

UNIVERSITÀ DEGLI STUDI DI SALERNO

---

Tesi di Laurea Magistrale

## Wild animal recognition using an Edge Device



**Relatrice:**

Prof.ssa Tortora Genoveffa

**Correlatori:**

Prof.ssa Sebillo Monica

Dott. Battistoni Pietro

**Laureando:**

Russo Vincenzo

Matr: 0522500860

---

Anno Accademico 2021/2022

# Ringraziamenti

Dedico questo lavoro di tesi alla persona che fin dall'inizio più di tutti ha creduto in me, supportandomi in tutti i miei momenti di debolezza, ma anche e soprattutto nei momenti di gioia. Se sono arrivato a fine di questo percorso è in parte grazie a lei e spero che un po' sarà orgogliosa di me. A Carmen.

Il primo ringraziamento va sicuramente ai relatori di questa tesi, i professori Genoveffa Tortora e Monica Sebillo, che mi hanno seguito con disponibilità e precisione nello sviluppo e nella stesura della tesi.

Il secondo ringraziamento va al dottore Pietro Battistoni che mi è stato d'aiuto durante tutto il periodo di progettazione, aiutandomi nelle difficoltà senza mai lasciarmi da solo.

Ringrazio tutti i miei amici e tutte le persone che ho incontrato in questo percorso dove ognuno nella loro particolarità ha contribuito a rendermi la persona che sono oggi. Un grazie particolare va ai miei genitori che mi hanno permesso di poter arrivare alla fine di questo percorso e a tutta la mia famiglia.

E a me stesso dico:

L'informatica non riguarda i computer più di quanto l'astronomia riguardi i telescopi.

*Edsger Wybe Dijkstra*

# Abstract

Nell'ultimo decennio è aumentata la richiesta di dispositivi intelligenti e per soddisfare tale richiesta è cresciuta in maniera esponenziale la ricerca e lo sviluppo nell'ambito dell'intelligenza artificiale. In questo contesto, le reti neurali e il Deep learning stanno subendo uno sviluppo fortissimo, in quanto con le nuove tecnologie e centri di calcolo sempre più performanti, si riesce ad effettuare operazioni che fino a pochi anni fa risultavano impossibili a causa di tempi di elaborazione troppo elevati.

Nel caso specifico, analizzeremo il problema dell'image recognition sfruttando tecniche di deep learning. L'image recognition consiste nell'individuare oggetti, animali, persone, trame specifiche, dette pattern, presenti in un'immagine o in un video. Una volta individuato ciò che serve, possiamo poi attivare un sistema o un servizio legato a ciò che stiamo osservando, come per esempio l'arresto immediato di un veicolo in caso di pericolo imminente.

In questa tesi inoltre si affronterà brevemente la possibilità di sfruttare una configurazione Edge/Cloud che permette di ottenere velocità nell'addestramento delle reti neurali, capacità di operare real-time, elasticità, resistenza dell'intero sistema in caso di guasti. Tale configurazione permette di avere costi ridotti grazie all'utilizzo di strumenti meno costosi nella regione di sensing e nella regione di Edge. È stato implementato un sistema di edge computing per il riconoscimento di animali selvatici come capre e cinghiali.

# Introduzione

## Il problema affrontato

L'Italia possiede una consistente popolazione di animali selvatici che vagano liberi nelle aree rurali, in particolare cinghiali e capre. Sebbene abitino prevalentemente in aree boschive, si aggirano nei villaggi e sulle strade e il loro numero è in aumento. Quando si guida attraverso tali aree note per essere densamente popolate da animali selvatici, c'è sempre la possibilità dell'arrivo improvviso e inaspettato di un animale che attraversa la strada sul percorso di un'auto, dando al conducente poco o nessun tempo per fermarsi. I turisti in vacanza in Italia sono spesso completamente all'oscuro della possibilità che si verifichi una situazione del genere che sta diventando sempre più comune, negli ultimi cinque anni la Coldiretti, un'associazione agricola nazionale, riferisce che la sola Lombardia ha registrato 400 incidenti di questo tipo dovuti ad animali selvatici [5].

