



POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Informatica (Computer Engineering)

Tesi di Laurea

Classificazione automatica di piante sane o malate con uso di immagini

Immagini collezionate con l'utilizzo di droni

Relatori

prof. Maurizio Morisio

prof. Luca Ardito

Candidato

Matteo CERUTTI

ANNO ACCADEMICO 2020-2021

Indice

1	Introduzione	5
2	Valutazione di droni e applicazioni di volo automatico	7
2.1	Introduzione	7
2.2	Valutazione di droni	7
2.2.1	Criteri di selezione	7
2.2.2	Tabella comparativa	8
2.2.3	Commenti	8
2.2.4	Conclusioni	13
2.2.5	Scelta d'acquisto	13
2.3	Valutazione di applicazioni per la definizione di piani di volo	13
2.3.1	Volo automatico e autonomo	13
2.3.2	Criteri di valutazione delle applicazioni	14
2.3.3	Applicazioni per la pianificazione di voli automatici	14
2.3.4	Scelta finale	18
2.4	Tabella comparativa droni	19
2.5	Tabella comparativa payload	24
3	Collezione e gestione delle immagini	27
3.1	Riprese delle immagini	27
3.1.1	Sopralluoghi preliminari	27
3.1.2	Definizione dei piani di volo	27
3.1.3	Riprese delle immagini	28
3.2	Repository di salvataggio	29
3.2.1	Locazione	29
3.2.2	Struttura	29
3.2.3	Metadati delle immagini	30

3.2.4	Formato dei dati aggiuntivi	31
3.2.5	File di supporto	33
3.3	Dataset per l'analisi di classificazione	33
3.3.1	Estrazione automatica del dataset	33
4	Analisi di classificazione delle immagini	36
4.1	Selezione di lavori di ricerca simili in letteratura	36
4.1.1	Modalità di ricerca	36
4.1.2	Lavori di ricerca selezionati	36
4.2	Modelli di rete neurale selezionati	39
4.2.1	AlexNet	40
4.2.2	GoogLeNet	40
4.2.3	VGG	41
4.3	Strumenti utilizzati	42
4.3.1	Google Colaboratory	42
4.3.2	Pytorch	44
4.4	Logica di analisi delle immagini	44
4.4.1	Grid search e iperparametri perfezionati	44
4.4.2	K-fold cross validation	45
4.4.3	Modelli pre-allenati	46
4.4.4	Metriche di valutazione	46
4.4.5	Suddivisione del dataset	47
4.5	Risultati e osservazioni	47
4.5.1	Grid search	47
4.5.2	Testing	48
5	Conclusioni	57
	Bibliografia	59

Capitolo 1

Introduzione

Nell'ambito della coltivazione di piante da frutteto, un fattore fondamentale per garantire un tasso positivo della propria produzione è quello riguardante la gestione delle malattie che possono colpire le piante e il tempestivo intervento per la cura o per il contenimento della stesse.

Il lavoro di questa tesi ha l'obiettivo d'identificare automaticamente eventuali piante con problematiche attraverso un modello predittivo in grado di analizzare immagini collezionate con riprese effettuate da droni. Questo permetterebbe un intervento anticipato e mirato sulle piante individuate come possibilmente alterate, utilizzando un mezzo rapido e comodo per scattare le immagini.

In questo elaborato verranno analizzate le prime fasi del progetto a partire dal suo principio, durante il periodo 2020-2021.

La prima fase è composta da un'analisi di mercato avente come obiettivo la ricerca di un drone adatto allo scopo del caso di studio, che quindi sia in grado di scattare immagini RGB di buona qualità e immagini multispettrali, cioè immagini che contengono i dati specifici riguardanti una sola banda multispettrale dello spettro visibile, come ad esempio il blu o il rosso (Capitolo 2).

In questa parte, oltre alla descrizione dei criteri utilizzati per effettuare la valutazione, vengono presentati differenti modelli di droni con le proprie caratteristiche, al fine di giungere alla conclusione di selezionare un modello adatto allo scopo con l'obiettivo di acquistarlo in seguito.

Inoltre è stata effettuata una ricerca di un'applicazione di supporto utile alla definizione di piani di volo automatici, al fine di poter automatizzare la collezione delle immagini durante le riprese in campo aperto e garantire una costanza nelle inquadrature delle immagini scattate, oltre a una riproducibilità futura delle riprese.

Il periodo di tempo in cui queste analisi sono state effettuate è compreso fra Dicembre 2020 e Febbraio 2021. In seguito a queste analisi, viene descritta poi la scelta effettuata per il progetto di studio sia a livello di modello di drone scelto, sia a livello di applicazione per la definizione di voli automatici selezionata e utilizzata.

Successivamente sono stati definiti i dettagli che riguardano le riprese effettuate durante il periodo fra Maggio e Luglio del 2021, descrivendo innanzitutto le modalità con cui esse sono state effettuate e sulle tipologie di dati raccolti (Capitolo 3). Infatti durante i momenti di ripresa, un gruppo di esperti nel settore botanico ha effettuato una serie di campionamenti fisiologici e patologici al fine di poter definire una pianta come malata o sana da entrambi i punti di vista, nello specifico periodo di tempo.

In seguito viene presentato il modo di gestione e salvataggio di queste immagini all'interno di un repository, la cui struttura gerarchica ben definita ha permesso di raccogliere le immagini e una serie di dati di supporto al fine di agevolare sia la fase successiva di analisi, sia la futura continuazione del lavoro. Nell'elaborato, infatti, vengono descritti gli script utili alla generazione di questi dati, fra cui le convenzioni che permettono di mappare le piante alle proprie immagini scattate durante differenti riprese, con lo scopo di poter tracciare una cronologia visiva della

pianta. Un obiettivo ulteriore di questa fase è quello di poter replicare in futuro altre riprese nel modo più utile ai fini dell'analisi, seguendo quindi le convenzioni definite.

Nella parte finale di questo capitolo viene descritto come è stato generato il dataset utile ad allenare gli algoritmi d'intelligenza artificiale di classificazione descritti nel capitolo successivo.

La fase seguente tratta la parte di progetto riguardante l'analisi delle immagini raccolte nelle riprese del 2021, al fine di poter delineare la definizione di modelli in grado di classificare le immagini fra rappresentanti piante malate oppure piante sane.

Nel capitolo 4, dopo la descrizione dei lavori simili in letteratura a questo progetto di tesi, seguita dalla spiegazione della selezione dei modelli di rete neurale e della loro architettura, vengono descritte le fasi eseguite per trovare la versione migliore dei modelli scelti per lo scopo: il trattamento dei dati al fine di essere utilizzabili correttamente dagli algoritmi, la ricerca dei iperparametri migliori descriventi gli stessi, il training dei modelli, utilizzando i valori ricercati e il test dei modelli risultanti, utilizzando il dataset di test estratto prima della fase di training.

Per effettuare queste operazioni sono stati anche utilizzati alcuni strumenti che vengono descritti in modo dettagliato, cioè la piattaforma di esecuzione utilizzata, Google Colab, e le librerie di deep learning che hanno permesso un'agevolazione in fase d'implementazione della logica: Pytorch e Torchvision.

I modelli di rete neurale convoluzionale scelti sono stati: AlexNet, GoogLeNet e VGG-11. Le motivazioni di tale scelta sono descritte e approfondite, considerando anche i loro utilizzi nei progetti di ricerca descritti nei documenti selezionati in letteratura.

Nella parte finale del capitolo vengono presentati i dati raccolti durante le fasi di scelta dei migliori iperparametri delle reti e le performance dei modelli durante la fase conclusiva di testing, attraverso l'utilizzo d'immagini significative di riassunto e di comparazione al fine di avere una rappresentazione significativa e chiara delle scelte che si sono effettuate e dell'efficacia dei modelli risultanti.

Nella conclusione dell'elaborato vengono analizzati i risultati ottenuti al fine di poter individuare eventuali possibili sviluppi e approcci futuri, al fine di migliorare gli algoritmi e di conseguenza i risultati, considerando queste analisi e questo lavoro come un punto di partenza verso risultati sempre migliori (Capitolo 5).

Capitolo 2

Valutazione di droni e applicazioni di volo automatico

2.1 Introduzione

Nell'ambito del caso di studio analizzato in questo elaborato, l'elemento del drone è fondamentale nella fase di collezione delle immagini utili alla classificazione automatica finale. Infatti l'automatizzazione è molto utile per poter avere grandi quantità d'immagini, con caratteristiche simili e scattate a un'altezza simile, utili per l'affinamento corretto degli algoritmi d'intelligenza artificiale. Per questo l'analisi delle caratteristiche dei droni permette di poter trovare i modelli maggiormente compatibili con lo scopo sopracitato.

In questo capitolo si parte, quindi, dalla definizione dei criteri di valutazione utilizzati successivamente, per passare alla valutazione dei singoli modelli attualmente disponibili sul mercato. Nella sezione successiva si analizzano, invece, alcune applicazioni di volo, elemento che permette di automatizzare al meglio la collezione delle immagini, potendo definire tracce per il volo automatico dei droni.

Le seguenti valutazioni sono state effettuate nel periodo fra Dicembre 2020 e Febbraio 2021, con alcuni aggiornamenti nei mesi successivi.

2.2 Valutazione di droni

2.2.1 Criteri di selezione

Dimensioni: valori di lunghezza al fine di determinare il volume fisico del drone, espressi attraverso la misura diagonale oppure le misure di altezza, larghezza e profondità. Utili al fine di considerare la possibilità di volare in mezzo ai filari di nocciolo per effettuare fotografie o video ai lati delle piante, e in modo più generico per la gestione del trasporto.

Peso: peso del drone comprendente il veicolo, le eliche e il payload standard fornito dalla casa produttrice. Solitamente non comprende il peso di eventuali accessori montabili in modo indipendente.

Peso massimo al decollo: massimo peso trasportabile dal drone. Solitamente questo dato è disponibile nel caso di droni con possibilità di montare accessori ulteriori a quelli normalmente in dotazione.

Possibilità di montare telecamere aggiuntive: predisposizione del drone ad avere un meccanismo per poter smontare la gimbal o le telecamere al fine di poterne utilizzare di diverse tipologie, come ad esempio una telecamera a infrarossi. È necessario considerare sia gli eventuali inserti esistenti, sia la capacità di carico del drone.

Caratteristiche delle videocamere integrate: descrizione dei diversi sensori disponibili nelle videocamere integrate, comprendente il valore di qualità dell'immagine, il campo spettrale in cui esse vengono scattate e le modalità fotografiche disponibili. Questo criterio permette di dedurre la possibilità di effettuare fotografie a una altezza molto elevata, ma con una risoluzione adatta a eventuali ritagli dell'immagine scattata.

Rotazione delle videocamere integrate: capacità delle videocamere integrate di ruotare lungo i tre possibili assi: misura dell'angolo verso sinistra o verso destra rispetto al senso di marcia, misura dell'angolo verso l'alto o verso il basso rispetto al senso di marcia, rotazione sul proprio asse centrale (rotazione dell'inquadratura).

Licenza di guida necessaria: necessità di possedere la licenza di guida italiana per l'uso di droni. È determinata principalmente dal peso del drone per l'attuale governance (peso superiore ai 249 g).

Autonomia di volo: tempo di durata della batteria installata sul drone.

Costo: costo del pacchetto più minimale, comprendente al minimo il drone, disponibile sul portale del produttore oppure su portali esterni (ad esempio Amazon).

2.2.2 Tabella comparativa

La valutazione dei droni è posta in modalità tabellare, al fine di avere una comparazione migliore fra i modelli, in particolar modo riguardo il confronto di un singolo criterio (una colonna) fra tutti i modelli di droni differenti. La tabella è visualizzabile nella sezione 2.4, appartenente a questo capitolo.

2.2.3 Commenti

In questa sottosezione vengono aggiunti alcuni commenti descrittivi riguardo ad alcuni modelli analizzati. Fra questi vengono indicate le caratteristiche dei payload aggiuntivi per i modelli che dispongono di un'ampia gamma di dispositivi a essi compatibili.

DJI Mavic 2

Esistono due versioni di questo modello di drone: Pro e Zoom. L'unica differenza fra le due versioni riguarda la fotocamera, che ha caratteristiche differenti. La fotocamera della versione Zoom permette di avere uno zoom integrato e una serie di altre caratteristiche che lo predispongono come adatto a riprese cinematografiche. Queste caratteristiche esulano dallo scopo dell'analisi effettuata in questo capitolo, per cui non verranno approfondite ulteriormente.

PowerVision PowerEgg X

Per questo drone esiste una configurazione particolare di accessori che permette di avere una protezione dalla pioggia e per consentire l'atterraggio su terreni bagnati o direttamente sull'acqua: la modalità Wizard. Il drone, quindi, può essere comprato con questa configurazione di accessori, al costo di 1039€ sul PowerVision store o di 1145,07€ su Amazon.

DJI Inspire 2

Questo drone ha la possibilità di montare tre differenti modelli di fotocamere con gimbal: DJI Zenmuse X7, X5S e X4S. L'attacco compatibile è di tipologia a vite.

DJI Zenmuse X7 Fotocamera indirizzata alle produzioni cinematografiche. È dotata di DL-mount, un innesto integrato per ottiche aeree che permette d'intercambiare rapidamente le ottiche. Gli obiettivi supportati sono quattro, di marchio DJI, che si distinguono per la lunghezza focale (16, 24, 35, 50 mm).

Risoluzione: 24 MP.

Modalità fotografiche: scatto singolo, scatto a raffica (3, 5, 7, 10 scatti), AEB, a intervallo di tempo. Con l'utilizzo del supporto DJI CINESSD al posto di una scheda microSD, è possibile ampliare lo scatto a raffica finché il dispositivo di memoria lo permette.

Peso senza obiettivo: 449 g.

Costo: 2999€ (DJI store, senza obiettivo), 2999€ (Amazon, senza obiettivo).

DJI Zenmuse X5S Fotocamera indirizzata alle produzioni cinematografiche. Supporta diverse tipologie di obiettivi, disponibili fra tre marchi differenti: DJI, Panasonic e Olympus. Si distinguono principalmente per lunghezza focale e per dimensione del diaframma.

Risoluzione: 20,8 MP.

Modalità fotografiche: scatto singolo, scatto a raffica (3, 5, 7, 10, 14 scatti), AEB, a intervallo di tempo. Con l'utilizzo del supporto DJI CINESSD al posto di una scheda microSD, è possibile ampliare lo scatto a raffica finché il dispositivo di memoria lo permette.

Costo: 1599€ (DJI store, senza obiettivo), 2199€ (Amazon, con un obiettivo DJI M4/3, L9-45mm, F1.7).

DJI Zenmuse X4S Fotocamera con un solo obiettivo compatibile: DJI F/2.8-11, 8.8mm (35mm, equivalente 24mm).

Risoluzione: 20 MP.

Modalità fotografiche: scatto singolo, scatto a raffica (3, 5, 7, 10, 14 scatti), AEB, a intervallo di tempo.

Costo: 699€ (DJI Retail Store).

Italdron EVO4HSE RTK

Per questo drone è disponibile solamente il dato di peso comprensivo degli accessori a disposizione al momento della vendita: 2500g. Ha la possibilità di cambiare la videocamera all'interno del gimbal, oltre che la possibilità di utilizzare delle flange a innesto rapido per poter agganciare sensori aggiuntivi alla gimbal, senza la necessità di rimuovere la videocamera principale.

DJI P4 Multispectral

Il drone possiede un otturatore globale che scatta fotografie in contemporanea utilizzando tutti e sei i sensori. Perciò a ogni scatto si avranno sei immagini in output, di cui una RGB in formato JPEG e cinque nella banda specifica del sensore in formato TIFF. Cambiando la modalità di visualizzazione, è possibile cambiare la mappa di colori dell'immagine RGB in un'altra tra NDVI, GNDI o NDRE, mappe di colori derivanti dalla definizione di indici vegetazionali.

DJI M210 V2

Questo drone è compreso in una serie di droni denominata Serie Matrice 200 V2, che include, oltre a questo modello, anche i modelli M200 V2 e M210 RTK V2. Questi modelli si distinguono solamente per poche caratteristiche:

- il modello M200 non possiede la possibilità di avere due dispositivi di bordo in contemporanea verso il basso e nessun dispositivo di bordo verso l'alto, caratteristica posseduta invece dai modelli M210 e M210 RTK;
- il modello M210 RTK si distingue dagli altri per avere la compatibilità dell'utilizzo di dispositivi RTK.

I dispositivi RTK sono delle stazioni di terra mobili che supportano i voli a livello di navigazione satellitare, di trasmissione dati e di coordinazione di missioni con più droni, con il fine principale di correggere la posizione satellitare del drone in tempo reale.

Tutti i droni della serie supportano differenti videocamere:

- del brand DJI Zenmuse X4S, X5S, X7, XT, XT2, XTS e Z30;
- del brand SlantRange 4P e 4P+;
- del brand Sentera AGX710.

I dispositivi appartenenti a produttori terzi possono interfacciarsi con il drone in due modi: attraverso il DJI SkyPort V2, adattatore standard fra l'insero del drone e il dispositivo terzo, oppure attraverso il DJI X-Port, una gimbal stabilizzata su tre assi che possiede l'adattatore SkyPort V2 per inserire il dispositivo terzo e comunicare con esso.

I primi tre modelli di videocamere compatibili sono già stati analizzati in precedenza, nella parte riguardante il modello DJI Inspire 2 nella sottosezione [2.2.3](#) (X4S, X5S, X7).

DJI Zenmuse XT Fotocamera termica ad alta sensibilità, ideale per la telemetria e le analisi in ambiti tecnici, come quello agricolo.

Disponibile con due tipologie di telecamere diverse, che si distinguono per il loro valore di formato dei video digitali (640×512 , 336×256). Ognuna delle tipologie possiede quattro differenti obiettivi (rispettivamente 7.5mm, 9mm, 13mm, 19mm per la prima; 6.8mm, 9mm, 13mm, 19mm per la seconda).

Peso: 270g.

Rotazione gimbal: verticale $[+35^\circ, -90^\circ]$; orizzontale $[+320^\circ; -320^\circ]$; rotazionale sull'asse $[-15^\circ; +15^\circ]$.

Sensibilità: <50mK a F1.0.

Costo: Non disponibile al pubblico.

DJI Zenmuse XT2 Fotocamera che possiede un doppio sensore: uno termico, molto simile a quello sopra descritto del modello DJI Zenmuse XT, e uno ottico. Ne esistono due versioni che differiscono, come per la versione XT, solamente per il formato dei video digitali del termosensore (640×512 , 336×256). Ognuna delle versioni possiede tre differenti ottiche (rispettivamente 13mm, 19mm, 25mm per la prima; 9mm, 13mm, 19mm per la seconda).

Peso: 629 g con obiettivo da 25mm; 588g con gli altri obiettivi.

Rotazione gimbal: verticale $[+30^\circ, -90^\circ]$; orizzontale $[+320^\circ; -320^\circ]$.

Sensibilità (sensore termico): <50mK a F1.0.

Risoluzione (sensore ottico): 12 MP.

Modalità fotografiche (sensore ottico): scatto singolo, scatto a raffica (3, 5 scatti), a intervallo di tempo.

Costo: Non disponibile al pubblico.

