e la regressione delle immagini utilizzando le immagini e le etichette dei pixel, secondo le opzioni specificate.

3.5 Risultati allenamento

La funzione "trainNetwork", relativa al codice MATLAB presentato nell'appendice 3, ci ha consentito di ottenere i grafici rappresentati nella *figura 3.4* per quanto riguarda la fibra di vetro, e nella *figura 3.5* per la fibra di lino. Questi grafici includono diversi parametri che ci permettono di effettuare una valutazione preliminare dei risultati dell'allenamento della rete neurale. Tra questi parametri figurano:

- *epoch*: numero di epoche, in cui un'epoca corrisponde ad un passaggio completo dei dati;
- *iteration*: numero di iterazioni, in cui ciascuna iterazione corrisponde ad un mini-batch;
- *time elapsed*: indica il tempo trascorso;
- *mini-batch accuracy*: precisione della classificazione sul mini-batch all'iterazione specificata, ovvero la percentuale di esempi correttamente classificati nel corrente mini-batch di dati di addestramento;

- *mini-batch loss*: perdita sul mini-batch. Si riferisce alla misura della funzione di perdita, calcolata su un mini-batch. La funzione di perdita è una misura dell'errore tra le previsioni della rete e i valori corretti. La sua finalità è quella di valutare quanto bene la rete sta apprendendo durante l'allenamento. Il calcolo viene eseguito su ciascun mini-batch durante ogni iterazione dell'allenamento;
- *base learning rate*: velocità di apprendimento di base. Il software moltiplica i fattori della velocità di apprendimento dei livelli per questo valore.

Questi parametri svolgono un ruolo cruciale nell'analisi e nell'ottimizzazione dell'addestramento della rete neurale per ottenere risultati desiderati.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:10	54.99%	0.7851	0.0100
5	50	00:08:13	97.36%	0.0645	0.0100
9	100	00:16:32	98.60%	0.0350	0.0100
10	120	00:19:54	98.76%	0.0313	0.0100

Figura 3.4: risultati ottenuti dalla funzione "trainNetwork", applicata alle immagini relative alle fibre orientate verticalmente della fibra di vetro.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:09	46.92%	0.8961	0.0100
5	50	00:08:17	95.34%	0.1140	0.0100
9	100	00:16:41	96.82%	0.0816	0.0100
10	120	00:20:04	96.47%	0.0862	0.0100

Figura 3.5: risultati ottenuti dalla funzione "trainNetwork", applicata alle immagini relative alle fibre orientate verticalmente della fibra di lino.

Un aspetto di notevole importanza da sottolineare è l'osservazione che la percentuale di accuratezza aumenta con l'incremento del numero di iterazioni, mentre il valore del mini-batch loss diminuisce. Ciò ci suggerisce che la rete neurale, all'aumentare del numero di epoche, migliora progressivamente le sue capacità di effettuare previsioni sempre più accurate.

Un ulteriore elemento di valutazione dei risultati dell'addestramento della rete, è stato ottenuto mediante il calcolo dell'indice IOU (Intersection Over Union), comunemente noto come indice Jaccard (*Appendice 4*). L'IOU rappresenta il rapporto fra l'area di sovrapposizione tra immagine predetta e immagine corrispondente al risultato atteso e l'area di unione tra le due aree come illustrato nella *figura 3.6*. Il valore di questo indice varia da 0 a 1, dove 0 indica l'assenza di sovrapposizione tra le aree delle due immagini e 1 indica una perfetta sovrapposizione [34].

Si tratta di una misura cruciale per valutare quanto bene la rete neurale è in grado di identificare e delimitare le regioni d'interesse rispetto alle immagini attese.

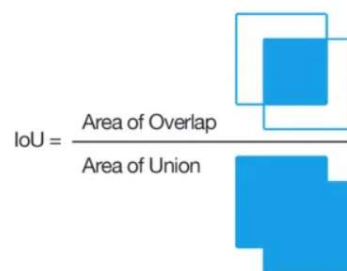

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Figura 3.6: IOU, Intersection Over Union o indice di Jaccard [34].