Quello che è ancora meno noto è che esiste la possibilità di chiedere il risarcimento dei danni per eventuali lesioni subite in tale incidente alla Regione in cui esso è avvenuto; addizionalmente, è possibile fare una richiesta di risarcimento fino a due anni dopo il verificarsi dell'incidente. Nelle zone in cui gli animali vaganti sono comuni c'è una responsabilità dovuta agli utenti della strada da parte dei proprietari di una strada per garantire, per quanto possibile, che gli animali selvatici siano contenuti da recinzioni, l'installazione di reti, fossati e guard-rail ai bordi delle strade e i proprietari devono mantenere le recinzioni ecc. e riparare eventuali danni che permetterebbero agli animali selvatici di accedere alla strada e anche per la segnaletica che avverte un automobilista della possibilità di animali selvatici vaganti. In Lombardia, nel gennaio di quest'anno, un gruppo di cinghiali ha scavato un buco sotto una recinzione di un'autostrada che ha permesso loro di accedere alla strada a scorrimento veloce, causando un tamponamento di tre auto e causando la morte di tre automobilisti e gravi ferite alle altre persone coinvolte. I cinghiali sono creature potenti e usano il loro muso per scavare e cercare il cibo.

Gli incidenti che coinvolgono gli animali selvatici sono così diffusi che è stato riportato dalla stampa questa settimana che due automobilisti a Sorso, in Sardegna, hanno inscenato un incidente usando il cadavere di un cinghiale nel tentativo di ottenere un risarcimento. Tuttavia, si è scoperto che il cinghiale era stato congelato e l'intero incidente è stato escogitato per estrarre denaro dalle autorità.

---

Se si verifica un vero incidente, spesso c'è una disputa tra la Regione e lo Stato su dove sia la responsabilità e su chi debba risarcire le parti lese [4]. Succede spesso che le due Autorità si impegnino in una lunga disputa legale su chi ha la maggiore responsabilità di pagare, questo succede soprattutto perché ci sono diverse leggi relative a questo problema [6].

## Sistemi proposti in letteratura

In letteratura [10] esistono due tipi di sistemi principali per il rilevamento degli animali selvatici:

1. ***Area-cover systems***: sono progettati per rilevare gli animali entro una certa area e portata da un sensore. Quest'area è tipicamente a forma di cono, stretta vicino al sensore e più ampia man mano che la distanza dal sensore aumenta. Essi rilevano gli animali in base al calore corporeo e al movimento.
2. ***Break-the-beam systems***: consistono di un trasmettitore che trasmette un segnale a un ricevitore. I sistemi Break-the-beam rilevano gli animali quando il loro corpo blocca il segnale o quando il segnale ricevuto dal ricevitore è molto ridotto. I sistemi break-the-beam utilizzano segnali radio a infrarossi, laser o microonde. L'area di rilevamento è l'area all'interno della quale i sistemi a copertura d'area dovrebbero rilevare gli animali grandi, e la linea di rilevamento è la linea tra i sensori dove i sistemi break-the-beam dovrebbero rilevare i grandi animali.

## Wildlife Detection System

Ogni anno, migliaia di collisioni con la fauna selvatica sono riportate sulle autostrade della British Columbia (BC). Wildlife-Vehicle Collisions (WVC) spesso causano gravi lesioni personali, mettono in pericolo varie specie di animali selvatici e causano perdite di milioni di dollari all'anno. Le statistiche della Insurance Corporation of BC (ICBC) mostrano che tra il 2011 e il 2015 una media annuale di 10.000 incidenti tra animali e veicoli in BC. Questi incidenti hanno provocato, in media, 570 lesioni personali e tre incidenti mortali all'anno. Oltre alle significative perdite personali derivanti dai WVC.