DJI Zenmuse Z30 Fotocamera con un potente zoom (ottico 30x e digitale 6x) che permette di raggiungere una totalità di 180x. Indirizzata a ispezioni aree su larga scala con il fine di non arrecare danno né al drone né ai piloti, raggiungendo comunque lo scopo di ottenere immagini ad alto livello di dettaglio degli obiettivi dell'ispezione. Un esempio è l'ispezione di pale eoliche.

Peso: 556 g.

Rotazione gimbal: verticale [+30°,-120°]; orizzontale [+320°;-320°].

Modalità fotografiche: a intervallo di tempo.

Costo: Non disponibile al pubblico.

SlantRange 4P e 4P+ Serie di sensori multispettrali indirizzati all'estrazione di dati in campo agricolo, come il numero di piante presenti in una certa ripresa oppure la loro grandezza o la loro forma. Oltre ai sensori multispettrali è presente anche un sensore ottico che permette di scattare immagini ad alta qualità nello spettro visibile.

I dispositivi incorporano anche un misuratore della luce solare, al fine di mantenere un costante livello di luce nelle immagini scattate in un singolo volo. L'otturatore per lo scatto delle immagini è globale per tutti i sensori.

La differenza fra i due modelli è solamente una: il 4P+ ha una risoluzione doppia rispetto al 4P.

Peso: 350 g.

Sensori: sei con diverse bande (posizionamento a 470, 520, 620, 670, 720, 850 nm).

Costo: Non disponibile al pubblico.

Sentera AGX710 Fotocamera indirizzata all'industria agricola, al fine di fornire riprese e dati precisi per l'analisi della salute delle proprie coltivazioni.

È disponibile in diverse configurazioni di sensori al fine di avere immagini sia RGB che multispettrali (tramite formati che evidenziano indici vegetazionali oppure nelle diverse bande spettrali).

Peso: 270 g.

Rotazione gimbal: verticale [0°,-90°].

Configurazioni disponibili:

1. RGB + NDVI (Rosso 625nm, NIR 850 nm)
2. RGB + NDRE (Red-Edge 720 nm, NIR 840nm)
3. NDVI + NDRE
4. Multispettrale (Blu 446nm, Verde 548nm, Rosso 650nm, Red-Edge 720 nm, NIR 840nm).

Costo: 4299\$ (Sentera store).

DJI Matrice 300 RTK

Questo modello di drone è compatibile con l'utilizzo di dispositivi RTK, cioè stazioni di terra mobili che supportano i voli a livello di navigazione satellitare, di trasmissione dati e di coordinazione di missioni con più droni, con il fine principale di correggere la posizione satellitare del drone in tempo reale.

Questo drone supporta differenti videocamere:

- del brand DJI Zenmuse H20, H20T, XT2, XTS e Z30;
- del brand SlantRange 4P e 4P+;
- del brand Sentera AGX710.

Come per il modello DJI M210 V2, i dispositivi appartenenti a produttori terzi possono interfacciarsi con il drone in due modi: attraverso il DJI SkyPort V2, adattatore standard fra l'inserto del drone e il dispositivo terzo, oppure attraverso il DJI X-Port, gimbal stabilizzata su tre assi che possiede l'adattatore SkyPort V2 per inserire e comunicare con il dispositivo terzo.

Nello stesso momento è possibile montare da uno a tre carichi, siccome questo drone possiede tre inserti: uno superiore e due inferiori. Le combinazioni possibili fra i carichi, però, sono limitate, come presentato nella tabella 2.1. In generale può essere montato solo un carico di terze parti alla volta, insieme a carichi della DJI, mentre il carico XT2 può essere montato solo nel primo inserto verso il basso.

Diversi modelli di videocamere compatibili sono già stati analizzati in precedenza, nella sottosezione 2.2.3 nella parte riguardante il modello DJI M210 V2 (DJI Zenmuse XT2, XTS e Z30; SlantRange 4P e 4P+; Sentera AGX710).

DJI Zenmuse H20 e H20T Carichi caratterizzati dalla loro adattabilità grazie ai tre sensori a loro disposizione: un sensore ottico con zoom, un sensore ottico grandangolare e un sensore LRF (laser rangefinder), che permette di effettuare misure fino a 1200m.

Il modello H20T possiede anche un sensore termico.

Peso: H20 - 678 g; H20T - 828 g.

Rotazione gimbal: verticale [+30°,-120°]; orizzontale [+320°;-320°].

Sensibilità (sensore termico, solo H20T): <50mK a F1.0.

Risoluzione (sensore ottico zoom): 20 MP, con zoom fino a 200x.

Risoluzione (sensore ottico grandangolare): 12 MP.

Risoluzione (sensore termico): 640 × 512, con zoom fino a x8.

Intervallo di misurazione (sensore telemetro laser): da 3m a 1200m.

Costo: Non disponibile al pubblico.

<i>Combinazione dei carichi</i>		<i>Tipologia carico compatibile</i>
Carico Singolo	Verso l'alto	H20, H20T, XTS, Z30, carichi terzi
	Verso il basso	H20, H20T, XTS, XT2, Z30, carichi di terze parti
Carico doppio	Verso il basso entrambi	XT2+Z30; XT2+H20; XTS+Z30;
	Uno verso l'alto, uno verso il basso	XTS+H20; tipologia carichi appena citati + carico di terze parti
Carico triplo	Uno verso l'alto, due verso il basso	Una fra combinazioni (XT2+Z30, XT2+H20, XTS+Z30, XTS+H20) + carico di terze parti

Tabella 2.1. Combinazioni possibili di payload per il modello DJI Matrice 300 RTK.

2.2.4 Conclusioni

Tutti i droni analizzati nella tabella hanno specifici pregi e difetti, che in una visione più generale possono essere espressi dal concetto di **costo**, solitamente più alto per i droni con caratteristiche migliori.

I migliori droni, considerando lo scopo della valutazione e i criteri utilizzati, sono i modelli **DJI P4 Multispectral**, **Italdron EVO4HSE RTK** e **DJI Matrice 300**. Il primo di questi risulta leggermente più compatto rispetto agli altri due, sia in termini di dimensioni che di peso, ma non permette di poter cambiare i sensori fotografici. Tutti e tre i modelli possono essere accessoriati con sensori multispettrali. Solamente il modello Italdron EVO4HSE RTK e DJI Matrice 300 possono montare anche sensori di tipo termico. I difetti comuni di questi tre modelli sono la dimensione consistente, che potrebbe dare problemi per riprese all'interno dei filari di piante di nocciolo, e la necessità di licenza di guida.

Per questo motivo si potrebbe considerare l'utilizzo di un modello di dimensioni ridotte per lo scopo fotografico all'interno dei filari, mentre per le fasi iniziali di prove sul campo si potrebbe valutare l'utilizzo di un modello che non necessiti di licenza di guida per il pilota. Il **DJI Mavic Mini** unirebbe entrambe queste necessità.

Il modello **DJI Inspire 2** potrebbe essere considerato valido solamente dal punto di vista della flessibilità di scelta riguardo la qualità delle immagini RGB, in quanto supporta tre differenti modelli di gimbal, con differente varietà di obiettivi fotografici. Fra questi dispositivi, però, nessuno è in grado di scattare foto nella banda infrarossa, il che esclude questo modello fra quelli validi per lo scopo dell'analisi.

Fra i modelli della serie **DJI M200 V2**, è stato elevato il modello **M210** solamente per avere a disposizione la possibilità di montare due payload contemporaneamente, e perché non è necessario al momento avere la compatibilità con dispositivi di terra RTK, ma è stato escluso dalla scelta finale perché a breve verrà interrotta la sua produzione.

In conclusione i modelli che permettono di avere più flessibilità di scelta a livello sensoristico sono Italdron EVO4HSE RTK e DJI Matrice 300, che mettono a disposizione un'ampia gamma di payload compatibili, ma allo stesso tempo il costo di acquisto diventa più elevato sia in termini del drone, sia in termini di dispositivi compatibili a esso. Per questi motivi il modello DJI P4 Multispectral potrebbe essere un acquisto valido: un modello compatto con tutte le caratteristiche necessarie allo scopo, senza la necessità di avere costi ulteriori e scelte complicate per trovare i payload migliori allo scopo. Ovviamente potrebbe essere limitante nel caso in cui le richieste a livello di qualità delle immagini ottiche e multispettrali diventassero più stringenti in futuro.

Per poter confrontare meglio i payload disponibili per i modelli di droni DJI P4 Multispectral, Italdron EVO4HSE RTK e DJI Matrice 300, è disponibile una tabella di confronto nella sezione [2.5](#) con i dati più rilevanti.

2.2.5 Scelta d'acquisto

Fra Marzo e Aprile 2021, in seguito alla ricerca di mercato e alle valutazioni sopra descritte, si è deciso e si è provveduto all'acquisto di un drone del modello DJI P4 Multispectral. In questo modo si è potuto unire la compattezza del drone, con la disponibilità di utilizzare sia sensori nello spettro del visibile sia multispettrali, considerando non necessaria la possibilità di cambiare il payload per gli scopi del caso di studio.

2.3 Valutazione di applicazioni per la definizione di piani di volo

2.3.1 Volo automatico e autonomo

Nell'uso comune questi due aggettivi, in riferimento al termine volo, vengono utilizzati impropriamente come sinonimi. Ma nell'ambito del pilotaggio di droni i due termini assumono significati

diversi:

Volo automatico: il drone possiede, attraverso funzionalità intrinseche o programmi esterni, la capacità di volo senza ausilio di un pilota remoto, ma richiede input umani per la partenza, per la pianificazione del volo e per la supervisione dello stesso. Inoltre durante tutta la tratta di volo automatico, un essere umano deve sempre osservare il drone ed essere pronto a intervenire in caso di problematiche, con l'eventualità di disabilitarlo completamente, con la seguente caduta, in caso di non risposta ai comandi umani.

Volo autonomo: il drone può svolgere in assoluta autonomia tutte le fasi di volo, senza nessun intervento umano.

A livello legislativo, in Italia, il volo automatico è permesso, mentre il volo autonomo è vietato.

La modalità di volo utilizzata più comunemente, che avviene attraverso il completo controllo del drone da parte del pilota, viene definita **manuale**.

2.3.2 Criteri di valutazione delle applicazioni

1. Funzionalità offerte e tipologia di piattaforma d'accesso a esse.
2. Tipologia e costi dei piani di abbonamento o delle licenze software.
3. Modelli di droni compatibili.

2.3.3 Applicazioni per la pianificazione di voli automatici

Dronelink

Dronelink è una piattaforma composta da due parti: una tramite web browser e una parte tramite applicazione nativa (Android e Ios). Quella tramite web browser permette la definizione dei propri piani di volo, comprendenti waypoint, aree di mappatura sia dall'alto che 3D, specificando l'orientamento della videocamera a ogni tratto. Esistono anche funzionalità più "ludiche", come ad esempio la fotografia a 360° intorno a un punto selezionato oppure l'hyperlapse, definizione di uno scatto complesso che non definibile da altri componenti base. Quella tramite piattaforma nativa, invece, permette di connettersi con il drone e di eseguire i piani di volo precedentemente definiti, avendo la possibilità di controllare totalmente il drone durante il suo volo automatico.

Le varie tipologie di abbonamento si dividono principalmente in due categorie: gli abbonamenti per hobbisti e gli abbonamenti per professionisti, cioè per coloro che utilizzano le funzionalità a scopi commerciali. Nel primo caso gli abbonamenti sono pagati una tantum, mentre nel secondo sono pagati mensilmente e con un limite di utilizzo in minuti. All'interno di ogni categoria i livelli di abbonamento presenti sono poi tre, con funzionalità disponibili crescenti e, di conseguenza, prezzo crescente. Un riassunto viene presentato utilizzando la tabella 2.2. Maggiori dettagli sono disponibili presso il sito descrittivo dell'applicazione¹.

I modelli di droni compatibili comprendono un'ampia gamma di droni DJI e, ancora in fase sperimentale, il supporto al Parrot Anafi (lista completa²).

¹<https://app.dronelink.com/pricing>

²<https://support.dronelink.com/hc/en-us/articles/360025829933-What-are-the-system-requirements>

Hobbyist (una tantum)	Basic	Premium	Elite
	20\$	40\$	80\$
Business (al mese)	Starter	Growth	Enterprise
	20\$	50\$	Non pubblico

Tabella 2.2. Dronelink: riassunto degli abbonamenti disponibili.

UGCS

UGCS è una piattaforma composta da due componenti: un software che permette di avere una ground station, compatibile con i sistemi operativi Mac, Windows e Ubuntu, e un'applicazione mobile, compatibile con Android ed Ios, che si collega con il drone durante il suo funzionamento. Il software di ground station permette innanzitutto di controllare nella totalità il drone utilizzato, con la possibilità di avere dati telemetrici in tempo reale e la visione in 3D delle No-Fly-Zones. Inoltre permette la creazione di piani di volo automatici nelle più svariate modalità, a partire da quella punto a punto fino a quelle di mappatura del terreno oppure delle facciate di edifici. In particolare, per quanto riguarda i piani di volo per la mappatura, è possibile impostare anche mappature ad altezza costante, inserendo all'interno del software opportune mappe di altitudine della zona geografica in cui verrà eseguito il volo. Il software supporta anche l'importazione e l'esportazione dei dati riguardanti i piani di volo creati, in diversi formati.

Gli abbonamenti si dividono in due macro-categorie: abbonamento Pro e abbonamento Enterprise. La differenza riguarda principalmente il video-streaming in tempo reale verso piattaforme esterne e la possibilità di avere gli SDK per creare applicazioni che integrano UGCS al proprio interno. Per maggiori informazioni, si può avere la comparazione completa presso il sito descrittivo dell'applicazione³. Le macro-categorie si suddividono a loro volta in categorie basate sulla validità temporale: mensile oppure permanente. Tutti gli abbonamenti sono da intendersi per singolo utilizzatore. Maggiori dettagli sono disponibili all'interno di una delle pagine del sito sopraccitato⁴. Esiste comunque una versione gratuita per l'utilizzo della piattaforma browser per la pianificazione di voli.

La compatibilità rispetto ai modelli dei droni è molto ampia: comprende un'ampia gamma di droni DJI, e alcuni modelli dei marchi MavLink, Parrot, Mikrokopter, Micropolito e Lockheed Martin. La lista completa e dettagliata è accessibile presso una delle pagine del sito sopraccitato⁵.

	Subscription	Perpatual
Pro	39\$	600\$
Enterprise	99\$	2000\$

Tabella 2.3. UGCS: riassunto degli abbonamenti disponibili.

Litchi

Litchi è una piattaforma composta da due elementi principali: un'applicazione mobile, per iOS e Android, e un Mission Hub disponibile via browser. L'applicazione mobile permette l'esecuzione di piani di volo automaticamente, controllando direttamente il drone. I piani di volo devono prima essere definiti attraverso il Mission Hub e caricati online sul cloud, al fine poi di poterli utilizzare attraverso l'applicazione. Essi devono essere definiti utilizzando dei waypoint, con la possibilità d'impostare l'inclinazione della videocamera, l'altezza del drone, e altri parametri specifici di volo.

³<https://www.ugcs.com/files/manuals/ugcs-license-feature-comparison-200320.pdf>

⁴<https://www.ugcs.com/pricing>

⁵https://www.ugcs.com/supported_drones_autopilots

È possibile importare piani di volo ed esportarli all'interno del Mission Hub, eventualmente anche definiti con altri software, attraverso l'uso di formati standard.

L'utilizzo del Mission Hub è gratuito, richiede solo la creazione di credenziali d'accesso per poter salvare i piani di volo definiti. L'applicazione, invece, è a pagamento una tantum: 25€ sia per la versione iOS che per quella Android.

Esistono anche altre applicazioni di supporto della medesima software house, che permettono di creare dei ponti radio tra dispositivi per il controllo del medesimo drone, oppure di effettuare streaming in tempo reale delle immagini catturate dal drone.

Litchi è compatibile solamente con droni DJI, dei seguenti modelli: Mavic Mini 1, Mavic 2 (Zoom e Pro), Mavic (Air e Pro), Phantom 4 (Standard, Advanced, Pro e ProV2), Phantom 3 (Standard, 4K, Advanced e Professional), Inspire 1 (X3, Z3, Pro e RAW), Inspire 2 and Spark.

Airmap

Airmap è un'applicazione mobile, per iOS e Android, che permette la visualizzazione di aree a volo limitato e la definizione di piani di volo, tramite waypoint, al fine di poter vederne eventualmente la fattibilità atmosferica e legislativa. Solamente negli USA permette anche di avere la convalidazione del piano di volo da parte dell'ente preposto. Durante la fase di volo è possibile controllare eventuale traffico aereo nelle vicinanze attraverso degli avvisi visivi, per portare l'attenzione al pilota per un controllo maggiormente accurato del mezzo.

Esiste anche la possibilità di fare pianificazioni attraverso una piattaforma browser, per poi averle salvate in cloud e poterle utilizzare nell'applicazione, attraverso le medesime credenziali di login.

L'utilizzo dell'applicazione e della piattaforma browser è gratuito. Per gestire flussi più complessi di approvazione dei voli, è necessario un abbonamento, che però esula dagli scopi dell'analisi. Per la versione gratuita non sono esplicitati eventuali limiti in termini di numero di piani voli caricabili sul cloud o in termini di utilizzo di altre tipologie di risorse.

La compatibilità rispetto ai droni comprende solamente droni del marchio DJI: Matrice, Inspire 1 e 2, Mavic Air, Mavic Air 2 (Pro e Zoom), Phantom 4, Phantom 4 Pro, Phantom 3 Series e Spark.

DJI Terra

DJI Terra è un'applicazione desktop, per sistema operativo Windows, sviluppata dall'azienda produttrice di droni DJI, al fine di offrire una piattaforma comoda e intuitiva da utilizzare per la guida e l'analisi d'immagini scattate dai modelli più tecnologici del proprio marchio. Le funzionalità principali sono raggruppabili in tre categorie principali:

Definizione di piani d'ispezione attraverso la definizione e l'esecuzione di voli automatici in diverse modalità: copertura di un'area definita, volo a waypoint (anche su modelli 3D), volo in linea retta e volo obliquo, per estrarre grandi quantità di dati utili alla mappatura 3D;

Mappatura 2D e 3D permettendo la generazione di ortomosaici ad altissima qualità alla conclusione della ripresa, ortomosaici a qualità medio-alta in tempo reale durante la ripresa, mappe multispettrali 2D e modelli 3D dell'area ripresa;

Analisi dei dati con la possibilità di effettuare misurazioni di lunghezza o di ampiezza sui modelli 2D e 3D realizzati, e di etichettare le misure o i punti critici all'interno dei modelli.

Le licenze sono tutte a pagamento e della durata di un anno, tranne nel caso della licenza Cluster che è valida in modo permanente. In ordine crescente di autorizzazioni concesse, le licenze sono:

Agricoltura che consente la definizione, l'esecuzione di piani di volo e la mappatura 2D anche multispettrale;

Pro che aggiunge tutte le altre funzionalità riguardanti la parte 3D;

Elettrica che comprende anche un supporto per ispezioni dettagliate in campo elettrico;

Cluster che permette di elaborare i dati contemporaneamente su più dispositivi.