Nella *tabella 2* sono rappresentati i valori medi dell'indice IOU ottenuti sia per la fibra di vetro che per la fibra di lino. Emerge come risultato rilevante che,

nel caso della fibra di lino, l'indice IOU manifesti valori inferiori. Questa situazione può essere interpretata come un indicatore di una minore precisione nella segmentazione dei dati relativi alla fibra di lino.

	Vetro	Lino
Fibre verticali	0,831	0,631
Fibre orizzontali	0,867	0,821

Tabella 2: valori dell'indice IOU per la fibra di vetro e di lino.

Come ultimo passo, abbiamo effettuato il calcolo del numero di pixel bianchi, che corrispondono alle fibre nelle immagini, confrontando le immagini relative ai risultati attesi e quelle ottenute dall'allenamento della rete neurale. Da tale analisi è emerso che le immagini generate dalla rete addestrata presentano approssimativamente l'84.12% di pixel bianchi ovvero di fibre su tutta l'immagine, mentre le immagini relative ai risultati attesi mostrano una percentuale del 92.96% di pixel bianchi. Si osserva quindi uno scostamento percentuale pari all'8.84% tra le due tipologie di immagini. È importante notare che tale procedimento è stato eseguito esclusivamente per le immagini relative alla fibra di vetro, poiché dai risultati precedenti abbiamo potuto constatare una maggiore precisione della rete neurale addestrata per questo tipo di immagini.

Capitolo 4

Conclusioni

L'obiettivo principale del presente studio consisteva nell'addestrare un modello di deep learning al fine di condurre il riconoscimento e la classificazione delle fibre all'interno dei materiali compositi. Questo processo si è articolato in diverse fasi, partendo dalla generazione delle immagini mediante tomografia computerizzata XCT, seguita da un'elaborazione delle immagini e un'etichettatura manuale. Successivamente, si è proceduto con la fase di addestramento della rete neurale.

Dai risultati ottenuti emerge che il processo di addestramento è stato più efficace nel caso delle fibre di vetro. Questa discrepanza di successo è attribuita alla difficoltà nell'effettuare una distinzione netta tra le fibre di lino orientate verticalmente, quelle orientate orizzontalmente e la resina nelle immagini in scala di grigi. Ciò ha comportato risultati di addestramento che non sono stati in perfetta congruenza con le aspettative. In contrasto, nel caso delle immagini delle fibre di vetro, la distinzione tra fibre orientate verticalmente, fibre orientate orizzontalmente e resina era notevolmente più evidente, risultando in una maggiore coerenza tra i risultati attesi e quelli ottenuti.

Pertanto, possiamo concludere che per migliorare l'accuratezza del modello è fondamentale che le immagini di partenza presentino un contrasto adeguato, al fine di evitare le difficoltà riscontrate nel caso delle immagini delle fibre di lino. Un ulteriore passo per aumentare la precisione del modello consiste nell'incrementare la dimensione del dataset di immagini di addestramento. Ciò permette alla rete neurale di apprendere da un numero più ampio di immagini, acquisendo così una maggiore comprensione delle diverse caratteristiche di queste ultime. Tuttavia, è importante sottolineare che un dataset più ampio

comporta un aumento dei tempi necessari per l'elaborazione manuale delle immagini e, soprattutto, per il processo di addestramento.

Questo studio potrebbe servire come punto di partenza per sviluppi futuri in tutti quei settori in cui il riconoscimento delle fibre all'interno dei materiali compositi riveste un ruolo cruciale.

Appendice 1

```
data = permute(Data_organized,[3,2,1]);

D_cropped = imread('ritaglioprovaloop30.tif');
b1 = mean2(D_cropped);
% for k=1:200
for j=1:100
    C = data(:,:,500+j);
    C_cropped = imcrop(C,[350 550 400 500]);
    b2 = mean2(C_cropped);
    NewImage2 = C_cropped + (b1-b2);%seleziono l'area da ritagliare
da imtool
    %figure, imshow(C_cropped)
    imwrite(NewImage2,sprintf('ritaglioprovaloopvetro%d.tif',j));
end
% end

%
% imgMean = mean(C_cropped,'all');
% imgStd = std(double(C_cropped(:)));
% refImgMean = mean(D_cropped,'all');
% refImgStd = std(double(D_cropped(:)));
%
% adjustedImg = imadjust(C_cropped, [imgMean-3*imgStd
imgMean+3*imgStd]/65535, [refImgMean-3*refImgStd
refImgMean+3*refImgStd]/65535);

b1 = mean2(D_cropped);
b2 = mean2(C_cropped);
NewImage2 = C_cropped + (b1-b2);
```