Il sistema [43] consiste in una suite di componenti Sistemi di Trasporto Intelligenti (ITS) integrati con tecnologie di sicurezza ad alte prestazioni. I dispositivi richiesti per la funzionalità di base del sistema includono Dynamic Message Signs (DMS), telecamere termiche e a colori, e vari dispositivi di comunicazione. Il sistema è destinato principalmente a rilevare la presenza di grandi animali, vale a dire cervi, alci e pecore utilizzando tecnologie sviluppate principalmente per l'industria della sicurezza. Il minimo obiettivo rilevabile è all'incirca la dimensione di un cerbiatto, anche se sono stati rilevati animali più piccoli. Quando un grande animale viene rilevato, il sistema di controllo attiva una serie di segnali di avvertimento che avvisano gli automobilisti della presenza di animali per aiutare a garantire la sicurezza degli animali e dei viaggiatori. Quando l'animale selvatico abbandona

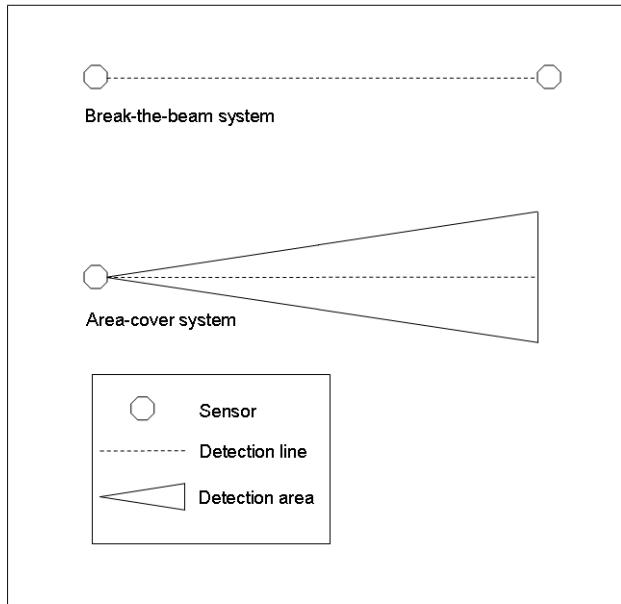


Figura 1: Rappresentazione schematica dei due tipi di sistemi.

la zona di rilevamento, il WDS disattiva i segnali di avvertimento. Inoltre, il video termico e ad alta definizione degli animali selvatici rilevati viene registrato e memorizzato per analisi e revisioni future.

Piuttosto che recintare l'intera area gli esperti hanno ritenuto più economico installare questo sistema tenendo in conto i seguenti fattori:

- **Fattore meteorologico:** in ogni singolo WDS implementato in precedenza, senza successo, l'incapacità dei sensori di operare in varie condizioni atmosferiche è stata elencata come il fattore decisivo prominente per l'interruzione del sistema. I dispositivi resistenti alle intemperie e certificati.

**Ambiente reale:** rilevare la fauna selvatica in un banco di prova risulta più semplice dal punto di vista tecnico. Rilevare la fauna selvatica in modo affidabile per decenni in un ambiente reale non lo è. È importante progettare un sistema che avrebbe dimostrato di essere affidabile per gli anni a venire, senza dover essere continuamente monitorato, testato e ricalibrato per mantenere il livello richiesto di capacità di rilevamento.

- **Falsi positivi e valore del sistema:** un alto tasso di falsi positivi è una delle lamentele più comuni tra gli utenti; molti hanno dichiarato di aver iniziato a ignorare completamente il sistema a causa dei falsi positivi. UN WDS può rilevare tutta la fauna selvatica che passa, ma se ha un alto tasso di falsi positivi, è essenzialmente un segnale stradale troppo costoso. Il numero di falsi positivi per questo WDS è

---

stato monitorato su un ampio System Acceptance Testing (SAT), inoltre è stato ottimizzato per ridurre il più possibile i tassi di falso allarme.

- **La convalida dei dati:** i sistemi implementati in precedenza generalmente non avevano l'infrastruttura necessaria per permettere una rigorosa convalida dei dati per determinare l'accuratezza di rilevamento effettivamente raggiunta.

I sistemi precedenti hanno fallito anche a causa di tecnologie poco sofisticate e non adatte all'ambiente da monitorare. La tecnologia utilizzata per questo sistema è stata accuratamente selezionata in base al terreno, clima ed i requisiti funzionali del Ministero Canadese. Il sistema deve tenere in conto di:

- Animali multipli.
- Animali che pascolano ai lati della strada.
- Auto, camion, moto, ciclisti e pedoni.
- Maltempo.