La licenza Pro può essere utilizzata sia online che offline. Quest'ultima viene rilasciata con lo scopo di essere utilizzata da utenti con requisiti elevati in termini di sicurezza delle informazioni, per cui non è necessario un collegamento a internet per verificare le autorizzazioni possedute. La licenza Cluster, invece, è disponibile solo offline.

Il software è compatibile con i prodotti DJI della serie Phantom 4 per la definizione e l'esecuzione dei piani di volo automatici e il processamento e l'analisi dei dati, mentre è compatibile con i payload Zenmuse P1, L1 e X7 solo per l'elaborazione dei dati.

DJI GS Pro

DJI GroundStation Pro è un'applicazione mobile, disponibile solamente su sistema operativo iOS, indirizzata a facilitare l'esecuzione automatica e la gestione di piani di volo per i possessori di droni appartenenti al marchio DJI, casa produttrice sia del software che dell'hardware compatibile con l'applicazione.

Le funzionalità dell'applicazione possono essere riassunte in diverse categorie:

- **Definizione di piani di volo automatici** attraverso la selezione di un'area 3D da coprire durante la ripresa, oppure attraverso un percorso di waypoint configurando per ognuno l'altezza e l'inquadratura della videocamera;
- **Esecuzione di piani di volo automatici** permettendo il controllo totale del drone e il salvataggio sia delle immagini, sia delle informazioni riguardanti il volo (log);
- **Definizione dei limiti di sorvolo** grazie alla configurazione di perimetri virtuali utili nel caso di aree con limiti legislativi particolari, cioè a volo limitato;
- **Gestione dei team** attraverso la definizione di progetti e di ruoli, che permette di avere differenti permessi di visibilità: Member, che può lavorare solo a livello di progetto, Administrator, che può anche modificare i team e Agent, che può anche modificare i permessi e gestire le licenze.

Sono inoltre disponibili alcune funzionalità particolari acquistabili direttamente all'interno dell'applicazione:

- **3D Map POI** che permette di gestire riprese legate alla mappatura 3D di oggetti;
- **Importazione di file KML/SHP** che permette d'importare i file in questi particolari formati al fine di definire un piano di volo utilizzabile dalla piattaforma;
- **PhotoMap** utile alla ricomposizione delle foto di ripresa al fine di creare un mosaico completo dell'area.

Tutte le informazioni legate a un account vengono salvate in cloud e possono essere disponibili su differenti dispositivi.

Le tipologie di licenze disponibili sono tre: una gratuita, che permette di caricare in cloud un numero limitato di missioni, una a pagamento, che elimina il limite numerico, e una sempre a pagamento che permette di ottenere l'accesso alle funzionalità di gestione dei team.

DJI GS Pro è compatibile con i modelli di droni DJI A3, N3, della serie Mavic, della serie Phantom, della serie Matrice, della serie Inspire e dei modelli di fotocamera Zenmuse della serie X e della serie Z (la lista dettagliata è disponibile presso il sito di presentazione della piattaforma).

Conclusioni

Tutte le applicazioni hanno le caratteristiche più importanti ricercate da questa valutazione: la creazione di piani di volo e l'esecuzione di essi in modo automatico. Fra queste, però, solamente UGCS è compatibile con droni di marchi differenti da DJI, ma non è compatibile con droni del marchio Italdron.

Per quanto riguarda la definizione di piani di volo, UGCS, Dronelink, DJI Terra e DJI GS Pro offrono anche modalità più legate alla topografia, permettendo mappature aree (DJI Terra e DJI GS Pro anche in 3D).

DJI Terra e DJI GS Pro offrono una buona quantità di funzionalità già a partire dalla licenza più economica, con la limitazione di essere compatibile solo con un numero limitato di modelli, tutti appartenenti al marchio DJI. Il vantaggio in ciò è che i produttori del software e dell'hardware sono i medesimi, per cui i problemi di compatibilità sono ridotti al minimo.

È possibile tenere in considerazione un approccio ibrido, utilizzando quindi UGCS per la pianificazione dei piani di volo, esportandoli e importandoli poi successivamente nelle piattaforme di pianificazione di Litchi o Dronelink, al fine di utilizzare le loro applicazioni di controllo ed esecuzione automatica dei voli. Questa possibilità potrebbe essere valutata nel caso di problemi di accessibilità dell'applicazione mobile di UGCS, oppure per questioni di costo, visto che Litchi e Dronelink sono a pagamento una tantum.

In conclusione le piattaforme migliori per gli scopi sopracitati sono **UGCS, DJI Terra e DJI GS Pro**, privilegiando quest'ultime due nel caso di acquisto e utilizzo di un drone DJI.

2.3.4 Scelta finale

Fra Marzo e Aprile 2021, insieme all'acquisto del drone, si è scelto di utilizzare l'applicazione DJI GS Pro, considerando che possiede le funzionalità di creazione e di gestione dei voli automatici utili allo scopo del lavoro di tesi e che è ottimizzata per il modello del drone scelto.

2.4 Tabella comparativa droni

<i>Nome e marchio</i>	<i>Criterio 1</i>	<i>Criterio 2</i>	<i>Criterio 3</i>	<i>Criterio 4</i>	<i>Criterio 5</i>	<i>Criterio 6</i>	<i>Criterio 7</i>	<i>Criterio 8</i>	<i>Criterio 9</i>
DJI Mavic Air 2	183 × 253 × 77 mm (Aperto) 180 × 97 × 74 mm (Chiuso)	570 g	Dato mancante	No	Risoluzione: 12 MP Modalità fotografiche: singolo, raffica, panorama, a intervallo di tempo, AEB.	Orizzontale: [-80°; 80°] Verticale: [-90°; 24°]	Sì	34 minuti (senza vento)	849 € (DJI store) 819 € (Amazon)
DJI Mavic 2	322 × 242 × 84 mm (Aperto) 214 × 91 × 84 mm (Chiuso)	Pro: 907 g Zoom: 905 g	Dato mancante	No	Risoluzione: Pro - 20 MP Zoom - 12 MP Modalità fotografiche: singolo, raffica, panorama, a intervallo di tempo, AEB	Orizzontale: [-75°; 75°] Verticale: [-90°; 30°]	Sì	31 minuti (senza vento)	Pro: • 1499 € (DJI store); • 1461,80 € (Amazon). Zoom: • 1249 € (DJI store); • 1225,94 € (Amazon).
DJI Phantom 4 Pro V2.0	350 mm (Diagonale)	1375 g	Dato mancante	No	Risoluzione: 20 MP Modalità fotografiche: singolo, raffica, a intervallo di tempo, AEB.	Verticale: [-90°; 30°]	Sì	30 minuti (senza vento)	1699 € (DJI store)

<i>Nome e marchio</i>	<i>Criterio 1</i>	<i>Criterio 2</i>	<i>Criterio 3</i>	<i>Criterio 4</i>	<i>Criterio 5</i>	<i>Criterio 6</i>	<i>Criterio 7</i>	<i>Criterio 8</i>	<i>Criterio 9</i>
PowerVision PowerEgg X	427,5 mm (Diagonale)	862 g	Dato mancante	No	Risoluzione: 20 MP Modalità fotografiche: singolo, raffica, a intervallo di tempo, AEB.	Verticale: [-90°; 20°]	Sì	30 minuti	769 € (PowerVision store) 839,95 € (Amazon)
DJI Inspire 2	427 × 425 × 317 mm (Aperto)	3440 g (senza gimbal e fotocamera)	4250 g	Sì	Videocamere integrate non presenti. Modelli di telecamere compatibili: DJI Zenmuse X7, X5S e X4S.	Videocamere integrate non presenti.	Sì	tra 20 e 25 minuti (dipende dal peso al decollo)	3399 € (DJI store, incluse due telecamere con gimbal)
DJI M210 V2	883 × 886 × 398 mm (Aperto) 722 × 282 × 242 mm (Chiuso)	4800 g	6140 g	Sì	Videocamere integrate non presenti. Modelli di telecamere compatibili: DJI Zenmuse X4S, X5S, X7, XT, XT2, XTS, Z30; SlantRange 4P e 4P+; Sentera AGX710.	Videocamere integrate non presenti.	Sì	Tra 20 e 34 minuti (nessun carico; carico massimo)	Non pubblico: è necessario contattare DJI.

<i>Nome e marchio</i>	<i>Criterio 1</i>	<i>Criterio 2</i>	<i>Criterio 3</i>	<i>Criterio 4</i>	<i>Criterio 5</i>	<i>Criterio 6</i>	<i>Criterio 7</i>	<i>Criterio 8</i>	<i>Criterio 9</i>
DJI P4 Multispectral	350 mm (Diagonale)	1487 g	Dato mancante	No	Sei sensori CMOS: uno per immagini RGB, cinque sensori multispettrali (dal blu all'infrarosso). Inoltre è presente un sensore che permette la compensazione delle immagini in base alla luce solare in tempo reale. Risoluzione di ogni sensore: 2,08 MP.	Verticale: [-90° ; 30°]	Sì	27 minuti	Non pubblico: è necessario contattare DJI.
DJI Matrice 300	810 × 670 × 430 mm (Aperto) 430 × 420 × 430 mm (Chiuso)	3600 g (senza batterie) 6300 g (con due batterie TB60)	9000 g	Sì	Videocamere integrate non presenti. Modelli di telecamere compatibili: DJI Zenmuse H20, H20T, XT2, XTS, Z30; SlantRange 4P e 4P+; Sentera AGX710.	Videocamere integrate non presenti.	Sì	55 minuti (senza carichi) 45 minuti (carico da 700 g)	Non pubblico: è necessario contattare DJI.

<i>Nome e marchio</i>	<i>Criterio 1</i>	<i>Criterio 2</i>	<i>Criterio 3</i>	<i>Criterio 4</i>	<i>Criterio 5</i>	<i>Criterio 6</i>	<i>Criterio 7</i>	<i>Criterio 8</i>	<i>Criterio 9</i>
Italdron EVO4HSE RTK	630 × 630 × 430 mm	Dato mancante	9000 g	Sì	Telecamera principale: Sony A7RII 42 MP, obiettivo Zeiss FE 35mm F2.8. Flangia a innesto rapido, provvista di tre sensori: <ul style="list-style-type: none"> • Termocamera Flir • Vue Pro con ottica da 13 mm; • Camera multispettrale con cinque bande separate: blu, verde, rosso, IR, red-edge; • Camera di riscontro da 16 MP. 	360° su tutti e tre gli assi: verticale, orizzontale e rotazionale.	Sì	tra 15 e 30 minuti	Non pubblico: è necessario contattare Italdron.
DJI Mavic Mini	159 × 202 × 55 mm (Aperto) 140 × 81 × 57 mm (Chiuso)	249 g	Dato mancante	No	Risoluzione: 12 MP Modalità fotografiche: singolo, a intervallo di tempo.	Verticale: [-90°; 20°]	No	30 minuti (senza vento)	399 € (DJI store) 349 € (Amazon)

<i>Nome e marchio</i>	<i>Criterio 1</i>	<i>Criterio 2</i>	<i>Criterio 3</i>	<i>Criterio 4</i>	<i>Criterio 5</i>	<i>Criterio 6</i>	<i>Criterio 7</i>	<i>Criterio 8</i>	<i>Criterio 9</i>
Parrot Anafi	175 × 240 × 65 mm (Aperto) 224 × 67 × 65 mm (Chiuso)	320 g	Dato mancante	No	Risoluzione: 16 MP	Verticale: [-90°; 90°]	Sì	25 minuti	699 € (PowerVision store) 699 € (Amazon)

2.5 Tabella comparativa payload

Nome e marchio	Drone compatibile	Rotazione della gimbal	Costo	Peso	Sensore termico			Sensore ottico		Sensore multispettrale
					Sensibilità	Risoluzione	Modalità fotografiche	Risoluzione	Modalità fotografiche	
/	DJI P4 Multispectral	Verticale: [-90°; 30°]	/	Incluso nel drone	Assente			2.08 MP	Scatto singolo	Cinque sensori dal blu all'infrarosso. (bande a 450, 560, 650, 730 840 nm).
DJI Zennuse XT2	DJI Matrice 300 RTK	Verticale: [-90°; 30°] Orizzontale: [-320°; 320°]	/	629 g (obiettivo da 25 mm) 588 g (con altri obiettivi)	<50mK a F1.0	336 × 256 oppure 640 × 512 pixel (dipende dal modello)	/	12 MP	Scatto singolo, scatto a raffica (3 o 5 scatti), a intervallo di tempo.	Assente
DJI Zennuse XTS	DJI Matrice 300 RTK	/	/	387 g	<40mK a F1.0	640 × 512 pixel	A intervallo di tempo	Assente	Assente	Assente
DJI Zennuse Z30	DJI Matrice 300 RTK	Verticale: [-120°; 30°] Orizzontale: [-320°; 320°]	/	556 g	Assente			/	A intervallo di tempo	Assente

Nome e marchio	Drone compatibile	Rotazione della gimbal	Costo	Peso	Sensore termico			Sensore ottico		Sensore multispettrale
					Sensibilità	Risoluzione	Modalità fotografiche	Risoluzione	Modalità fotografiche	
SlatRange 4P e 4P+	DJI Matrice 300 RTK	Dipende dalla gimbal utilizzata	/	350 g	Assente			/		Sei sensori, di cui uno ottico, a diverse bande di frequenza (470, 520, 620, 670, 720, 850 nm)
Sentera AGX710	DJI Matrice 300 RTK	Verticale: [-90°; 0°]	4299\$	270 g	Assente			/		Sei sensori, di cui uno ottico, a diverse bande di frequenza (446 nm Blu, 548 nm Verde, 650 nm Rosso, 720 nm Red-Edge, 840 nm NIR)
Sony A7RII	Italdron EVO4HSE RTK	360° su tutti gli assi	/	625 g	Può essere usata contemporaneamente alla flangia a innesto rapido			42 MP	Scatto singolo, scatto a raffica, a intervallo di tempo e altre (fotocamera completa e utilizzabile in autonomia)	Può essere usata contemporaneamente alla flangia a innesto rapido

Nome e marchio	Drone compatibile	Rotazione della gimbal	Costo	Peso	Sensore termico			Sensore ottico		Sensore multispettrale
					Sensibilità	Risoluzione	Modalità fotografiche	Risoluzione	Modalità fotografiche	
Flangia a innesto rapido	Italdron EVO4HSE RTK	360° su tutti gli assi	/	/	336 × 256 oppure 640 × 512 pixel (dipende dal modello)	/	/	16 MP	/	Cinque sensori, uno per ogni banda spettrale: blu, verde, rosso, red-edge, IR.

Capitolo 3

Collezione e gestione delle immagini

3.1 Riprese delle immagini

3.1.1 Sopralluoghi preliminari

Al fine di poter definire quali piante dovessero essere monitorate e fotografate e le modalità con cui venissero effettuate le riprese, nel periodo di Maggio 2021 sono stati eseguiti alcuni sopralluoghi preliminari con il supporto di alcuni esperti in botanica e con il pilota del drone, cioè colui in possesso della patente necessaria alla guida del drone acquistato DJI P4 Multispectral.

Attraverso questi incontri si sono definite quali piante dovessero essere monitorate sia dal punto di vista botanico, sia attraverso le riprese automatiche del drone. Si è deciso di monitorare quaranta differenti piante, di cui ventiquattro adiacenti in un'area ristretta, mentre le altre sedici sparse in altri filari del campo. Inoltre è stato definito che l'inquadratura delle immagini dovesse essere dall'alto, a un'altezza di poco superiore alla pianta, in modo che essa fosse ripresa completamente dall'immagine e centrata in essa, cercando di escludere al massimo possibile eventuali altre piante nell'inquadratura. In questo modo le immagini non hanno richiesto un successivo lavoro di rielaborazione prima della loro analisi.

3.1.2 Definizione dei piani di volo

Grazie ai vincoli definiti nei sopralluoghi preliminari, sono stati definiti i piani di volo automatici che permettessero la ripresa delle immagini sempre nella medesima modalità, per le piante che richiedevano il monitoraggio. Proprio per questa motivazione i piani di volo definiti sono stati utilizzati in tutte le riprese effettuate.

Per effettuare questa operazione è stato utilizzato il software DJI GS Pro.

I piani di volo sono stati configurati attraverso waypoint, cioè definendo un punto specifico per ogni pianta monitorata, indicandone l'altezza e l'inclinazione dell'inquadratura. L'altezza è stata definita all'incirca a 10 metri, al fine d'incontrare le richieste espresse nella sottosezione precedente 3.1.1. L'inclinazione dell'inquadratura, invece, è stata impostata a un'angolazione di -90° , considerando lo zero come la visuale verso la direzione di moto del drone, i valori positivi di angolo verso l'alto e quelli negativi verso il basso.

Nelle figure 3.1 e 3.2 si possono osservare due piani di volo definiti e utilizzati nelle riprese, con una parte parziale del campo visualizzata.

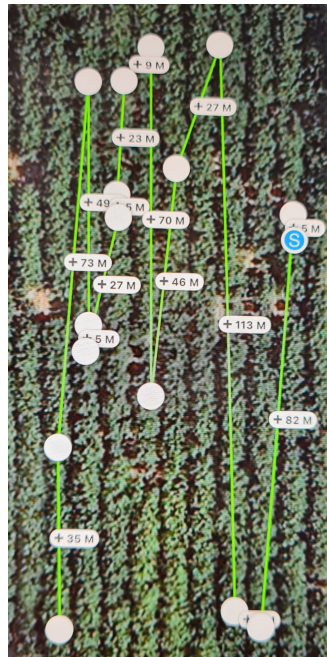


Figura 3.1. Piano di volo definito con l'applicazione DJI GS Pro - I.



Figura 3.2. Piano di volo definito con l'applicazione DJI GS Pro - II.

3.1.3 Riprese delle immagini

Nel periodo fra Giugno e Luglio 2021, sono state effettuate quattro sessioni di riprese, in cui sono stati eseguiti i piani di volo visualizzabili nelle figure 3.1 e 3.2.

Per ogni ripresa, considerando che le piante monitorate erano quaranta, sono state collezionate duecentoquaranta immagini. Infatti per ogni scatto del drone vengono scattate un'immagine per ogni sensore, che sono in totale sei: uno RGB e cinque per ogni banda dello spettro elettromagnetico visibile (blu a 450 nm, verde a 560 nm, rosso a 650 nm, red-edge a 730 nm e near-infrared a 840 nm).

Per questo motivo il formato delle immagini è di due tipologie differenti, in base al tipo d'informazione che necessitano rappresentare:

- Formato JPEG (estensione .JPG), utilizzato per le immagini RGB (scattate sull'intero spettro luminoso della luce visibile); nonostante il formato JPEG sia un formato compresso con perdita per la rappresentazione d'immagini, viene utilizzato dal drone al fine di ridurre la dimensione totale delle immagini scattate a ogni ripresa, considerando l'elevato numero delle medesime;
- Formato TIFF (estensione .TIF), utilizzato per le immagini multispettrali (scattate in una singola banda dello spettro luminoso visibile); il formato TIFF permette di salvare una maggior quantità di dati in un formato compresso senza perdita, e permette una rappresentazione corretta e trasportabile dei dati multispettrali visto che i livelli dimensionali possono essere anche in un numero elevato, come i livelli multispettrali catturati dalla fotocamera.