Appendice 2

```
file = uigetfile('.png','Select a file','MultiSelect', 'on'); %png
format perchè le immagini annotate esportate da label studio sono
in questo formato
%seleziono tutte le matrix, un task alla volta
Add_matrix = imread(file); % se solo una maschera per classe è
presente in un determinato task
a = {};
for k = 1:length(file)
    a{k} = imread(file{1,k});
end

Add_matrix = a{1};
for k = 2:length(file)
    Add_matrix = imadd(a{k}, Add_matrix);
end

figure, imshow(Add_matrix)
Add_matrix(Add_matrix>210) = 255;
figure, imshow(Add_matrix)
Add_matrix = imcomplement(Add_matrix);
%Add_matrix = ~ Add_matrix;
Add_matrix = bwareaopen(Add_matrix,30);
Add_matrix = imcomplement(Add_matrix);
figure, imshow(Add_matrix)
Add_matrix = imfill(Add_matrix,'holes');
figure, imshow(Add_matrix)
imwrite(Add_matrix, sprintf('matrix17.png')); % cambio ogni volta
il numero nel nome dell'immagine

% below, the same for warp
clear all
close all

file = uigetfile('.png','Select a file','MultiSelect', 'on');
%seleziono tutte le warp
Add_warp = imread(file); % se solo una maschera per classe è
presente in un determinato task
a = {};
for k = 1:length(file)
    a{k} = imread(file{1,k});
end

Add_warp = a{1};
for k = 2:length(file)
    Add_warp = imadd(a{k}, Add_warp);
end
```

```

figure, imshow(Add_warp)
Add_warp(Add_warp>210) = 255;
figure, imshow(Add_warp)
Add_warp = imcomplement(Add_warp);
Add_warp = bwareaopen(Add_warp,30);
Add_warp = imcomplement(Add_warp);
figure, imshow(Add_warp)
%Add_warp = imfill(Add_warp,'holes');
%figure, imshow(Add_warp)
imwrite(Add_warp ,sprintf('warp29.png'));

% below, the same for weft
clear all
close all

file = uigetfile('.png','Select a file','MultiSelect', 'on');
%seleziono tutte le weft
Add_weft = imread(file); % se solo una maschera per classe è
presente in un determinato task
a = {};
for k = 1:length(file)
    a{k} = imread(file{1,k});
end

Add_weft = a{1};
for k = 2:length(file)
    Add_weft = imadd(a{k}, Add_weft);
end

figure, imshow(Add_weft)
Add_weft(Add_weft>0) = 255;
figure, imshow(Add_weft)
Add_weft= imcomplement(Add_weft);
Add_weft = bwareaopen(Add_weft,30);
Add_weft = imcomplement(Add_weft);
figure, imshow(Add_weft)
Add_weft = imfill(Add_weft,'holes');
figure, imshow(Add_weft)
imwrite(Add_weft ,sprintf('weft24.png'));
currentFolder = pwd;

```


Appendice 3

```
dataDir = fullfile('C:\Users\INSITER\Desktop\label-studio');
imDir = fullfile(dataDir, 'scala_di_grigi_vetro');
imds = imageDatastore(imDir);
pxDir = fullfile(dataDir, 'prova');
classNames = ["warp", "background"];
LabelIDs = [1 0];
pxds = pixelLabelDatastore(pxDir, classNames, LabelIDs);
imageSize = [256 256];
numClasses = numel(classNames);
lgraph = deeplabv3plusLayers(imageSize, numClasses, 'resnet18');
cds = combine(imds, pxds);
opts = trainingOptions('sgdm', ...
    'MiniBatchSize', 8, ...
    'MaxEpochs', 10);
net = trainNetwork(cds, lgraph, opts);
```