Il sistema possiede anche i seguenti parametri per la messa a punto come la quantità massima di tempo in cui il sistema deve rimanere attivato, dimensione del bersaglio e velocità minima/massima del bersaglio.

I criteri di performance di un sensore sono:

- **Falsi positivi:** eventi non validi che innescano il funzionamento del sistema. Potrebbero essere causati da fonti al di fuori della soglia di rilevamento desiderata (es. il sistema rileva un coyote quando dovrebbe rilevare solo cervi di grandi dimensioni e oltre), tempo, temperatura, connessioni difettose, malfunzionamento delle apparecchiature o danni, ecc.
- **Falsi negativi:** Eventi che avrebbero dovuto innescare il funzionamento del sistema, ma non l'hanno fatto. Questo potrebbe essere causato dal mancato rilevamento di una fonte valida.
- **False Alarm Rate (FAR):** Il tasso totale di eventi non validi (falsi positivi) rispetto alla quantità totale di eventi.
- **Detection Rate (DR):** La quantità totale di eventi di rilevamento rispetto alla quantità totale di falsi eventi negativi.
- **Tempo di inattività:** La quantità di tempo in cui il sistema non è disponibile per il funzionamento normale.

Le metriche raccolte dal WDS ,oltre a quelle raccolte dai sensori di rilevamento, per scopi di testing, validazione e valutazione delle performance sono le seguenti:

- 
- **Tassi di incidenti WVC:** Le informazioni su questi tassi sono regolarmente raccolte dal Ministero.
  - **Traffico:** Informazioni sulla quantità di traffico veicolare che passa attraverso il corridoio.
  - **Velocità dei veicoli attraverso il corridoio:** Queste informazioni aiuteranno a determinare come i conducenti stanno rispondono al sistema quando i segnali sono attivati.
  - **Tassi di attivazione:** Informazioni sul numero di rilevamenti di animali selvatici lungo il corridoio in un dato periodo di tempo.



Figura 2: Sistema WDS attivato dal passaggio di animali.

---

Il lavoro di tesi è strutturato come di seguito:

- **Primo capitolo:** descrizione dettagliata del dataset Missouri Camera Traps [23].
- **Secondo capitolo:** approfondimento delle tecnologie utilizzate per poter addestrare i modelli lato software e per il deployment lato hardware.
- **Terzo capitolo:** definizione dei processi di addestramento dei modelli sul cloud [9].
- **Quarto capitolo:** descrizione degli script eseguiti sull'edge device.
- **Quinto capitolo:** discussione dei risultati raggiunti, quindi delineazione delle conclusioni e sviluppi futuri.

# Capitolo 1

## Descrizione del dataset

### 1.1 Labeled Information Library of Alexandria

Il dataset è stato fornito da Labeled Information Library of Alexandria: Biology and Conservation (LILA)[18], essa contiene oltre 10 milioni di immagini etichettate inerenti alla biologia, flora e fauna. Le immagini provengono in maniera particolare da trappole fotografiche.

LILA BC è un repository per set di dati relativi alla biologia e alla conservazione, inteso come una risorsa sia per i ricercatori di machine learning (ML) che per quelli che vogliono sfruttare il ML per la biologia e la conservazione.

L'apprendimento automatico dipende da dati etichettati, ma l'accesso a tali dati in biologia è una sfida. Di conseguenza, tutti beneficiano quando i dati etichettati sono resi disponibili. I biologi e gli scienziati della conservazione traggono beneficio dall'avere dati su cui addestrare i loro modelli, e l'hosting gratuito permette ai team di moltiplicare l'impatto dei loro dati (suggeriamo di elencare questo beneficio nelle proposte di sovvenzione che finanziano la raccolta dei dati). I ricercatori di ML traggono vantaggio dall'avere dati da sperimentare.

LILA BC è destinato ad ospitare dati di varia natura, ma l'enfasi è posta sulle immagini etichettate; attualmente ospitiamo oltre dieci milioni di immagini etichettate.