Durante le riprese gli esperti botanici hanno effettuato delle misurazioni e dei campionamenti, sia fisiologici sia patologici, al fine di poter definire una pianta sana o malata, nello specifico periodo di tempo, per ognuno dei due punti di vista. Queste misurazioni hanno permesso l'etichettatura delle immagini utile all'analisi di classificazione descritta nella sezione [3.3](#).

Nelle ultime due riprese sono state scattate delle immagini in più riguardanti alcune piante riconosciute dagli esperti botanici come malate dal punto di vista patologico, per un totale di venticinque piante aggiuntive risultanti in ulteriori centocinquanta immagini fra le due riprese. Inoltre alcune immagini aggiuntive sono state recuperate da alcune prove effettuate durante i sopralluoghi preliminari attraverso un ritaglio delle stesse, in modo che la pianta fosse al centro dell'inquadratura e occupasse al massimo l'immagine, rimanendo comunque interamente al suo interno.

3.2 Repository di salvataggio

Al fine di collezionare le immagini scattate durante riprese effettuate con il drone, è stato creato e utilizzato un repository con una struttura gerarchica per campo e giorno di ripresa. Nelle cartelle sono state inserite tutte le immagini con l'aggiunta di dati utili e necessari alla loro analisi successiva in documenti a sé stanti.

Il concetto alla base della definizione di specifiche tecniche del formato di salvataggio dei dati è quello di permettere una ripetibilità della collezione dei dati e delle analisi, avendo sempre il medesimo interfacciamento fra script automatici e repository, non richiedendo quindi ulteriori modifiche future rilevanti.

3.2.1 Locazione

Il repository è posto in cloud attraverso l'utilizzo del software Dropbox, messo a disposizione dal Politecnico di Torino attraverso le sue licenze. Dropbox mette a disposizione delle API che sono risultate utili nel meccanismo di estrazione automatica del dataset usato per i task di classificazione. Un approfondimento di queste e in particolare dello SDK Python viene descritto nella sottosezione [3.3.1](#).

3.2.2 Struttura

Il repository è strutturato secondo uno schema gerarchico. Alla radice vi sono una directory per ogni campo differente su cui sono state effettuate delle riprese, il cui nome è un codice univoco utile all'identificazione del campo (con il formato *C_<numero di quattro cifre>*).

Attualmente il campo utilizzato è solamente uno, ma la struttura è stata ideata e realizzata nel caso in futuro la ricerca proseguisse e comprendesse anche altri campi per effettuare le riprese.

Allo stesso livello delle directory principali può essere posizionato un documento che contiene una serie di dati utili alla descrizione di ogni singolo campo, come ad esempio la sua locazione geografica.

All'interno di ognuna delle directory per campo sono contenute differenti cartelle, una per ogni giorno di ripresa. Per questo motivo il loro nome è identificato da una data, rappresentata secondo il formato ISO-8601¹ (anno-mese-giorno). Nelle cartelle di ogni ripresa possono essere presenti, oltre alle immagini collezionate in quella specifica sessione, una serie di documenti per il salvataggio di dati utili all'analisi delle immagini, come ad esempio le etichette utili al task di classificazione.

Nella figura 3.3 è rappresentato uno schema che descrive a livello visivo tutto ciò che è stato descritto in questa sottosezione.

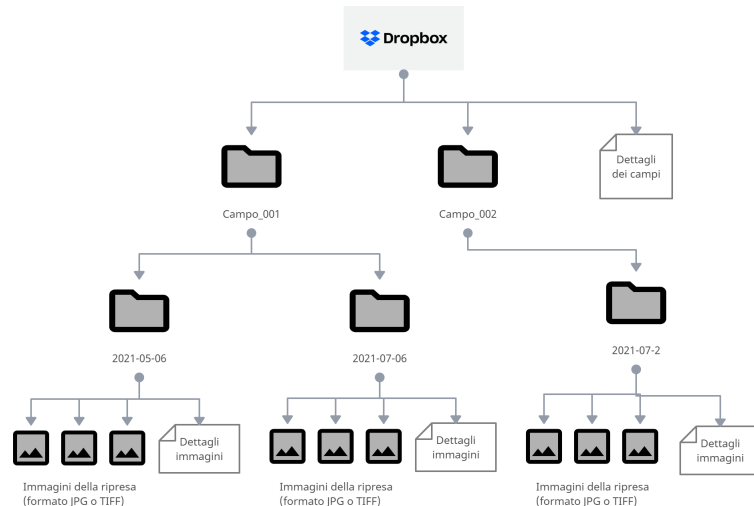


Figura 3.3. Struttura del repository di salvataggio delle immagini.

3.2.3 Metadati delle immagini

Ogni immagine possiede un set di dati automaticamente popolati dal software interno al drone:

- **Dati riguardanti le caratteristiche dell'immagine**, come ad esempio risoluzioni orizzontale e verticale, profondità di bit;
- **Dati riguardanti le caratteristiche della fotocamera** che ha scattato l'immagine, come il produttore, il modello, l'ampiezza del diaframma, il tempo di esposizione;
- **Dati satellitari** identificati come coordinate GPS, riguardanti il luogo in cui l'immagine è stata scattata: latitudine, longitudine e altitudine.

Questi dati vengono salvati seguendo la specifica Exif, una specifica utilizzata dai produttori di macchine fotografiche che permette di salvare questi dati direttamente all'interno del file dell'immagine, senza necessità di cambiare formato. I formati supportati da questa specifica sono JPEG, TIFF e PNG. La sua definizione è stata effettuata per la prima volta dalla Japan Electronic Industries Development Association (JEIDA) nel 1995, e la versione in uso dalle immagini collezionate è la 2.30² definita nel 2010.

¹<https://www.iso.org/iso-8601-date-and-time-format.html>

²https://www.cipa.jp/std/documents/e/DC-008-2012_E.pdf

3.2.4 Formato dei dati aggiuntivi

I dati aggiuntivi utili alle analisi successive delle immagini vengono salvati e rappresentati all'interno dei documenti attraverso uno schema tabellare: nella prima riga sono contenuti i nomi delle colonne della tabella, suddivisi singolarmente dal carattere tabulazione (`\t`), mentre nelle righe successive sono contenute le informazioni relative a ogni singola immagine, anch'esse suddivise dal carattere tabulazione per ogni colonna (`\t`).

Attraverso questo formato le informazioni possono essere lette comodamente sia da un individuo, sia possono essere trattate attraverso l'utilizzo di sistemi automatici come script, visto che esso risulta molto simile al formato CSV³.

Le colonne definite all'interno dei file sono:

- **Nome_immagine** che contiene il nome dell'immagine definito direttamente dal software del drone, per cui composto da un prefisso *DJI*, un numero di quattro cifre sequenziale, legato all'ordine di scatto dell'immagine, e infine dall'estensione dell'immagine (esempio *DJI_0010.jpg*);
- **Label_Fisio_(sano-0/malato-1)** e **Label_Pato_(sano-0/malato-1)** che contengono rispettivamente un valore fra 0 e 1 che indica se la pianta ripresa da quell'immagine è stata definita malata (valore 1) o sana (valore 0) da parte degli esperti botanici; la prima colonna rappresenta il punto di vista fisiologico, mentre la seconda quello patologico;
- **Plant_id** che contiene un id univoco a livello di campo utile a identificare la medesima pianta fra differenti immagini scattate, che potrebbe eventualmente permettere di delineare un'evoluzione della stessa durante le successive sessioni di riprese; l'id è definito con un prefisso *P_* seguito da un numero di quattro cifre;
- **Dati fisiologici** che sono un insieme di colonne che possono contenere differenti dati fisiologici misurati da parte degli esperti botanici nella specifica sessione di ripresa; nel caso in cui il dato non sia presente per quella pianta, viene utilizzato il carattere / per indicarne l'assenza.

Nella tabella 3.1 è possibile trovare un esempio rappresentativo del formato del file e delle colonne definite al suo interno.

Script per la definizione dei file

Al fine di velocizzare e automatizzare parte della scrittura dei file contenenti i dettagli delle immagini, sono stati implementati due script in linguaggio Python.

Il primo permette di generare il file con le informazioni aggiuntive riguardo le immagini, secondo il formato definito precedentemente in questa sezione. Richiede la configurazione di cinque parametri d'input, quattro dei quali si basano sul numero dell'immagine, cioè il valore contenuto nel nome successivamente al prefisso (*DJI_<numero>.<estensione>*):

1. **initial_path** che indica quale cartella d'immagini deve essere analizzata per popolare le righe del file;
2. **label_fisio_0_numbers** array di valori che contiene i numeri delle immagini delle piante sane fisiologicamente;
3. **label_fisio_1_numbers** array di valori che contiene i numeri delle immagini delle piante malate fisiologicamente;
4. **label_pato_0_numbers** array di valori che contiene i numeri delle immagini delle piante sane patologicamente;

³<https://datatracker.ietf.org/doc/html/rfc4180>

5. **label_pato_1_numbers** array di valori che contiene i numeri delle immagini delle piante malate patologicamente;

Il file viene generato automaticamente nel percorso indicato nel parametro *initial_path*, con il nome di *dettagli_immagini.txt*. Esso contiene i valori popolati per ogni immagine trovata all'interno di tutta la cartella indicata, comprese le sottocartelle, seguendo come parametro di filtro le estensioni dei file, che in questo caso possono essere solamente .jpg o .tiff. I valori popolati sono legati solamente alle prime tre colonne.

Il secondo script permette di definire l'id univoco per ogni pianta e di popolare correttamente i file di *dettagli_immagini.txt* trovati. Di conseguenza questo script, in una situazione di prima esecuzione, deve essere eseguito in seguito a quello precedentemente descritto.

La logica di questo script è legata alla lettura dei dati GPS delle immagini, in quanto questo dato permette d'identificare, con una certa approssimazione, quali immagini corrispondano alla medesima pianta. Al fine di compiere questa operazione, viene utilizzato un modulo di Python esterno chiamato *exifread*. Questo modulo permette di poter leggere i dati aggiuntivi delle immagini, secondo la specifica Exif, tramite un dizionario composto da chiave-valore. Di conseguenza, dopo aver estratto il dizionario dei metadati Exif contenuto in una specifica immagine, lo script estrae le coordinate GPS utilizzando le chiavi univoche definite dal modulo per leggere queste informazioni. In seguito compara le coordinate dell'immagine a ogni coordinata di ogni pianta già definita in precedenza, al fine di determinare il suo id: se viene trovato un valore simile, basandosi su un valore di tolleranza passato in input, l'immagine verrà legata all'id trovato, altrimenti verrà definito un nuovo id e le coordinate verranno salvate come riferimento di confronto per le immagini successivamente analizzate. Infine viene aggiornato il file *dettagli_immagini.txt*, relativo alla ripresa di cui l'immagine fa parte, popolando la colonna *Plant_id* per la riga collegata all'immagine.

I valori d'input di questa funzione sono tre:

1. **tolerance** che contiene il valore massimo di differenza per i secondi per le metriche di longitudine e di latitudine di due coordinate GPS;
2. **initial_path**: percorso della cartella iniziale in cui tutte le immagini sono contenute;
3. **allowed_path**: elenco di percorsi in cui lo script può navigare per trovare le immagini da analizzare; parametro inserito al fine di escludere eventuali cartelle ridondanti o che contengono immagini che non devono essere analizzate dalla logica.

I risultati di questo script vanno controllati manualmente al fine di essere sicuri che la relazione immagine-plant_id sia corretta. Questo è dovuto alla relativa precisione del GPS che, nel momento in cui si tratta di distinguere fra oggetti a distanza di pochi metri, può risultare a volte fallace. Il valore di tolleranza, che introduce un concetto di approssimazione, attenua leggermente questo problema, ma in alcuni casi non lo risolve completamente.

Popolazione dei dati fisiologici

I dati estratti dai campionamenti botanici al fine dell'analisi fisiologica delle piante sono stati inseriti all'interno dei file di dettagli aggiuntivi riguardo alle immagini manualmente, questo principalmente per due motivi:

- i dati erano disponibili da fonti esterne che non offrivano la possibilità di lettura o estrazione automatica;
- i dati riguardavano un numero limitato di piante per ogni ripresa.

Per queste ragioni la creazione di uno script automatico non è stata necessaria, considerando anche il fatto che un eventuale script avrebbe richiesto una quantità di dati in input consistente, la cui configurazione sarebbe stata pari allo sforzo d'inserimento manuale effettuato.

<i>Nome_immagine</i>	<i>Label_Fisio_(sano-0/malato-1)</i>	<i>Label_Pato_(sano-0/malato-1)</i>	<i>Plant_id</i>	<i>Dato_Fisio_1</i>
<i>DJI_0010.jpg</i>	0	0	<i>P_0001</i>	306,75
<i>DJI_0011.tiff</i>	1	0	<i>P_0002</i>	/
<i>DJI_0120.jpg</i>	1	1	<i>P_0010</i>	265,05

Tabella 3.1. File di esempio contenente i dati aggiuntivi delle immagini.

3.2.5 File di supporto

All'interno della cartella specifica per il singolo campo, è possibile andare a inserire alcuni file di supporto utili all'analisi delle immagini oppure ad aumentare la comprensibilità generale dei dati contenuti al suo interno. Nel caso specifico del campo analizzato nel 2021, sono stati creati tre differenti tipologie di file di supporto:

1. un file contenente i dati fisiologici misurati durante le diverse sessioni di ripresa da parte degli esperti botanici;
2. un file contenente una matrice che permette di correlare l'id della singola pianta alla sua effettiva posizione all'interno del campo, indicata attraverso il numero di filare (colonna) e la sua posizione all'interno del filare (riga); in questo modo ogni pianta è identificabile a livello fisico durante un sopralluogo futuro;
3. un file contenente l'elenco d'id delle piante alle quali sono state scattate immagini aggiuntive durante le ultime due sessioni di riprese, come descritto nella sottosezione 3.1.3.

3.3 Dataset per l'analisi di classificazione

Dall'insieme delle immagini collezionate con le riprese effettuate fra Giugno e Luglio del 2021, si sono estratte le immagini RGB in formato JPEG al fine di determinare un dataset adatto alla classificazione. In totale le immagini all'interno del dataset sono state centonovantaquattro.

Si è deciso che per le prime prove di classificazione, al fine d'individuare i modelli utilizzabili e performanti per lo scopo, le eventuali problematiche possibili durante la configurazione delle prove e i possibili miglioramenti per le prove successive, essa fosse di tipo binario, cioè che il dataset fosse suddiviso in sole sue classi differenti: piante sane, con la label 0, e piante malate, con la label 1. Considerando che per ogni immagine le etichette collegate sono due, si è determinato che le piante sane dovessero avere sia l'etichetta fisiologica sia quella patologica a 0, mentre le piante malate potessero avere uno dei due valori a 1. Attraverso questa logica si è determinata la suddivisione e, di conseguenza, la distribuzione del dataset sulle due differenti classi: centoquarantasette immagini appartenenti alla classe 0 (piante sane), e quarantasette immagini appartenenti alla classe 1 (piante malate); con una distribuzione percentuale del 76% per la classe 0 e del 24% per la classe 1. Nel grafico a torta 3.4 è possibile osservare visivamente la distribuzione del dataset fra le due classi.

3.3.1 Estrazione automatica del dataset

Al fine di automatizzare l'estrazione del dataset dall'insieme delle immagini collezionate e contenute all'interno di un repository Dropbox, è stato utilizzato il supporto delle API definite da Dropbox all'interno di un SDK in codice Python, visto che gli script di configurazione e definizione dell'analisi di classificazione sono stati scritti nel medesimo linguaggio.

Le API messe a disposizione sono interrogabili attraverso il protocollo HTTP, utilizzando delle richieste in metodo POST con strutture specifiche per i parametri della richiesta. Utilizzando uno degli SDK messi a disposizione non è necessario implementare e configurare manualmente

le richieste, ma basta utilizzare i metodi messi a disposizione direttamente dalle librerie di supporto. Nel caso specifico del Python, è necessario importare e utilizzare il modulo *dropbox*. La documentazione completa è disponibile all'interno del sito di Dropbox Developers⁴.

A questo punto sono necessarie due operazioni distinte, al fine di poter effettuare le chiamate verso un specifico repository.

Abilitazione del repository alle API

È necessario che il repository che si vuole utilizzare sia abilitato all'utilizzo delle API, cioè che diventi non solo una piattaforma di gestione di file attraverso l'interfaccia di Dropbox, ma anche attraverso le chiamate HTTP definite dal software. Per fare ciò basta accedere con le proprie credenziali alla console delle applicazioni⁵ di Dropbox e abilitare il proprio repository all'utilizzo delle API. Nell'immagine 3.5 è possibile osservare la prima schermata visualizzabile all'accesso alla console dell'applicazione. Da qui sono configurabili diverse impostazioni che riguardano l'autenticazione, l'autorizzazione, il branding e l'osservazione delle statistiche di utilizzo dell'applicazione stessa.

La parte significativa riguarda l'autenticazione attraverso un token *OAuth2*, utile per utilizzare qualsiasi endpoint dell'API. Infatti i permessi di sicurezza sono incrementali, e dipendono da quali operazioni si vogliono effettuare all'interno del repository interpellato. Il numero di permessi configurabili è cospicuo, ma in generale si suddividono in due categorie principali: permessi individuali e permessi di team. I primi riguardano la possibilità di leggere o di scrivere (permessi distinti) alcune tipologie di dati, come ad esempio il contenuto dei file, i metadata degli stessi o i contatti degli utenti che hanno accesso al repository. I secondi riguardano, invece, la gestione dei team, interessando quindi la modifica dei membri di ogni team, la lettura dei log delle attività effettuate all'interno del repository e la gestione dei gruppi. Per il nostro scopo è stato necessario creare un token a durata illimitata contenente i permessi di lettura del contenuto dei file e quelli di default, cioè lettura dei metadata dei file e lettura delle info di un account. Nell'immagine 3.6 è possibile osservare la schermata d'impostazione dei permessi di questo specifico caso.

Utilizzo del token di sicurezza

Una volta ottenuto un token valido, è possibile utilizzare un oggetto Dropbox, la cui classe è definita all'interno delle librerie Python di Dropbox, per poter effettuare le chiamate HTTP agli endpoint dell'applicazione.

Nel caso specifico degli script di analisi di classificazione delle immagini, il metodo utilizzato è stato quello di download dell'intera cartella del repository attraverso l'utilizzo del metodo *files_download_zip_to_file*, che richiede come parametri il path della cartella all'interno del repository che si vuole scaricare e il path di destinazione all'interno del sistema chiamante. Dopo il download del pacchetto, i dati del repository vengono analizzati e rimaneggiati in modo da estrarre il dataset completo.

⁴<https://www.dropbox.com/developers/documentation>

⁵<https://www.dropbox.com/developers/apps>

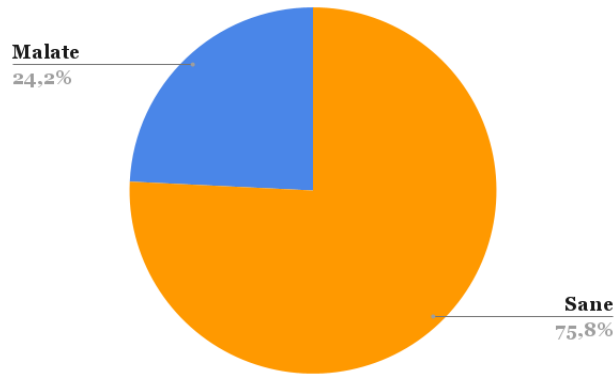


Figura 3.4. Distribuzione del dataset fra le due classi.