Appendice 4

```
II = imread('im35.jpg');
CC = semanticseg(II,net);
BB = labeloverlay(II,CC);
figure, imshow(BB)

expectedResult = imread('bo35.tif');
actual = uint8(CC);
expected = uint8(expectedResult);
figure, imshowpair(actual, expected);

actual(actual==1)=0;
actual(actual==2)=1;
actual=logical(actual);
expected = logical(expected);
expected = imcomplement(expected);

iou = jaccard(actual,expected);
table(classNamees,iou)

actual = imcomplement(actual);
imwrite(actual,sprintf('weftlino35.tif'));
```

Elenco delle figure

1.1	<i>tipica struttura di un materiale composito [4].</i>	6
1.2	<i>fibra di carbonio nel telaio di una monoposto di Formula 1 e all'interno di una struttura alare [7] [8].</i>	9
2.1	<i>architettura di una rete neurale tradizionale (a sinistra) e di una rete neurale profonda (a destra) [9].</i>	15
2.2	<i>la CNN esegue funzioni relative all'immagine su ogni layer; quindi esegue il sottocampionamento dell'immagine utilizzando un livello di pooling (in verde). Questo processo viene ripetuto più volte per la prima metà della rete. L'output della prima metà di questo diagramma è seguito da una quantità uguale di layer di unpooling (in arancione) [20].</i>	24
2.3	<i>esempio di applicazione della segmentazione semantica ad un veicolo a guida autonoma [20].</i>	25
3.1	<i>immagini in scala di grigi della fibra di vetro (a) e di lino (b).</i>	29
3.2	<i>(a) fibra di vetro in scala di grigi (ottenuta tramite tomografia XCT). (b) fibra di vetro in cui è stata effettuata la segmentazione manuale nelle tre diverse categorie presenti: verde per le fibre orizzontali, rosso per le fibre verticali e blu per la resina.</i>	35
3.3	<i>immagine in binario della fibra di vetro. (a) resina, (b) fibre orientate orizzontalmente, (c) fibre orientate verticalmente.</i>	36

3.4	<i>risultati ottenuti dalla funzione "trainNetwork", applicata alle immagini relative alle fibre orientate verticalmente della fibra di vetro.....</i>	40
3.5	<i>risultati ottenuti dalla funzione "trainNetwork", applicata alle immagini relative alle fibre orientate verticalmente della fibra di lino.....</i>	40
3.6	<i>IOU, Intersection Over Union o indice di Jaccard [34].....</i>	41

Bibliografia e sitografia

- [1] «INFOBUILD. Materiali compositi, cosa sono e come vengono prodotti,» [Online]. Available: <https://www.infobuild.it/approfondimenti/materiali-compositi/>.
- [2] «definizione di materiale composito,» [Online]. Available: https://it.wikipedia.org/wiki/Materiale_composito.
- [3] «Gavazzi Tessuti Tecnici - cosa sono i materiali compositi» [Online]. Available: <https://www.gavazzispa.it/blog/cosa-sono-i-materiali-compositi>.
- [4] «Wikipedia - struttura di un materiale composito,» [Online]. Available: https://it.wikipedia.org/wiki/Materiale_composito.
- [5] «wirutex - caratteristiche e proprietà materiali compositi» [Online]. Available: <https://blog.wirutex.com/it/cosa-sono-materiali-compositi>.
- [6] «SkyScooter - le applicazioni dei materiali compositi» [Online]. Available: <https://www.skyscooter.it/2017/01/22/le-applicazioni-dei-materiali-compositi/>.
- [7] «HEXAGON - materiali compositi nel settore aerospaziale,» [Online]. Available: <https://hexagon.com/it/resources/resource-library/composite-materials-aerospace>.
- [8] «motorsport.com - fibra di carbonio nelle vetture di formula 1,» [Online]. Available: <https://it.motorsport.com/f1/news/f1-ferrari-stesso-fondo-ma-con-direzione-delle-fibre-diversa/10323184/>.