### 1.2 Missouri Camera Traps

Il dataset Missouri Camera Traps [23] contiene circa 25.000 immagini di trappole fotografiche che rappresentano 20 specie (per esempio, le etichette più comuni sono cervo rosso, muflone e cervo dalla coda bianca). Le immagini all'interno di ogni sequenza condividono la stessa etichetta di specie (anche se l'animale potrebbe non essere stato registrato in tutte le immagini della sequenza). Sono inclusi circa 900 bounding boxes. Si tratta di sequenze molto impegnative con scene altamente disordinate e dinamiche. Le risoluzioni spaziali delle immagini variano da  $1920 \times 1080$  a  $2048 \times 1536$ . La lunghezza delle sequenze varia da 3 a più di 300 fotogrammi.

Il dataset compresso ha una dimensione su disco di 9.41 GB e le immagini in esso contenute sono suddivise nelle seguenti cartelle contenenti immagini dei seguenti animali selvatici:

- Agouti
- Collared Peccary
- Paca
- Red Brocket Deer
- White-nosed Coati
- Spiny Rat
- Ocelot
- Red Squirrel
- Common Opossum
- Bird spec
- Great Tinamou
- White Tailed Deer
- Mouflon
- Red Deer
- Roe Deer
- Wild Boar
- Red Fox
- European Hare
- Wood Mouse
- Coiban Agouti

Nel nostro studio abbiamo utilizzato soltanto due delle seguenti classi, il cinghiale selvatico (Wild Boar) e la capra selvatica (Mouflon). La cartella Wild Boar contiene 100 sequenze, contenenti in totale 1883 immagini, mentre per Mouflon sono presenti 100 sequenze e 2365 immagini. Nel dataset originale le immagini appartenenti alle classi interessate non risultavano etichettate, per questo è stato svolto un lavoro di labeling manuale con il tool Roboflow[35]. Tramite Roboflow è stato applicato uno split 80-20-10 rispettivamente per training, test e validation set. Il dataset è stato esportato nel formato YOLOv5 richiesto dalle librerie in Python, contiene le immagini come file jpeg e le etichette come file di testo.

Il formato COCO ripartisce le immagini in 3 cartelle, '*train*', '*test*', '*valid*', ed in ciascuna di esse è presente un file di testo in formato JSON per definire gli oggetti e le etichette. Il file JSON è strutturato in 5 sezioni:

- 'info': contenente informazioni sulla creazione del dataset;

```
"info": {  
    "year": "2022",  
    "version": "1",  
    "description": "Exported from roboflow.ai",  
    "contributor": "",  
    "url": "https://public.roboflow.ai/object-detection/undefined",  
    "date_created": "2022-03-18T10:25:37+00:00"  
}
```

- 'licenses': contenente informazioni sulla licenza del dataset;

```
"licenses": {  
    {  
        "id": 1,  
        "url": "https://creativecommons.org/licenses/by/4.0/",  
        "name": "CC BY 4.0"  
    }  
}
```

- 'categories': indica l'id e il nome delle categorie degli oggetti identificati nelle immagini;

```
"categories": [  
    {  
        "id": 0,  
        "name": "goat-boars",  
        "supercategory": "none"  
    },  
    {  
        "id": 1,  
        "name": "boar",  
        "supercategory": "goat-boars"  
    },  
    {  
        "id": 2,  
        "name": "goat",  
        "supercategory": "goat-boars"  
    }  
]
```

- 'images': contenente per ciascuna immagine il nome, path, id e metadati su di essa.

```
"images": [
  {
    "id": 0,
    "license": 1,
    "file_name": "853934932_75e3c59432.jpg",
    "height": 416,
    "width": 416,
    "date_captured": "2022-03-18T10:25:37+00:00"
  }
]
```

- 'annotations': per ciascun oggetto presente nel dataset viene associato un id, una bounding box, e l'id dell'immagine dove esso si trova. Una bounding box è definita da 4 numeri che rappresentano le coordinate dei 2 angoli opposti della bounding box;

```
"annotations": [
  {
    "id": 0,
    "image_id": 0,
    "category_id": 1,
    "bbox": [
      105,
      48,
      212,
      367.5
    ],
    "area": 77910,
    "segmentation": [],
    "iscrowd": 0
  }
]
```

Il formato YOLO fornisce una suddivisione delle immagini del dataset in 2 cartelle in formato COCO, gli oggetti sono definiti in un file YAML 'data.yaml', e per ogni immagine è presente un file di testo. Il file YAML definisce i path delle cartelle contenenti le immagini di training e validation oltre al numero di classi da identificare e al loro nome. Il file di testo di una singola immagine contiene una riga per ogni oggetto riconosciuto nell'immagine, la riga è composta dall'id della classe a cui appartiene l'oggetto e dai 4 numeri che compongono le coordinate dei 2 angoli opposti dell'immagine.