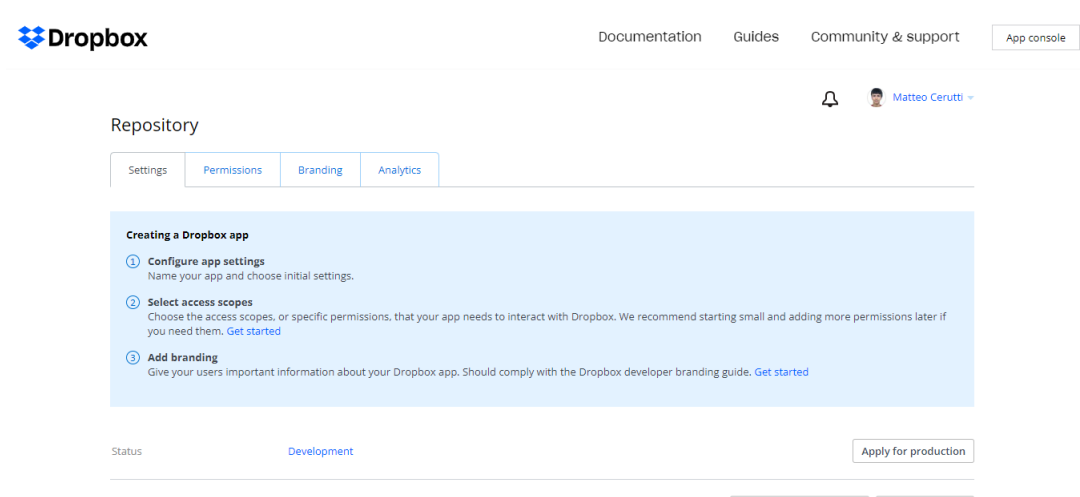


Figura 3.5. Schermata iniziale delle impostazioni di un'applicazione Dropbox.

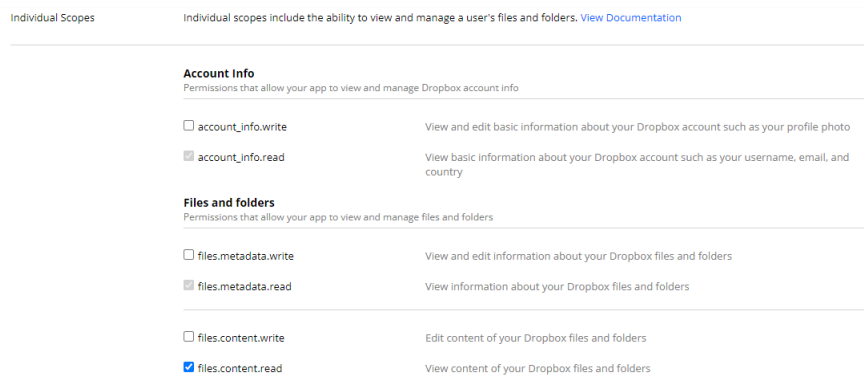


Figura 3.6. Insieme dei permessi abilitati per la definizione del token di sicurezza.

Capitolo 4

Analisi di classificazione delle immagini

4.1 Selezione di lavori di ricerca simili in letteratura

Al fine di poter selezionare i modelli migliori di reti neurali utili alla classificazione delle immagini, considerando l'approccio come uno dei più promettenti nell'ambito dell'intelligenza artificiale applicata all'agricoltura di precisione, si è effettuata una ricerca in letteratura al fine di trovare lavori di studio simili a quelli proposti da questo lavoro di tesi.

4.1.1 Modalità di ricerca

Per compiere questa ricerca è stato utilizzato il portale di Google Scholar¹, una piattaforma che permette di cercare in contemporanea fra differenti fonti autorevoli in termini di articoli e documenti scientifici di ricerca. In particolare i documenti selezionati in questa sezione provengono maggiormente dalle piattaforme di ScienceDirect² e Springer Link³.

Le chiavi di ricerca sono state principalmente tre: **deep learning**, **plant images** e **drones**. Dalla combinazione di queste si sono ricavati gli effettivi due ambiti di ricerca che sono stati utilizzati per selezione i progetti di lavoro affini a questo: **deep learning on plant images** e **deep learning on plant images from drones**.

4.1.2 Lavori di ricerca selezionati

Le ricerche selezionate di cui verrà descritto nelle sottosezioni seguenti sono state suddivise in base alle parole di ricerca con cui sono state trovate all'interno di Google Scholar.

Deep learning on plant images

I casi di studio inerenti a questo ambito hanno raggiunto i loro risultati, nella maggior parte dei casi, attraverso l'utilizzo d'immagini delle foglie delle piante analizzate. Ciò che varia da una ricerca all'altra è principalmente l'obiettivo che si vuole raggiungere con l'utilizzo di modelli di rete neurale, che quindi possono aver portato a utilizzare modelli creati appositamente per lo scopo e di tipologia differente rispetto alle reti neurali convoluzionali più largamente utilizzate.

¹<https://scholar.google.com/>

²<https://www.sciencedirect.com/>

³<https://link.springer.com/>

Nel documento *Growing period classification of Gynura bicolor DC using GL-CNN* (Hao et al. 2020)[1], il modello di classificazione proposto è stato chiamato GL-CNN (Global-local convolutional neural network), una rete neurale convoluzionale basata sul concetto di fusione, cioè di estrazione delle caratteristiche delle immagini viste sia nel loro insieme (parte global), sia in modo parziale (parte local) contemporaneamente, avendo due parti di rete distinte per l'estrazione delle singole features e in seguito l'unione dei valori estratti per determinare la classificazione. Il dataset utilizzato era composto da immagini di foglie di piante seminate e raccolte appositamente per lo scopo dello studio. In questo caso l'efficacia di questa rete è stato confrontato con differenti altri modelli di reti neurali convoluzionali: AlexNet, VGG-16, GoogleNet, ResNet-50 e DenseNet. L'accuratezza del test è stato: per GL-CNN il 95,63%, che è stato il migliore nel compito di classificazione, per AlexNet il 91.01%, per GoogleNet il 95.20%, per VGG16 il 83.04%, per ResNet50 il 91.30% e per DenseNet il 92.60%.

Anche nella ricerca *Plant disease identification using explainable 3D deep learning on hyperspectral images* (Nagasubramanian et al. 2019)[2] si è realizzato un modello specifico di rete neurale composto da due livelli convoluzionali, indirizzato all'analisi della malattia denominata "marciume carbonioso", una malattia fungina del suolo che colpisce la colture di soia. Le immagini utilizzate sono state multispettrali, mantenendo interamente tutti i differenti spettri contemporaneamente come livelli della medesima immagine. La rete ha mostrato un'accuratezza di classificazione del 95,73% e un F1 score della classe infetta dello 0,87. È stato inoltre osservato che le lunghezze d'onda più significative per la classificazione sono state quelle appartenenti alla regione del near-infrared.

Per il progetto di lavoro di cui si parla nel documento *An explainable deep machine vision framework for plant stress phenotyping* (Ghosal et al. 2018)[3] è stato definito un nuovo modello di rete neurale composto da cinque livelli convoluzionali, al fine di renderlo in grado di effettuare una misura quantitativa della gravità dello stress subito da una pianta, attraverso la sua identificazione, la sua classificazione e la sua quantificazione. L'accuratezza di testing raggiunta è stata del 94.13%. Il modello si è dimostrato essere robusto alle perturbazioni dell'immagini d'input, ed essere agnostico alle specie, dimostrando quindi una capacità di transfer-learning.

Nell'articolo di ricerca *Deep Learning for Plant Identification in Natural Environment* (Sun et al. 2017)[4] si descrive come è possibile modificare la struttura di una rete neurale già esistente e largamente diffusa, ResNet, per adattarla ai propri scopi rendendola composta da 26 livelli convoluzionali. Per questo è stata nominata ResNet26. Nel lavoro di ricerca è stato utilizzato un dataset di 10.000 immagini di cento specie di piante ornamentali collezionato tramite telefono cellulare nel campus della Beijing Forestry University. Come termine di paragone sono stati utilizzati i modelli di ResNet a 18 strati (ResNet18), a 34 strati (ResNet34) e a 50 strati (ResNet50). Le accuratze raggiunte dai modelli sono state: per ResNet18 il 89.27%, per ResNet34 il 88.28%, per ResNet50 il 86.15% e per il nuovo modello ResNet26 il 91,78%.

Nel caso di studio descritto nel paper *Analysis of transfer learning for deep neural network based plant classification models* (Kaya et al. 2019)[5], invece, si è deciso di utilizzare un approccio differente, andando ad analizzare nello specifico il comportamento dei modelli di rete neurale nel caso di transfer-learning, cioè di estrazioni di features utilizzando modelli differenti e dataset differenti. Nello specifico sono stati utilizzati quattro dataset pubblici. I vari esperimenti effettuati sono stati tipologie differenti, in modo da poter effettuare una comparazione significativa e completa:

- l'utilizzo di un modello custom definito per un allenamento senza transfer learning;
- l'utilizzo di AlexNet e VGG16 per il transfer learning attraverso i dataset;
- l'utilizzo di AlexNet e VGG16 per l'estrazione delle deep features classificate in seguito con l'utilizzo degli algoritmi LDA (Linear Discriminant Analysis) e SVM (Support Vector Machines) con kernel lineare;
- l'utilizzo di AlexNet e VGG16 per l'estrazione delle deep features poi combinate attraverso la fusione con una rete convoluzionale ricorrente per la classificazione.

Lo studio sperimentale ha dimostrato che il transfer learning può fornire importanti benefici e può migliorare i modelli di classificazione a basse prestazioni, raggiungendo accuratèzze al di sopra del 90% per tutti gli esperimenti effettuati che includevano questa tecnica.

Un altro studio indirizzato all'aumento di dati nel caso di piccoli dataset è quello descritto nel paper *Plant disease identification from individual lesions and spots using deep learning* (Barbedo 2019)[6]. Nell'articolo viene esplorata la possibilità di utilizzare parti molte specifiche della foglia di una pianta, anziché considerarla nella sua interezza. Poiché ogni parte ha le proprie caratteristiche, la variabilità dei dati viene così aumentata e ciò permette anche l'identificazione di più malattie sulla stessa foglia, oltre a garantire un numero più elevato di dati per singola foglia. Il modello di rete neurale convoluzionale utilizzato è stato GoogLeNet. Le accuratèzze ottenute con questo approccio sono state, in media, del 12% superiori a quelle ottenute utilizzando le immagini intere, mantenendo i loro valori al di sopra del 75% nel caso di considerazione di dieci malattie differenti come classi del dataset.

Infine nel lavoro descritto nell'articolo *Deep learning models for plant disease detection and diagnosis* (Ferentinos 2018)[7], sono stati allenati alcuni modelli di CNN per eseguire il rilevamento e la diagnosi delle malattie delle piante senza effettuare nessuna modifica alla loro struttura. L'addestramento è stato eseguito con l'uso di un dataset di 87.848 immagini, contenente venticinque specie piante che, combinate a differenti malattie o all'assenza di esse, hanno generato cinquantotto classi distinte di combinazioni. I modelli utilizzati sono stati AlexNet, AlexNetO-WTBn, GoogLeNet, Overfeat e VGG. Le accuratèzze risultate dal testing sono state: per AlexNet il 99.06%, per AlexNetOWTBn il 99.44%, per GoogLeNet il 97.27%, per Overfeat il 98.96% e per VGG il 99.48%.

Deep learning on plant images from drones

Il punto comune fra i progetti di ricerca descritti all'interno dei documenti presentati in questa parte è quello che riguarda il dataset, visto che sono tutti composti da immagini collezionate attraverso l'utilizzo di droni e riguardanti aree vegetali oppure colture di piante. Questi dataset sono poi stati analizzati con l'utilizzo di modelli di deep learning.

Nell'articolo *Evaluating techniques for mapping island vegetation from unmanned aerial vehicle (UAV) images: Pixel classification, visual interpretation and machine learning approaches* (Hamylton et al. 2020)[8] viene descritto il progetto di ricerca sviluppato nell'ambito della riabilitazione della riserva naturale delle Cinque Isole, nell'area di Wollongong in Australia, che consisteva nell'estirpazione di piante dannose e la reintroduzione di piante autoctone. Il dataset è stato generato attraverso le immagini collezionate durante quattro indagini aeree di Big Island effettuate con l'utilizzo di droni, tra aprile 2017 e luglio 2018. L'obiettivo del lavoro è stato quello di rilevare gli esemplari delle nuove piante introdotte, seguendo tre differenti approcci: la definizione di un algoritmo di classificazione pixel-based utilizzando le immagini multispettrali raccolte; l'interpretazione visiva manuale della vegetazione all'interno delle immagini; l'applicazione di un algoritmo di apprendimento automatico basato sul modello rete neurale convoluzionale LeNet. La comparazione dell'efficacia dei tre approcci è stata valutata prima restringendo il campo di valutazione a un'area ristretta, con il risultato del 82%, 91% e 85% rispettivamente; in seguito la valutazione è stata estesa a tutta l'area dell'isola, raggiungendo i migliori risultati per il terzo approccio. L'utilità di considerare approcci differenti è evidenziata dal fatto che i essi hanno portato a mappe della vegetazione marcatamente diverse, offrendo quindi tipi d'informazioni diverse sul carattere della stessa.

Anche nel progetto di studio descritto nel documento *Automated crop plant counting from very high-resolution aerial imagery* (Valente et al. 2020)[9] è stato utilizzato un modello di rete neurale convoluzionale non modificato rispetto alla versione originale: AlexNet. L'unica differenza a livello di training è stato il fatto che il modello è stato utilizzato pre-allenato su un altro dataset, ImageNet, per cui si è effettuata un'operazione di fine tuning dei parametri dell'algoritmo. Il dataset realizzato era composto da immagini UAV ad altissima risoluzione, collezionate attraverso l'utilizzo di droni, di piante di spinaci di dieci settimane seminate all'interno di un campo sperimentale con una superficie di 3,2 ettari. I dati di validazione e testing dell'efficienza dell'algoritmo erano disponibili per un ottavo della superficie totale. I risultati hanno mostrato

che la metodologia proposta può contare le piante con una precisione del 95% per una risoluzione spaziale di 8 mm/pixel in un'area fino a 172 metri quadrati, mentre sulla superficie totale si ha avuto un errore del 42,5%.

Nell'articolo *An Automated Framework for Plant Detection Based on Deep Simulated Learning from Drone Imagery* (Hosseiny et al. 2020)[10], invece, viene presentato un caso di studio in cui si è utilizzato un modello di rete neurale largamente utilizzato, ResNet101, all'interno di un'architettura più complessa, la R-CNN (Regional Convolutional Neural Network), utile all'identificazione di oggetti all'interno delle immagini. L'obiettivo preciso è stato quello d'identificare le piante per contarne il numero all'interno delle immagini analizzate. L'efficienza del modello allenato è stata valutata da due diversi set d'immagini di due campi di grano ottenuti attraverso l'utilizzo di droni. I risultati mostrano che il metodo proposto porta a una precisione media di conteggio del 90,9%. Inoltre, sulla base della distanza media di Hausdorff, è stato ottenuto un errore medio di localizzazione degli oggetti di 11 pixel.

Infine nella ricerca presentata nel documento *UAV and a deep convolutional neural network for monitoring invasive alien plants in the wild* (Qian et al. 2020)[11], è stato progettato per lo scopo specifico un nuovo modello di rete neurale, denominato IAPsNet. Il dataset utilizzato era composto da 6400 campioni suddivisi in sette classi di piante invasive, estratte con l'utilizzo di droni da un'area selvatica monitorata. Al fine di confrontare i risultati di questo nuovo modello, sono state anche allenate e testate tre modelli di CNN largamente utilizzati: LeNet, AlexNet e VGG-11. I risultati in termini di accuratezza sono stati: per LeNet 87.94%, per AlexNet 86.88%, per VGG-11 85.06% e per IAPsNet 93.39%. Inoltre IAPsNet ha dimostrato un'alta capacità anti-interferenza contro la sfocatura e contro l'influenza delle condizioni meteorologiche.

4.2 Modelli di rete neurale selezionati

Attraverso l'analisi dei documenti di ricerca descritti nella sezione precedente, si sono identificati quattro differenti modelli di rete neurale convoluzionale largamente diffusi e utilizzati nella computer vision e dimostratisi validi per lo scopo di questo lavoro di tesi: AlexNet, GoogLeNet, ResNet e VGG.

Fra questi quattro modelli si è ridotta la selezione a tre di questi principalmente per una ragione di velocità a livello di training. Infatti come si evince anche dal paper *Growing period classification of *Gynura bicolor* DC using GL-CNN*[1], fra questi quattro modelli i più veloci risultano AlexNet e GoogLeNet, mentre ResNet e VGG si attestano a tempi di training simili. Fra questi due modelli rimanenti, si è effettuata la scelta in favore di VGG per il semplice fatto che questo modello è stato maggiormente utilizzato nella sua struttura originale all'interno dei casi di studio in letteratura selezionati, raggiungendo risultati soddisfacenti, per cui dimostrandosi robusto ed efficiente per lo scopo di questo lavoro di tesi.

I modelli di rete custom, progettati appositamente nei lavori di ricerca selezionati, sono stati esclusi dai modelli scelti per due motivi principali:

1. questo lavoro di tesi è stato il primo passo affrontato nei confronti dell'analisi delle immagini collezionate dalle riprese descritte nella sezione 3.1, per cui è meglio utilizzare dei modelli più robusti al fine di poter individuare problematiche più generali riguardanti i metodi di analisi e la definizione del dataset, prima d'introdurre nuove variabili riguardo a un modello appositamente progettato per lo scopo;
2. i modelli di rete custom devono essere implementati interamente all'interno dei propri script, visto che non sono inclusi come classi utilizzabili all'interno delle librerie di deep learning, come nel caso di Pytorch utilizzata in questo lavoro di tesi e descritta nella sottosezione 4.3.2; quindi, oltre a richiedere un effort maggiore in termini di sviluppo della logica di analisi, non possono essere utilizzati come modelli pre-allenati, visto che i pesi sono messi a disposizione dalle librerie solamente per i modelli più conosciuti.

In conclusione, i modelli selezionati per la classificazione delle immagini sono stati **AlexNet**, **GoogLeNet** e **VGG**, che verranno descritti più approfonditamente nelle prossime sottosezioni.

4.2.1 AlexNet

Nel 2012 è stato pubblicato il paper *ImageNet Classification with Deep Convolutional Neural Networks* (Alex Krizhevsky et al. 2012)[12] che descriveva una nuova architettura di rete neurale convoluzionale indirizzata all'analisi del dataset ImageNet nel contesto della ImageNet Large Scale Visual Recognition Challenge del 2010 (ILSVRC-2010): AlexNet, dal nome del ricercatore principale del progetto.