- [9] «AI 4Business - deep learning» [Online]. Available: <https://www.ai4business.it/intelligenza-artificiale/deep-learning/deep-learning-cose/>.
- [10] «MAC Formazione - differenza tra Machine Learning e Deep Learning» [Online]. Available: <https://www.macformazione.com/qual-e-la-differenza-tra-deep-learning-e-machine-learning/>.
- [11] «Digital Guide IONOS - deep learning vs machine learning» [Online]. Available: <https://www.ionos.it/digitalguide/online-marketing/marketing-sui-motori-di-ricerca/deep-learning-vs-machine-learning/>.
- [12] «Coursera - machine learning e deep learning» [Online]. Available: <https://www.coursera.org/articles/ai-vs-deep-learning-vs-machine-learning-beginners-guide>.
- [13] «unite.ai - differenze chiave tra machine learning e deep learning» [Online]. Available: <https://www.unite.ai/it/differenze-chiave-tra-machine-learning-e-deep-learning/>.
- [14] «Oracle Cloud Infrastructure (OCI),» [Online]. Available: <https://www.oracle.com/it/artificial-intelligence/machine-learning/what-is-deep-learning/#deep-learning-neutral-networks>.
- [15] «IBM - deep learning» [Online]. Available: <https://www.ibm.com/it-it/topics/deep-learning>.
- [16] «Digitalic - reti neurali» [Online]. Available: <https://www.digitalic.it/hardware-software/reti-neurali>.
- [17] «Medium» [Online]. Available: <https://medium.com/@jacopokahl/i-tre-principali-tipi-di-machine-learning-77ca20d0dbdd>.

- [18] «Capterra - modelli di machine learning» [Online]. Available: <https://www.capterra.it/blog/2523/modelli-machine-learning-supervisionato-o-no>.
- [19] «intelligenzaartificiale.it,» [Online]. Available: <https://www.intelligenzaartificiale.it/machine-learning/>.
- [20] «MathWorks - semantic segmentation» [Online]. Available: <https://it.mathworks.com/solutions/image-video-processing/semantic-segmentation.html>.
- [21] S.Gholizadeh, «A review of non-destructive testing methods of composite materials,» *Procedia Structural Integrity*, vol. 1, pp. 50-57, 2016.
- [22] Emilia Sabau. et al, «A Novel Polymer Concrete Composite with GFRP Waste: Applications, Morphology, and Porosity Characterization,» *Applied sciences*, vol. 10, n. 6, 2020.
- [23] S.C. Garcea et al, «X-ray computed tomography of polymer composites,» *Composites Science and Technology*, vol. 156, pp. 305-319, 2018.
- [24] Pavel. Iassonov et al, «Segmentation of X-ray computed tomography images of porous materials: A crucial step for characterization and quantitative analysis of pore structures,» *WATER RESOURCES RESEARCH*, vol. 45, n. 9, 2009.
- [25] MATLAB, «deeplabv3plusLayers - MathWorks,» [Online]. Available: <https://it.mathworks.com/help/vision/ref/deeplabv3pluslayers.html>.
- [26] Muhammad. A. Ali. et al, «Deep learning based semantic segmentation of μ CT images for creating digital material twins of fibrous reinforcements,» *Composites Part A: Applied Science and Manufacturing*, vol. 139, 2020.

- [27] MATLAB, «Reti neurali profonde preaddestrate - MathWorks,» [Online]. Available: <https://it.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html>.
- [28] MATLAB, «Rete neurale convoluzionale ResNet-18 - MathWorks,» [Online]. Available: <https://it.mathworks.com/help/deeplearning/ref/resnet18.html>.
- [29] «IBM, data labeling,» [Online]. Available: <https://www.ibm.com/it-it/topics/data-labeling>.
- [30] Label Studio, «What is Label Studio.,» [Online]. Available: https://labelstud.io/guide/get_started.
- [31] «MATLAB - pixelLabelDatastore,» [Online]. Available: <https://it.mathworks.com/help/vision/ref/pixellabeldatastore.html>.
- [32] «MATLAB - deeplabv3plusLayers,» [Online]. Available: <https://it.mathworks.com/help/vision/ref/deeplabv3pluslayers.html>.
- [33] «MATLAB - trainingOptions,» [Online]. Available: <https://it.mathworks.com/help/deeplearning/ref/trainingoptions.html>.
- [34] E. Tiu, «Towards data science - Metrics to Evaluate your Semantic Segmentation Model,» [Online]. Available: <https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2>.