# Capitolo 2

## Tecnologie utilizzate

In questo capitolo verranno descritte tutte le tecnologie utilizzate per la costruzione della board e il software da essa eseguito.

### 2.1 Software

#### 2.1.1 Linguaggio Python

Python [20] è un linguaggio di programmazione di più "alto livello" rispetto alla maggior parte degli altri linguaggi, orientato a oggetti, adatto, tra gli altri usi, a sviluppare applicazioni distribuite, scripting, computazione numerica e system testing.

Ideato da Guido van Rossum all'inizio degli anni Novanta è un linguaggio multi-paradigma che ha tra i principali obiettivi: dinamicità, semplicità e flessibilità. Supporta il paradigma object oriented, la programmazione strutturata e molte caratteristiche di programmazione funzionale e riflessione.

Le caratteristiche più immediatamente riconoscibili di Python sono le variabili non tipizzate e l'uso dell'indentazione per la sintassi delle specifiche, al posto delle più comuni parentesi.

Altre caratteristiche distintive sono l'overloading di operatori e funzioni tramite delegati, la presenza di un ricco assortimento di tipi e funzioni di base e librerie standard, sintassi avanzate quali slicing e list comprehension.

Il controllo dei tipi è forte (strong typing) e viene eseguito in runtime (dynamic typing): una variabile è un contenitore a cui viene associata un'etichetta (il nome) che può essere associata a diversi contenitori anche di tipo diverso durante il suo tempo di vita. Fa parte di Python un sistema garbage collector per liberazione e recupero automatico della memoria di lavoro.

Python ha qualche somiglianza con Perl, ma i suoi progettisti hanno scelto una sintassi più essenziale e uniforme con l'obiettivo di migliorare la leggibilità del codice. Analogamente a Perl è classificato spesso come linguaggio di scripting, ma pur essendo utile per scrivere

script di sistema, in alternativa per esempio a bash, la grande quantità di librerie disponibili e la facilità con cui il linguaggio permette di scrivere software modulare favoriscono anche lo sviluppo di applicazioni molto complesse.

Python è un linguaggio di programmazione multi-paradigma. La programmazione orientata agli oggetti e la programmazione strutturata sono pienamente supportate, e molte delle sue caratteristiche supportano la programmazione funzionale e la programmazione orientata agli aspetti (inclusa la meta-programmazione e i metaoggetti (metodi magici)). Molti altri paradigmi sono supportati tramite estensioni, inclusa la progettazione per contratto e la programmazione logica.

Python usa la tipizzazione dinamica e una combinazione di conteggio dei riferimenti e un garbage collector a rilevazione dei cicli per la gestione della memoria. Dispone anche della risoluzione dinamica dei nomi (late binding), che lega i nomi dei metodi e delle variabili durante l'esecuzione del programma.

Il design di Python offre un certo supporto per la programmazione funzionale nella tradizione Lisp. Ha funzioni filter, Map-Reduce; list comprehensions, dizionari, set, ed espressioni generatici. La libreria standard ha due moduli (`itertools` e `functools`) che implementano strumenti funzionali presi in prestito da Haskell e Standard ML.

Piuttosto che avere tutte le sue funzionalità incorporate nel suo nucleo, Python è stato progettato per essere altamente estensibile (con moduli). Questa modularità compatta lo ha reso particolarmente popolare come mezzo per aggiungere interfacce programmabili ad applicazioni esistenti. La visione di Van Rossum di un piccolo linguaggio di base con una grande libreria standard e un interprete facilmente estensibile deriva dalle sue frustrazioni con ABC, che sposava l'approccio opposto.