Quest'architettura consiste nella definizione di cinque livelli convoluzionali e tre fully connected, per un totale di otto livelli. Nell'immagine 4.1⁴ è possibile visualizzare la struttura della rete con i livelli posti sequenzialmente, con il dettaglio dei dati riguardanti i filtri per i livelli convoluzionali (dimensione, stride e padding) e il numero di neuroni per i livelli fully connected.

Le principali caratteristiche innovative, che hanno portato questo modello a essere utilizzato in modo diffuso negli anni seguenti la sua definizione, sono state:

- **Non linearità della funzione ReLU:** AlexNet utilizza questa funzione all'interno dei neuroni anziché la funzione tanh utilizzata maggiormente in precedenza, dando vantaggi sia nel tempo di training, sia nella riduzione dell'errore di previsione;
- **Uso di GPU multiple:** AlexNet permette di usare GPU multiple per il training offrendo la possibilità di suddividere i neuroni del modello fra le differenti GPU;
- **Pooling sovrapposto:** solitamente all'interno dei livelli di max pooling, la finestra che viene utilizzata per calcolare il massimo valore fra quelli presenti al suo interno viene spostata senza che si sovrapponga fra un ciclo e l'altro; ma in AlexNet le finestre di sovrappongono di 2 pixel fra un'esecuzione e l'altra, portando a una riduzione dell'errore del 0,5%;
- **Dropout:** questa tecnica consiste nello "spegnere" i neuroni di un livello con una certa probabilità a ogni iterazione, cioè che non vengano utilizzati per il calcolo dei valori, in modo che la struttura dei livelli vari a ogni ciclo; in AlexNet è stata applicata ai livelli fully connected, con un fattore di 0,5, al fine di ridurre l'overfit dei dati, ma aumentando anche il tempo di training perché il modello necessita di più iterazioni per convergere.

Attraverso queste innovazioni, questo modello di rete ha potuto raggiungere nella competizione d'ImageNet del 2010 il 47,1% dell'errore top-1 e il 28,2% dell'errore top-5, mentre nel 2012 ha vinto la competizione riproposta per quell'anno nella categoria dell'errore top-5, raggiungendo una percentuale del 15,3%.

4.2.2 GoogLeNet

GoogLeNet è un modello di rete neurale convoluzionale definito per la prima volta da un team di ricercatori di Google e descritto nell'articolo *Going deeper with convolutions*[13] presentato in occasione della ILSVRC del 2014.

La sua architettura è composta da ventidue livelli, non considerando fra questi cinque livelli di pooling, di cui la maggior parte convoluzionali. Nella tabella contenuta nell'immagine 4.4⁵ è possibile osservare nel dettaglio tutti i livelli, con i dati caratteristici di ognuno di esso.

La caratteristica principale di questo modello consiste proprio nei livelli convoluzionali, che non sono composti da filtri di un'unica dimensione, ma che invece sono strutturati in maniera differente in modo da poter avere più filtri eseguiti in parallelo di dimensioni differenti senza la necessità di avere più livelli sequenziali: i livelli inception. Per questo motivo in un singolo livello inception vengono posti filtri di differenti dimensioni, i cui risultati vengono poi concatenati tutti

⁴By Cmglee - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=104937230>

⁵Immagine ottenuta dall'articolo *A Review of Popular Deep Learning Architectures: AlexNet, VGG16, and GoogLeNet* sul sito PaperspaceBlog <https://blog.paperspace.com/popular-deep-learning-architectures-alexnet-vgg-googlenet/>

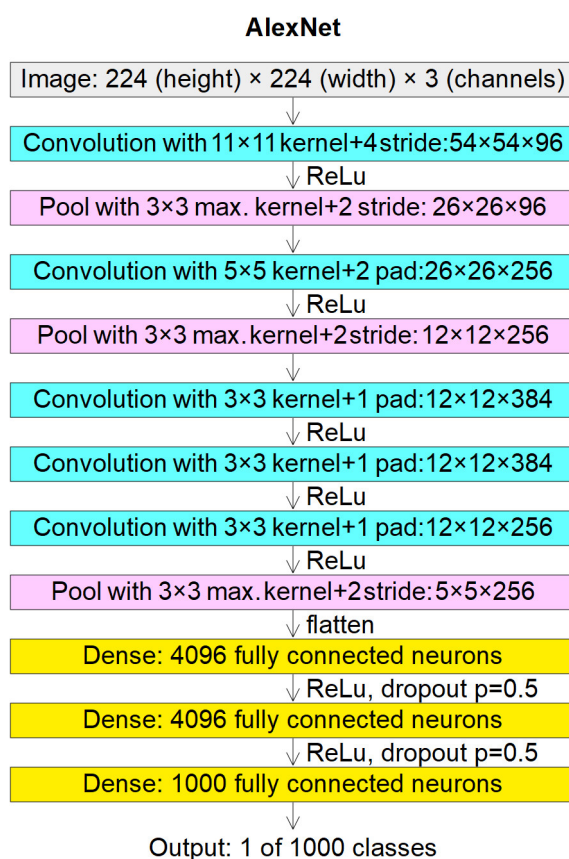


Figura 4.1. Struttura dei livelli sequenziali di AlexNet.

insieme per generare un output unico. In alcuni casi viene aggiunto anche un livello di max pooling in parallelo. La prima versione “naive” non introduceva nessun livello aggiuntivo prima dei filtri di dimensione maggiore a 1×1 , ma a causa della lentezza del training introdotto da questa versione, si è deciso di introdurre un primo livello di filtri di dimensione 1×1 al fine di semplificare l’input per agevolare i calcoli convoluzionali effettuati dai livelli successivi. Nell’immagine 4.2 è possibile avere uno schema visivo della versione dei livelli inception appena descritta e utilizzata nell’architettura. L’obiettivo principale di questo approccio è stato quello di prevenire l’overfit che si può presentare nel caso si abbiano molti livelli convoluzionali successivi.

Una caratteristica ulteriore di questo modello di rete è l’introduzione di alcuni componenti all’interno della rete denominati classificatori ausiliari. Questi componenti sono stati introdotti al fine di ridurre il problema del gradiente evanescente, un problema presente durante l’allenamento di reti con molti livelli che porta i livelli iniziali a non aggiornare i propri pesi con l’aumentare del numero di iterazioni. Infatti questi componenti, posti all’incirca ad un terzo e ai due terzi della profondità della rete, effettuano una classificazione come se fossero i livelli fully connected finali e calcolano un valore di loss che viene sommato a quello totale prima della fase di propagazione inversa, il tutto ad ogni iterazione della fase di training. Nell’immagine 4.3, che mostra l’intera architettura di GoogLeNet, i riquadri in rosso evidenziano proprio questi componenti.

GoogLeNet ha vinto la ILSVRC 2014 nella categoria dell’errore top-5, raggiungendo un valore di 6,67%.

4.2.3 VGG

Nell’ambito della ILSVRC del 2014, oltre a GoogLeNet, è stato presentato un nuovo modello di rete neurale convoluzionale: Visual Geometry Group (VGG), descritta all’interno del documento *Very Deep Convolutional Networks for Large-Scale Image Recognition* (Simonyan et al. 2015)[14].

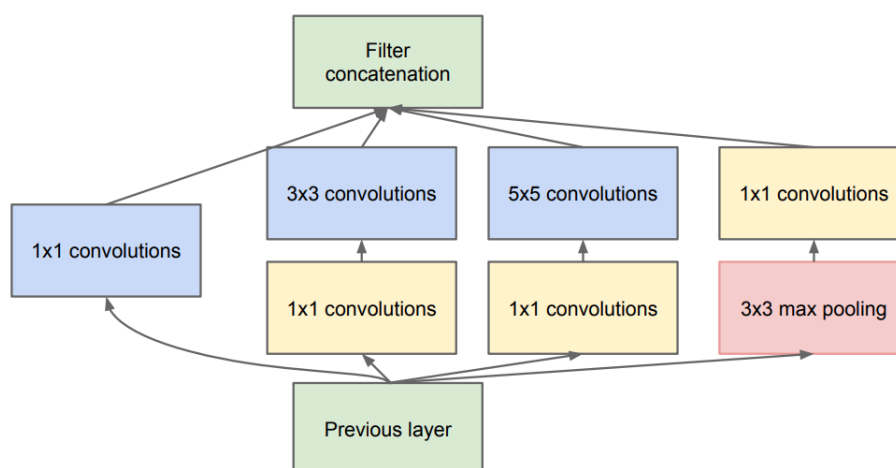


Figura 4.2. Schema di un generico modulo inception.

Quest'architettura si pone l'obiettivo di migliorare le prestazioni della precedente AlexNet, attraverso l'introduzione di un numero maggiore di livelli e, quindi, di una profondità di rete aumentata. A differenza di AlexNet, però, le dimensioni dei filtri utilizzati nei livelli convoluzionali non sono elevate, ma sempre contenute: 3×3 oppure 1×1 . Quest'ultima, in particolare, serve principalmente per introdurre una maggiore non linearità all'interno del modello, senza cambiare la dimensione degli input. Con questa scelta si sono ridotti il numero di parametri rispetto ad AlexNet, nonostante i livelli convoluzionali sia aumentati in numero.

Il modello è stato presentato in differenti configurazioni possibili, che ne aumentano la profondità attraverso l'aumento del numero di livelli convoluzionali, mentre quelli fully connected posti alla fine della struttura sono sempre tre. Le configurazioni diverse partono da undici livelli fino a arrivare a diciannove, passando per i valori di tredici e sedici. Nella tabella contenuta nell'immagine 4.5 sono presentati i dettagli di ogni livello per tutte le configurazioni possibili presentate nel paper citato. Nell'immagine 4.6, invece, si può osservare la riduzione di un'immagine di input a mano a mano che essa viene elaborata attraverso i livelli del modello VGG11.

Nella competizione ILSVRC del 2014 VGG non ha vinto per la categoria dell'errore top-5, visto che è stata superata da GoogLeNet, ma ha raggiunto un valore del 7,3%, molto vicino a quello del modello vincitore. In seguito alla competizione, il valore è stato diminuito fino al 6,8%.

4.3 Strumenti utilizzati

4.3.1 Google Colaboratory

Google Colaboratory è una piattaforma online, accessibile tramite browser, che permette di eseguire codice Python utilizzando sistemi in cloud offerti da Google. Le sue caratteristiche principali sono:

- **Nessuna configurazione necessaria** in quanto basta scrivere il proprio codice in Python, collegarsi a una macchina virtuale disponibile ed eseguire il proprio script; l'environment di esecuzione viene automaticamente generato dalla piattaforma;
- **Accesso gratuito alle GPU e alle TPU** visto che le macchine virtuali disponibili mettono a disposizione risorse di computazione GPU e TPU, molto utili in ambito di analisi d'intelligenza artificiale;
- **Condivisione semplificata** attraverso la scrittura del proprio codice in un blocco note Colab, un documento che viene salvato all'interno del proprio spazio Google Drive e che quindi può usufruire di tutti i meccanismi di condivisione offerti dalla piattaforma.

Blocco note Colab

Il blocco note Colab è un blocco note Jupyter ospitato nella piattaforma Colaboratory, cioè un ambiente interattivo che permette di combinare parti di codice Python con parti di documentazione scritta e immagini, utilizzando anche tag HTML e comandi LaTeX. Questo permette di definire un unico documento il codice, le spiegazioni del suo funzionamento e il commento dei risultati ottenuti, in modo da avere tutte le parti utili alla comprensione e alla discussione della logica insieme.

Le parti di codice possono essere eseguite singolarmente e nell'ordine che si preferisce. Questo permette di avere un'ampia flessibilità nella definizione della logica, senza la necessità di utilizzare variabili o comandi di compilazione per gestire differenti flussi logici o differenti configurazioni.

Limitazioni

La piattaforma è utilizzabile in modalità gratuita, avendo a disposizione circa tredici GB di RAM, circa cento GB di spazio disco e dodici GB di GPU, ma per un periodo di tempo limitato consecutivamente. In seguito a un'esecuzione, che viene interrotta in modo forzato dalla piattaforma, è necessario aspettare a lungo prima di poter di nuovo utilizzare la piattaforma con il medesimo account Google. Esistono una versione Pro e una versione Pro+ a pagamento mensile, che aumentano la disponibilità di risorse RAM e memoria disco, mettono a disposizione GPU e TPU più rapide e aumentano il tempo di esecuzione consecutiva possibile, ma sono disponibili solo in alcuni paesi: Brasile, Canada, Francia, Germania, Giappone, India, Regno Unito, Stati Uniti e Thailandia.

Queste limitazioni hanno comportato principalmente due conseguenze:

- **Dilatazione dei tempi di fine tuning** in quanto effettuare diverse fasi di training, al fine di trovare i valori dei iperparametri migliori per la classificazione, ha comportato dover eseguire un numero limitato di esse in una singola sessione di Colab e ripetere plurime volte i cicli di esecuzione e attesa per poter completare lo scopo;
- **Limitazione nella scelta degli iperparametri** in quanto alcuni iperparametri dei modelli di rete neurale dipendono dalla memoria GPU e avendo a disposizione un limite di questa risorsa e avendo reti e immagini particolarmente grandi, hanno portato a dover utilizzare obbligatoriamente alcuni valori senza possibilità di scelta.

Google Drive

Google Colab permette di aver un'integrazione con Google Drive direttamente all'interno del proprio codice. Questo è stato un vantaggio al fine di ovviare al problema della sessione limitata di esecuzione, che non permette quindi il salvataggio di dati nella memoria disco messa a disposizione poiché temporanea. Di conseguenza nei blocchi note Colab di analisi delle immagini è stata implementata una logica di salvataggio dei log di output direttamente sullo spazio di memoria di Google Drive, attraverso l'utilizzo della funzionalità di mounting.

Questa funzionalità consiste nel montare lo spazio di memoria di Google Drive come se fosse un disco di memoria esterno accessibile direttamente nel file system della macchina virtuale messa a disposizione da Colab. Una volta effettuata questa operazione, tutte le operazioni di lettura e scrittura del file system possono essere eseguite anche sul disco montato, a esclusione delle operazioni di modifica delle directory. L'unica operazione preliminare necessaria è quella dell'inserimento di un token che permette l'accesso dello script allo spazio di memoria, operazione che deve essere eseguita durante l'esecuzione poiché interattiva all'interno della console di I/O della cella eseguita nel proprio blocco note Colab.

4.3.2 Pytorch

Pytorch è una libreria indirizzata al deep learning ottimizzata nell'utilizzo di tensori, che esegue i propri calcoli computazionali sia utilizzando la potenza di calcolo della CPU, sia quella della GPU.

Un tensore è un array multidimensionale composto da numeri negativi e positivi decimali, che può rappresentare sia dati semplici come vettori di numeri, sia dati più complessi come le immagini oppure i pesi dei neuroni all'interno di una rete neurale. L'ottimizzazione nel loro utilizzo sia all'interno della CPU, sia all'interno della GPU è stato raggiunto attraverso l'integrazione di librerie di accelerazione hardware e l'implementazione di nuove logiche di allocazione della memoria.

Pytorch mette a disposizione diversi modelli di rete neurale già implementati in classi utilizzabili direttamente dall'utente, senza la necessità di configurazioni ulteriori. L'obiettivo non è solamente quello di mettere a disposizione delle classi già pronte all'uso per l'utente, ma offrire anche un modello facile da modificare per adattarlo alle proprie esigenze, visto che è possibile vedere l'implementazione di ogni modello e di modificare solamente i layer che si preferiscono. I modelli di reti neurali disponibili non sono solo quelli indirizzati alla classificazione d'immagini, come ad esempio AlexNet, DenseNet e ResNet, ma anche quelli indirizzati alla segmentazione semantica, come le FCN (fully convolutional network), all'identificazione di oggetti, come le Faster R-CNN e le SSD, e alla classificazione di video.

Inoltre Pytorch offre un insieme di dataset d'immagini, largamente utilizzati, pronti all'uso, come ad esempio Caltech256, composto da 30607 immagini varie appartenenti a 256 classi, MNIST, composto da 70000 immagini di cifre numeriche scritte a mano, e CIFAR10, composto da 60000 immagini di veicoli e animali suddivisi in dieci differenti classi. Sia i modelli delle reti neurali, sia i dataset appartengono a una sezione della libreria di Pytorch denominata Torchvision.

4.4 Logica di analisi delle immagini

4.4.1 Grid search e iperparametri perfezionati

Al fine di definire il miglior modello possibile per lo scopo di classificazione, definito in questo lavoro di tesi, si è deciso di utilizzare un metodo di ricerca dei valori degli iperparametri più performanti denominato grid search, cioè ricerca sulla griglia. Questa tecnica consiste nell'identificare gli iperparametri che si vogliono perfezionare, definire un insieme di valori per ognuno di essi e effettuare le fasi di allenamento e validazione per ogni combinazione possibile fra i valori definiti. Per ogni ciclo di allenamento e validazione devono essere salvati i valori delle metriche di valutazione, al fine poi di selezionare la combinazione di valori migliore da utilizzare per la conclusiva fase di testing.

Gli iperparametri selezionati per la grid search sono stati i seguenti:

1. **Learning Rate (LR):** è l'iperparametro che determina con quale velocità la rete possa imparare, visto che rappresenta il fattore con cui i pesi della rete vengono aggiornati a ogni iterazione durante la fase di backpropagation dell'errore;
2. **Weight Decay:** è l'iperparametro di regolarizzazione della funzione di costo che influenza la complessità del modello, al fine di evitare di avere pesi troppo elevati all'interno della rete;
3. **Numero di epoche:** indica il numero di iterazioni effettuate nell'addestramento del modello;
4. **Step Size:** indica il numero di epoche dopo il quale viene decrementato il learning rate; influenza quindi la modalità con cui il modello raggiunge la convergenza.

Fra questi iperparametri è stato escluso il **Batch Size**, che indica il numero di immagini inserite in input prima di effettuare l'aggiornamento dei pesi della rete, a causa dei limiti di Google Colab descritti nella sottosezione 4.3.1. Per questo motivo è stato utilizzato un valore di batch size pari a 2 per i modelli AlexNet e GoogLeNet, mentre pari a 1 per VGG.

I valori utilizzati nella grid search sono stati i seguenti:

- **Learning Rate:** 0,001, 0,0001 e 0,00001;
- **Weight Decay:** 0, 0,001, 0,0001 e 0,00001;
- **Numero di epoche:** da 1 fino a 50;
- **Step Size:** 15, 20 e 35.

Le combinazioni di valori sono state, quindi, trentasei, eseguendo la fase di addestramento per cinquanta epoche e per ognuna di queste effettuando la fase di validazione.

4.4.2 K-fold cross validation

La k-fold cross validation è una tecnica di gestione della suddivisione del dataset. L'idea alla base è quella di dividere il proprio dataset in k sottogruppi di egual dimensione e di effettuare differenti fasi di validazione per k volte, utilizzando k-1 sottogruppi tutti insieme per la fase di training, e il rimanente gruppo per la validazione. Solitamente alla fine dei k cicli, le misurazioni delle metriche effettuate durante ogni epoca e per ogni fase di validazione vengono combinate attraverso una media pesata, ma vi è anche la possibilità di analizzare semplicemente le metriche per scegliere la migliore suddivisione del dataset.

Questa tecnica, quindi, è solitamente utilizzata in fase di validazione e ricerca degli iperparametri migliori, ma può essere utilizzata anche per la scelta della suddivisione fra training e testing set. In quest'ultimo caso, però, sarà necessario effettuare tutte le operazioni di scelta degli iperparametri migliori, di training e testing finale per k volte, siccome si andrà a trattare con nuovo training set finale ogni volta e, quindi, k modelli differenti. Nell'immagine 4.7⁶ è possibile avere una rappresentazione visiva della creazione dei sottogruppi a ogni iterazione nel caso di k pari a cinque.

Lo scopo principale è di limitare i problemi causati dalla scelta dei dati all'interno di ogni set estratto dal dataset. Infatti le tipologie di dati all'interno del dataset potrebbero essere in cardinalità differenti, per cui questa tecnica permette di valutare differenti suddivisioni e di scegliere quella migliore.

Nel caso in cui venga applicata all'interno di una grid search, come presentato in questo lavoro per il modello AlexNet, questa operazione aiuta a sovrastare e limitare eventuali scelte "fortunate" o "sfortunate" di divisione del dataset in training e validation set, andando a effettuare k differenti cicli di training e validation per ogni combinazione d'iperparametri. In questo modo i valori delle metriche misurate per ogni epoca e per ogni ciclo vengono combinate con una media pesata, e permettono di poter scegliere i migliori iperparametri per il modello senza subire l'influenza di uno split sbagliato durante la fase di grid search.

Nel caso specifico presentato, è stato scelto un valore di k pari a cinque al fine di mantenere lo stesso fattore di suddivisione utilizzato nelle analisi di grid search senza k-fold cross validation (Sottosezione 4.4.5). È stata applicata solamente al modello AlexNet in quanto i tempi di grid search sono aumentati di un fattore pari a k, per cui utilizzarlo anche per modelli lenti durante la fase di training, come GoogLeNet e VGG, avrebbe comportato un maggiore dispendio di risorse.

⁶Immagine ottenuta dall'articolo *Cross Validation: cos'è e come usarla nel Machine Learning* <https://pulplearning.altervista.org/cross-validation-cos-e-come-usarla/>

4.4.3 Modelli pre-allenati

Per raggiungere l'obiettivo di classificazione delle immagini di piante sane o malate, si è deciso di utilizzare i modelli di rete neurale selezionati pre-allenati. Questo significa che i pesi dei modelli vengono inizializzati prima della fase di training ai valori che si sono ottenuti attraverso il training su un altro dataset. In questo caso specifico, il dataset di pre-allenamento è ImageNet, uno dei dataset più grandi in termini sia di numero di immagini (14197122), sia in termini di classi in cui può essere suddiviso (21841).

Attraverso l'utilizzo di Pytorch è possibile in modo semplice utilizzare i modelli di rete con i pesi inizializzati correttamente: basta attivare il parametro *pretrained* nel momento in cui si crea il nuovo oggetto rappresentante il modello. Questo è disponibile per tutti i modelli messi a disposizione dalla libreria.

Si è effettuata questa scelta per il fatto che il dataset utilizzato per il training fosse particolarmente esiguo in termini di numero d'immagini. Di conseguenza utilizzare dei modelli pre-allenati può ovviare questo problema e permette comunque ad essi di imparare a distinguere fra le due classi in cui è suddiviso il dataset.

Questo fattore è stato anche identificato con una prova preliminare sul modello AlexNet, in cui si è potuto verificare come, per gli stessi parametri, il modello pre-allenato imparasse a classificare correttamente, mentre quello senza pre-allenamento no. Nei grafici mostrati nelle immagini 4.8, 4.9, 4.10 e 4.11, infatti, si può notare questo paragone a parità di valori degli iperparametri: nel caso della rete pre-allenata, i valori dell'accuratezza sul training set e sul validation set variano, mentre per quella non pre-allenata no; la medesima cosa succede per i valori della matrice di confusione, che per la rete non pre-allenata indicano che essa risponde sempre lo stesso valore di predizione, cioè quello negativo oppure, in rari casi, quello positivo, per tutti i valori di input. I casi riportati sono solo una piccola parte di quelli effettivamente verificati, ma sono già rilevanti e significati per descrivere visivamente le motivazioni della scelta compiuta.

4.4.4 Metriche di valutazione

Al fine di poter valutare le prestazioni dei modelli sia nella fase di ricerca dei valori migliori degli iperparametri, sia nella fase di testing finale con i valori trovati, sono state utilizzate diverse metriche, tutte riconducibili ai valori contenuti all'interno della matrice di confusione. Questa matrice permette d'identificare gli errori di classificazione, in quanto mette in relazione la predizione effettuata dal modello rispetto all'effettiva classe dell'elemento del dataset analizzato.

Nel caso della classificazione con due classi si possono definire quattro metriche che compongono la matrice:

- **True positive (TP)**: numero degli elementi della classe positiva predetti correttamente come positivi;
- **True negative (TN)**: numero degli elementi della classe negativa predetti correttamente come negativi;
- **False positive (FP)**: numero degli elementi della classe negativa predetti erroneamente come positivi;
- **False negative (FN)**: numero degli elementi della classe positiva predetti erroneamente come negativi;

Nello specifico caso di studio di questa tesi, in cui la classificazione è binaria, i valori positive sono quelli associati alla label 1, cioè quella indicante una pianta malata, mentre i valori negative sono quelli associati alla label 0, cioè quella indicante una pianta sana. Di conseguenza i TP indicano piante malate predette malate, i TN piante sane predette come sane, i FP piante sane predette come malate e i FN piante malate predette come sane.

Fra le possibili metriche derivate, cioè che si possono calcolare utilizzando le quattro metriche appena descritte, la più significativa è l'accuratezza, che si calcola dividendo il numero degli

elementi correttamente predetti (TP + TN) per tutti gli elementi presenti nel dataset analizzato (TP + TN + FP + FN). L'unico difetto significativo di questa metrica riguarda la poca robustezza nel caso di dataset sbilanciati: se si predicesse sempre la medesima classe più rappresentata nel dataset, si raggiungerebbe il valore di distribuzione della classe stessa che, nel caso sia un valore elevato, può fuorviare la valutazione avendo un valore elevato di accuratezza. In questi casi è necessario, dunque, valutare anche le altre metriche, in modo da poter capire se il modello sta effettivamente imparando e non sta predicendo sempre il medesimo valore.

Nell'immagine 4.12 è possibile osservare un riassunto di tutte le metriche introdotte dalla matrice di confusione nel caso di classificazione binaria.

4.4.5 Suddivisione del dataset

Il dataset originale è stato suddiviso inizialmente in due sottoinsiemi: il training set, contenente l'80% degli elementi (centocinquanta immagini), e il test set, contenente il restante 20% (trentanove immagini). Questa suddivisione è stata utilizzata nella fase di testing finale, usando il training set per l'allenamento e il testing set per il testing.

Durante la fase di fine tuning degli iperparametri, invece, si è ricavato il set di validazione suddividendo il training set nuovamente con un fattore di 0,8. Il nuovo training set estratto è stato utilizzato per l'allenamento con un set d'iperparametri scelto dalla ricerca su griglia, mentre il validation set è stato utilizzato per valutare l'efficacia del modello appena allenato. Di conseguenza il training set, durante la fase di grid search, era composto da centoventiquattro elementi, mentre il validation set da trentuno elementi.

Tutti gli splitting sono stati effettuati in modo stratificato, cioè mantenendo la distribuzione del dataset di partenza.

4.5 Risultati e osservazioni

4.5.1 Grid search

Nelle heatmap contenute nelle immagini 4.13, 4.15 e 4.16 si possono osservare i valori di accuratezza calcolati dai valori misurati durante la validazione di ogni combinazione di valori d'iperparametri. Questi valori sono le coordinate di ogni heatmap: sull'asse orizzontale sono posti i valori di learning rate, mentre sull'asse verticale sono posti i valori di weight decay. Ogni heatmap, poi, è valida per un determinato valore di step size, indicato nella didascalia di ognuna di esse. Al di sotto di ogni valore di accuratezza è posto fra parentesi il numero di epoche in cui si è raggiunta quell'efficacia.

La scelta dell'epoca corretta da utilizzare è stata effettuata non solo valutando il miglior valore di accuratezza possibile fra le 50 epoche, ma anche considerando che il valore di accuratezza del training set fosse simile e coerente in quella specifica epoca. Questo per evitare di utilizzare un modello che ha già raggiunto l'overfit a quel numero d'iterazioni di training.

La scelta della migliore combinazione di valori da utilizzare per la fase di testing è stata effettuata tenendo in considerazione il valore di accuratezza, cercando di minimizzare il numero di epoche e controllando anche i valori di accuratezza sul training set. Tutto questo per scegliere un insieme di parametri per un modello che avesse effettivamente imparato a distinguere fra le classi, non avesse raggiunto l'overfit generalizzando quindi correttamente lo scopo della classificazione.

I valori di accuratezza minimi, cioè pari a 77,42%, non hanno indicato il numero di epoche per il semplice fatto che, per quella combinazione d'iperparametri, la rete neurale non ha imparato nulla riguardo il problema di classificazione. Questo è stato verificato anche attraverso l'osservazione della matrice di confusione, che ha indicato come la rete classificasse tutti gli elementi come immagini di piante sane. Ciò è confermato anche dal fatto che 77,42% è il valore di distribuzione della classe maggioritaria all'interno del dataset.

I valori selezionati per singola rete sono stati riassunti nella tabella 4.1.

Con 5-Fold cross validation

Utilizzando la tecnica della 5-fold cross validation, si può notare come i valori di accuratezza calcolati durante la fase di validazione siano diminuiti in modo significativo rispetto alle operazioni di grid search effettuate senza utilizzare la cross validation. Inoltre è possibile notare come la distribuzione dei valori migliori sia variata rispetto alla versione della grid search sul modello AlexNet senza la cross validation. Ciò ha comportato la scelta di una differente combinazione d'iperparametri migliori, indicata nella tabella riassuntiva 4.1.

Il paragone è osservabile analizzando le heatmap presenti nelle immagini 4.13 e 4.14.

Una motivazione possibile di questo comportamento è da ricercarsi nella ridotta dimensione del training set estratto dal primo split (composto da centocinquantacinque immagini). Infatti la probabilità che uno split, effettuato nella fase di cross validation, possa essere poco favorevole al training e successiva validazione è più elevata rispetto all'utilizzo di dataset più ampi.

Bisogna però considerare che l'obiettivo principale dell'utilizzo della cross validation è proprio quello di non vincolarsi all'utilizzo di un solo dataset di validation, che potrebbe portare a risultati particolarmente favorevoli o viceversa, per cui è usuale avere un'ampia variabilità dei risultati utilizzando questo metodo. Inoltre l'influenza di un errore in termini di accuratezza è rilevante, siccome il dataset di validazione è di dimensione esigua, perciò è più probabile che i valori si discostino maggiormente fra le diverse iterazioni della cross validation per epoca.

<i>Modello</i>	<i>Learning rate</i>	<i>Weight decay</i>	<i>Step size</i>	<i>Numero di epoche</i>
AlexNet	0,001	0,00001	35	31
AlexNet (5-fold CV)	0,0001	0,001	35	32
GoogLeNet	0,001	0	20	26
VGG11	0,0001	0,0001	35	28

Tabella 4.1. Valori degli iperparametri migliori scelti.

4.5.2 Testing

Utilizzando i valori degli iperparametri selezionati per ogni modello, si è effettuata una fase finale di testing utilizzando il training set nella sua interezza per l'allenamento e il testing set per calcolare l'efficacia finale dei modelli.

L'accuratezza finale raggiunta dai modelli è stata del 94,87% per AlexNet e VGG, mentre GoogLeNet e AlexNet con gli iperparametri scelti tramite cross validation hanno raggiunto un valore pari a 89,74%. Nella tabella 4.2 sono stati elencati anche i valori per le metriche della matrice di confusione.

Il testing ha mostrato come i modelli di rete neurale convoluzionale siano in grado di classificare correttamente le immagini di questo particolare dataset, nonostante fosse di dimensione esigua. Per questo motivo l'utilizzo di metodi pre-allenati è stato un fattore fondamentale per il raggiungimento di questo obiettivo.

Il modello GoogLeNet ha avuto una performance leggermente più bassa rispetto agli altri due modelli probabilmente a causa di una scelta di valori di learning rate non propriamente adatti al suo allenamento. Infatti, da come si evince anche dalle heatmap 4.15, i valori con maggiore accuratezza di validazione si sono tutti assestati intorno al valore 0.001 di learning rate, mentre per valori più piccoli la rete non è stata spesso in grado di imparare nulla. Probabilmente, effettuando una serie di nuovi training utilizzando valori maggiori di learning rate, si potrebbe raggiungere un valore migliore di accuratezza finale.

Invece il modello di AlexNet allenato con gli iperparametri scelti tramite l'utilizzo della cross validation ha avuto un'accuratezza finale di testing minore di quella del medesimo modello allenato con gli iperparametri scelti con un singolo split nella fase di grid search. Nonostante il valore

dell'accuratezza di testing sia in linea con quello della validazione calcolato durante la fase di grid search, probabilmente gli split effettuati durante la fase di cross validation hanno influenzato negativamente il risultato finale, non privilegiando correttamente le combinazioni di iperparametri che effettivamente avrebbero portato a risultati migliori.

Inoltre il meccanismo di media pesata per ogni epoca porta a non poter prendere in considerazione alcuni fattori secondari, come ad esempio l'accuratezza di training a quella specifica epoca. Questo può portare a scegliere alcuni risultati migliori da inserire nelle heatmap di analisi della grid search senza considerare se il modello sia sottoposto al fenomeno dell'overfit o dell'underfit. Per cui potrebbe essere utile, nel caso si volesse applicare la grid search nuovamente per rendere le analisi più robuste a problemi di split errati dei dataset, utilizzare un calcolo differente della media per ogni iterazione e per ogni epoca della cross validation, che tenga in considerazione anche i valori di metriche secondarie oltre all'accuratezza della validazione.

<i>Modello</i>	<i>Accuratezza</i>	<i>TP</i>	<i>TN</i>	<i>FP</i>	<i>FN</i>
AlexNet	94,87%	7	30	0	2
AlexNet (5-fold CV)	89,74%	5	30	0	4
GoogLeNet	89,74%	5	30	0	4
VGG11	94,87%	7	30	0	2

Tabella 4.2. Valori delle metriche di testing dei modelli.



Figura 4.3. Struttura di GoogLeNet (input in basso).

type	patch size/ stride	output size	depth
convolution	$7 \times 7 / 2$	$112 \times 112 \times 64$	1
max pool	$3 \times 3 / 2$	$56 \times 56 \times 64$	0
convolution	$3 \times 3 / 1$	$56 \times 56 \times 192$	2
max pool	$3 \times 3 / 2$	$28 \times 28 \times 192$	0
inception (3a)		$28 \times 28 \times 256$	2
inception (3b)		$28 \times 28 \times 480$	2
max pool	$3 \times 3 / 2$	$14 \times 14 \times 480$	0
inception (4a)		$14 \times 14 \times 512$	2
inception (4b)		$14 \times 14 \times 512$	2
inception (4c)		$14 \times 14 \times 512$	2
inception (4d)		$14 \times 14 \times 528$	2
inception (4e)		$14 \times 14 \times 832$	2
max pool	$3 \times 3 / 2$	$7 \times 7 \times 832$	0
inception (5a)		$7 \times 7 \times 832$	2
inception (5b)		$7 \times 7 \times 1024$	2
avg pool	$7 \times 7 / 1$	$1 \times 1 \times 1024$	0
dropout (40%)		$1 \times 1 \times 1024$	0
linear		$1 \times 1 \times 1000$	1
softmax		$1 \times 1 \times 1000$	0

Figura 4.4. Dettaglio dei livelli di GoogLeNet.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figura 4.5. Dettaglio dei livelli e delle configurazioni possibili di VGG.

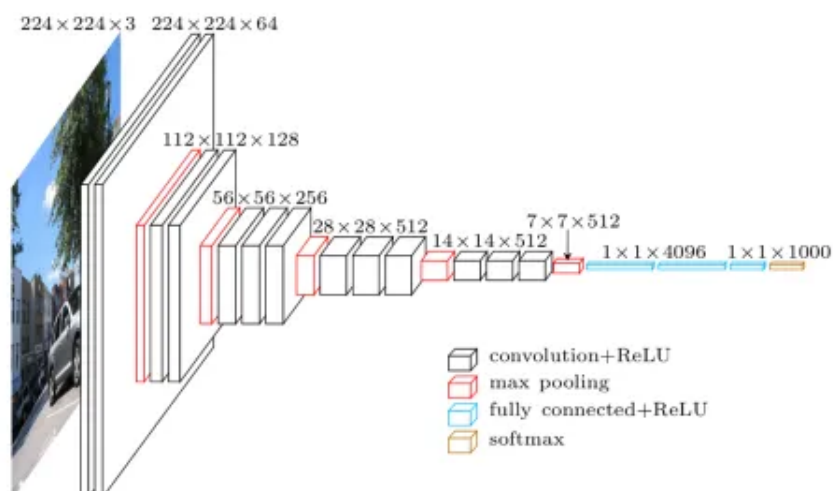


Figura 4.6. Variazione di un immagine d'input attraverso VGG11.

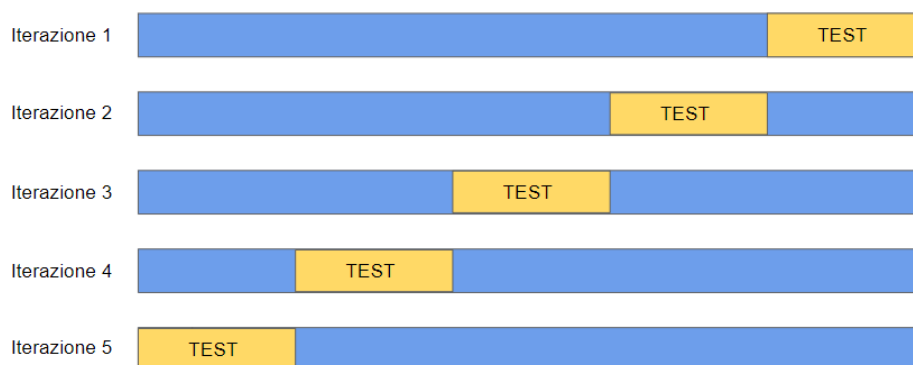
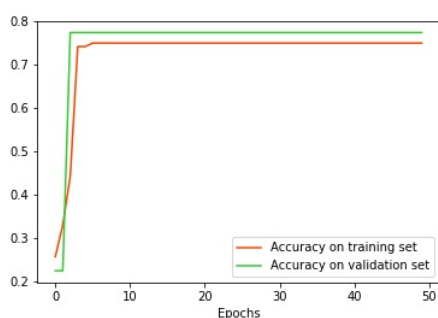
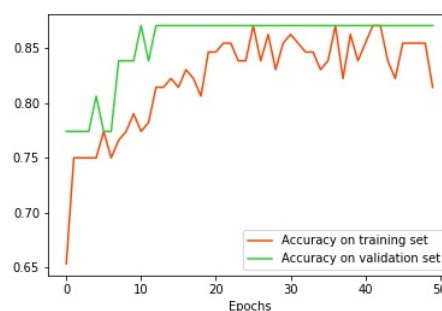


Figura 4.7. Rappresentazione grafica della 5-fold cross validation per ogni iterazione

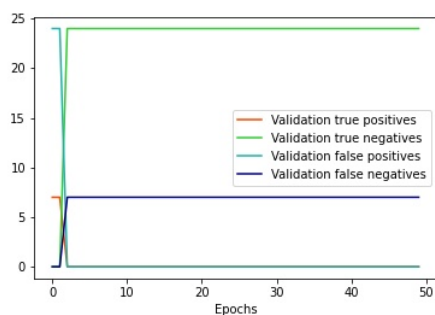


(a) Modello non pre-allenato

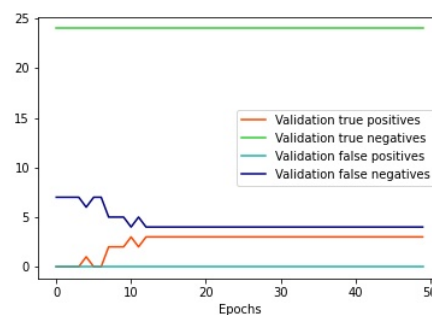


(b) Modello pre-allenato

Figura 4.8. Grafici della variazione dell'accuratezza ($lr = 0,00001$, $weight\ decay = 0$, $step\ size = 15$)



(a) Modello non pre-allenato



(b) Modello pre-allenato

Figura 4.9. Grafici della variazione dei valori della matrice di confusione ($lr = 0,00001$, $weight\ decay = 0$, $step\ size = 15$)

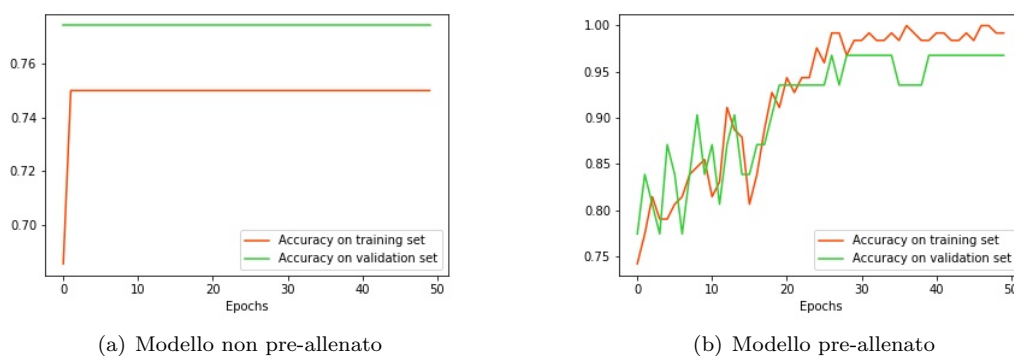


Figura 4.10. Grafici della variazione dell'accuratezza (lr = 0,001, weight decay = 0,0001, step size = 15)

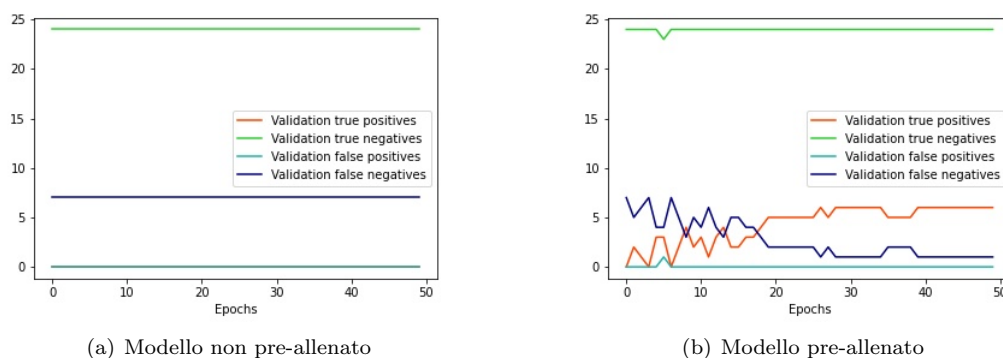


Figura 4.11. Grafici della variazione dei valori della matrice di confusione (lr = 0,001, weight decay = 0,0001, step size = 15)

		Predicted class		
		Positive	Negative	
Actual class	Positive	True positive (TP)	False Negative (FN)	Sensitivity $\frac{TP}{TP+TN}$
	Negative	False positive (FP)	True Negative (TN)	Specificity $\frac{TN}{TN+FP}$
		Precision $\frac{TP}{TP+FN}$	Negative Predictive Value $\frac{TN}{TN+FN}$	Accuracy $\frac{TP+TN}{TP+TN+FP+FN}$

Figura 4.12. Metriche della matrice di confusione per la classificazione binaria

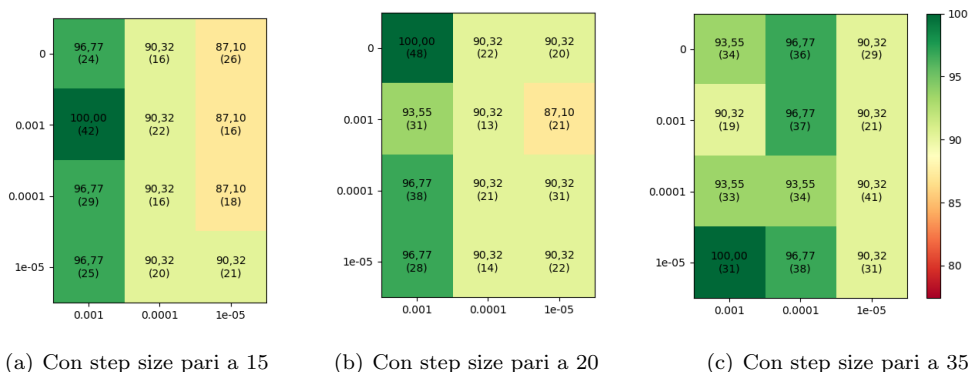


Figura 4.13. Valori di accuratezza del validation set nella grid search per AlexNet

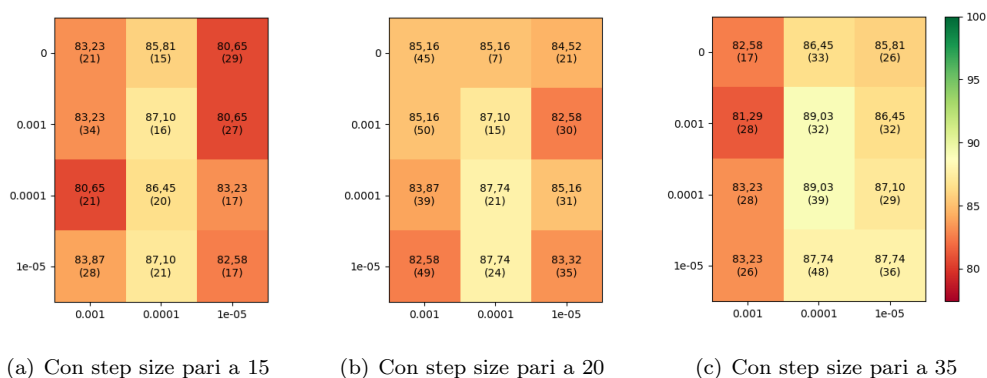


Figura 4.14. Valori di accuratezza del validation set nella grid search per AlexNet con l'utilizzo della 5-fold cross validation

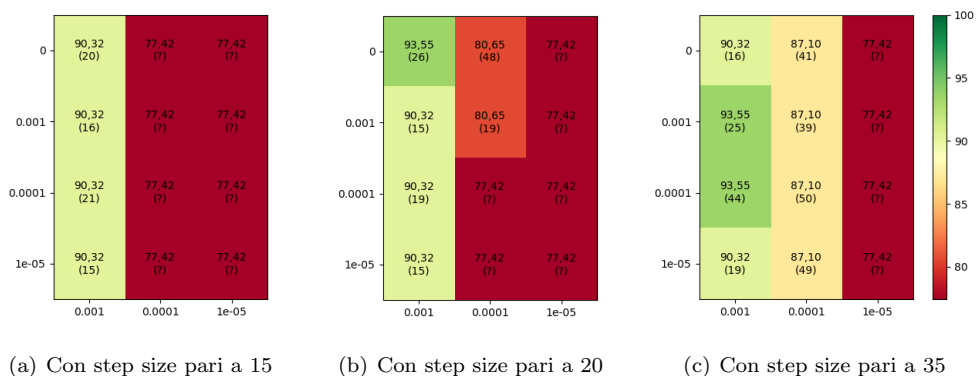


Figura 4.15. Valori di accuratezza del validation set nella grid search per GoogLeNet

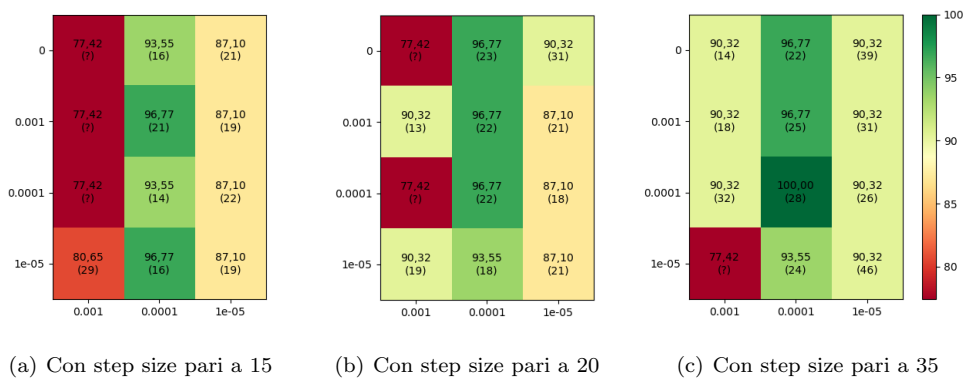


Figura 4.16. Valori di accuratezza del validation set nella grid search per VGG

Capitolo 5

Conclusioni

Il caso di studio presentato all'interno di questo documento ha mostrato come nell'ambito agricolo si possa attualmente immaginare e progettare una soluzione automatica che permetta il supporto nelle scelte che in precedenza sono sempre state effettuate manualmente, con un rischio maggiorato di errori anche nel caso si avesse avuto a disposizione il parere di persone esperte nel settore. Invece questo tipo di soluzioni, coadiuvate dagli algoritmi e dalle tecniche dell'intelligenza artificiale, permettono di poter condividere la conoscenza e il parere di un tecnico senza il suo supporto attivo e presente, ma immergendo il suo giudizio e le proprie competenze all'interno di una piattaforma accessibile a tutti. Infatti gli algoritmi allenati e risultanti dalla fase di analisi delle immagini possono essere tranquillamente integrati all'interno di una piattaforma disponibile online e consultabile da chiunque necessiti di un parere tecnico per poter effettuare alcune scelte puntuali sulle proprie coltivazioni.

Ovviamente bisogna considerare che nel percorso del lavoro di tesi descritto nei capitoli di questo elaborato, è possibile notare che il traguardo raggiunto abbia definito un processo di lavoro utile per uno sviluppo e un miglioramento futuri. Infatti, nonostante i risultati siano stati promettenti, raggiungendo valori finali di accuratezza nella classificazione delle immagini vicini al 95% per due dei tre modelli di rete neurale convoluzionale selezionati, si è potuto mostrare come il margine di miglioramento sia ancora ampio e che vi possano essere diverse vie da percorrere per continuare la strada delineata. Qui di seguito vengono elencati alcuni concetti che riguardano diversi possibili spunti di continuazione per migliorare ciò che di positivo è già stato raggiunto:

- Utilizzo di modelli di rete neurale differenti, magari anche definiti in modo specifico per lo scopo da zero, oppure modificando modelli già esistenti;
- Aumento dei dati di allenamento e testing, introducendo immagini collezionate anche su altri campi agricoli, al fine di aumentare la diversità delle caratteristiche e poter generalizzare meglio il problema di classificazione;
- L'introduzione di nuove classi in cui suddividere il dataset, attraverso analisi più approfondite degli esperti del settore, magari individuando anche la tipologia di stress o, persino, la malattia specifica della pianta;
- Utilizzo di tecniche di ribilanciamento del dataset e di aumento dei dati prima del processamento, al fine di migliorare la capacità di apprendimento dei modelli riguardo gli elementi delle classi meno rappresentate nel dataset;
- Ricerca e utilizzo di nuove tecniche di propagazione dell'errore all'interno degli algoritmi d'intelligenza artificiale;
- Utilizzo di piattaforme hardware più potenti, al fine di poter effettuare prove ed esperimenti che richiedono più risorse di calcolo;
- Aumento dei valori degli iperparametri ricercati con la tecnica della ricerca su griglia, al fine di trovare combinazioni ancora più performanti;

- Utilizzo della cross validation per tutti i modelli, applicando metriche composte per trovare la combinazione migliore d'iperparametri, tenendo in considerazione la possibilità di effettuare una ricerca del miglior valore di k determinante la k-fold cross validation.

Il fatto che le vie di sviluppo siano molte aumenta il numero delle possibilità di raggiungere nuovi obiettivi e nuove frontiere, generando strumenti di supporto via a via più utili e utilizzabili in un contesto reale di agricoltura di precisione.

Bibliografia

- [1] Xia Hao, Jingdun Jia, Abdul Mateen Khattak, Li Zhang, Xuchao Guo, Wanlin Gao, Minjuan Wang, “*Growing period classification of *Gynura bicolor* DC using GL-CNN*”, Computers and Electronics in Agriculture, Vol. 174, No. 105497, Luglio 2020, DOI [10.1016/j.compag.2020.105497](https://doi.org/10.1016/j.compag.2020.105497) (<https://www.sciencedirect.com/science/article/pii/S0168169919321064>)
- [2] Koushik Nagasubramanian, Sarah Jones, Asheesh K. Singh, Soumik Sarkar, Arti Singh, Baskar Ganapathysubramanian, “*Plant disease identification using explainable 3D deep learning on hyperspectral images*”, Plant Methods, Vol. 15, No. 98, Agosto 2019, DOI [10.1073/pnas.1716999115](https://doi.org/10.1073/pnas.1716999115) (<https://link.springer.com/article/10.1186/s13007-019-0479-8>)
- [3] Sambuddha Ghosal, David Blystone, Asheesh K. Singh, Baskar Ganapathysubramanian, Arti Singh, Soumik Sarkar “*An explainable deep machine vision framework for plant stress phenotyping*”, PNAS, Vol. 115, No. 18, Maggio 2018, pp. 4613-4618 DOI [10.1073/pnas.1716999115](https://doi.org/10.1073/pnas.1716999115) (<https://www.pnas.org/content/115/18/4613>)
- [4] Yu Sun, Yuan Liu, Guan Wang, Haiyan Zhang, “*Deep Learning for Plant Identification in Natural Environment*”, Computational Intelligence and Neuroscience, Vol. 2017, Id. 7361042, Maggio 2017, DOI [10.1155/2017/7361042](https://doi.org/10.1155/2017/7361042) (<https://www.hindawi.com/journals/cin/2017/7361042/>)
- [5] Aydin Kaya, Ali Seydi Keceli, Cagatay Catal, Hamdi Yalin Yalic, Huseyin Temucin, Bedir Tekinerdogan, “*Analysis of transfer learning for deep neural network based plant classification models*”, Computers and Electronics in Agriculture, Vol. 158, Marzo 2019, pp. 20-29 DOI [10.1016/j.compag.2019.01.041](https://doi.org/10.1016/j.compag.2019.01.041) (<https://www.sciencedirect.com/science/article/pii/S0168169918315308>)
- [6] Jayme Garcia Arnal Barbedo, “*Plant disease identification from individual lesions and spots using deep learning*”, Biosystems Engineering, Vol. 180, Aprile 2019, pp. 96-107 DOI [10.1016/j.biosystemseng.2019.02.002](https://doi.org/10.1016/j.biosystemseng.2019.02.002) (<https://www.sciencedirect.com/science/article/pii/S1537511018307797>)
- [7] Konstantinos P. Ferentinos, “*Deep learning models for plant disease detection and diagnosis*”, Computers and Electronics in Agriculture, Vol. 145, Febbraio 2018, pp. 311-318 DOI [10.1016/j.compag.2018.01.009](https://doi.org/10.1016/j.compag.2018.01.009) (<https://www.sciencedirect.com/science/article/pii/S0168169917311742>)
- [8] S.M. Hamylton, R.H. Morris, R.C. Carvalho, N. Roder, P. Barlow, K. Mills, L. Wang, “*Evaluating techniques for mapping island vegetation from unmanned aerial vehicle (UAV) images: Pixel classification, visual interpretation and machine learning approaches*”, International Journal of Applied Earth Observation and Geoinformation, Vol. 89, No. 102085, Luglio 2020, DOI [10.1016/j.jag.2020.102085](https://doi.org/10.1016/j.jag.2020.102085) (<https://www.sciencedirect.com/science/article/pii/S0303243419310293>)
- [9] João Valente, Bilal Sari, Lammert Kooistra, Henk Kramer, Sander Múcher, “*Automated crop plant counting from very high-resolution aerial imagery*”, Precision Agric, Vol. 21, Maggio 2020, pp. 1366-1384, DOI [10.1007/s11119-020-09725-3](https://doi.org/10.1007/s11119-020-09725-3) (<https://link.springer.com/article/10.1007/s11119-020-09725-3>)
- [10] Benyamin Hosseiny, Heidar Rastiveis, Saeid Homayouni, “*An Automated Framework for Plant Detection Based on Deep Simulated Learning from Drone Imagery*”, Remote Sensing, Vol. 12, No. 21, Ottobre 2020, DOI [10.3390/rs12213521](https://doi.org/10.3390/rs12213521) (<https://www.mdpi.com/2072-4292/12/21/3521>)

- [11] Wanqiang Qian, Yiqi Huang, Qi Liu, Wei Fan, Zhongyu Sun, Hui Dong, Fanghao Wan, Xi Qiao, “*UAV and a deep convolutional neural network for monitoring invasive alien plants in the wild*”, *Computers and Electronics in Agriculture*, Vol. 174, No. 105519, Luglio 2020, DOI [10.1016/j.compag.2020.105519](https://doi.org/10.1016/j.compag.2020.105519) (<https://www.sciencedirect.com/science/article/pii/S0168169920302921>)
- [12] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, “*ImageNet Classification with Deep Convolutional Neural Networks*”, *Communications of the ACM*, Vol. 60, No. 6, Maggio 2017, pp. 84-90, DOI [10.1145/3065386](https://doi.org/10.1145/3065386)
- [13] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, “*Going deeper with convolutions*”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Giugno 2015, pp. 1-9, DOI [10.1109/CVPR.2015.7298594](https://doi.org/10.1109/CVPR.2015.7298594) (<https://ieeexplore.ieee.org/document/7298594>)
- [14] Karen Simonyan, Andrew Zisserman “*Very Deep Convolutional Networks for Large-Scale Image Recognition*”, *International Conference on Learning Representations*, 2015, (<https://arxiv.org/abs/1409.1556>)