### 2.1.2 L'Ambiente di Sviluppo Integrato PyCharm

PyCharm [3] è un ambiente di sviluppo integrato (IDE) utilizzato nella programmazione di computer, in particolare per il linguaggio di programmazione Python. È sviluppato dalla società ceca JetBrains (precedentemente conosciuta come IntelliJ). Fornisce analisi del codice, un debugger grafico, un unit tester integrato, integrazione con i sistemi di controllo della versione (VCS), e supporta lo sviluppo web con Django così come processi di Data Science con Anaconda.

PyCharm è multipiattaforma, con versioni per Windows, macOS e Linux. L'edizione comunitaria è rilasciata sotto la licenza Apache, inoltre è presente l'edizione professionale con caratteristiche extra - rilasciata sotto una licenza proprietaria finanziata da un abbonamento e anche una versione educativa. Il motivo della sua scelta è la facilità di utilizzo e l'ottima integrazione con l'ecosistema Python.

### 2.1.3 Google Colab

Google Colaboratory [9] (noto anche come Colab) è un ambiente gratuito di Jupyter Notebook che opera su cloud e memorizza i suoi notebook su Google Drive. Colab fu originariamente un progetto interno di Google; La principale caratteristica è che non richiede

una configurazione e i notebook che si creano possono essere modificati simultaneamente dai membri del tuo team. Colab supporta molte popolari librerie di Machine Learning che possono essere facilmente caricate nel vostro notebook. Inoltre fornisce gratuitamente una GPU per una runtime massima di 12 ore. Sono presenti altre versioni del servizio come Pro e Pro+, che permettono di avere tempi di runtime superiori e GPU più veloci. In questa tesi è stata utilizzata la versione gratuita del servizio.

#### 2.1.4 Jupyter Notebook

Jupyter Notebook [16] (precedentemente IPython Notebooks) è un ambiente computazionale interattivo basato sul web per la creazione di documenti notebook.

Un documento Jupyter Notebook è un REPL basato su browser che contiene una lista ordinata di celle di input/output che possono contenere codice, testo (usando Markdown), matematica, grafici. Un notebook è un documento JSON, che segue uno schema di versioni, che di solito termina con l'estensione ".ipynb".

Jupyter Notebook può connettersi a molti kernel per permettere la programmazione in diversi linguaggi. Un kernel Jupyter è un programma responsabile della gestione di vari tipi di richieste (esecuzione di codice, completamento di codice, ispezione), e di fornire una risposta. I kernel parlano con gli altri componenti di Jupyter usando ZeroMQ, e quindi possono essere sulla stessa macchina o su macchine remote. A differenza di molte altre interfacce simili a Notebook, in Jupyter, i kernel non sono consapevoli di essere collegati ad uno specifico documento, e possono essere collegati a molti client contemporaneamente. Di solito i kernel permettono l'esecuzione di un solo linguaggio per default Jupyter Notebook viene fornito con il kernel IPython. Un Jupyter Notebook può essere convertito in un certo numero di formati di output standard aperti (HTML, slide di presentazione, LaTeX, PDF, ReStructuredText, Markdown, Python) attraverso "Scarica come" nell'interfaccia web.

#### 2.1.5 Libreria OpenCV

OpenCV [17] (Open Source Computer Vision Library) è una libreria software open source di Computer Vision e Machine Learning. OpenCV è stata costruita per fornire un'infrastruttura comune per le applicazioni di visione artificiale e per accelerare l'uso della percezione automatica nei prodotti commerciali. Essendo un prodotto con licenza BSD, OpenCV rende facile per le aziende utilizzare e modificare il codice.

La libreria ha più di 2500 algoritmi ottimizzati, che include una serie completa di algoritmi classici e all'avanguardia di visione artificiale e di apprendimento automatico. Questi algoritmi possono essere usati per rilevare e riconoscere volti, identificare oggetti, classificare azioni umane in video, tracciare i movimenti della telecamera, tracciare oggetti in movimento, estrarre modelli 3D di oggetti, produrre nuvole di punti 3D da telecamere stereo, unire immagini per produrre un'immagine ad alta risoluzione di un'intera scena, trovare immagini simili da un database di immagini, rimuovere gli occhi rossi da immagini scattate con il flash, seguire i movimenti degli occhi, riconoscere scenari e stabilire marcatori per sovrapporli alla realtà aumentata, ecc. OpenCV ha più di 47 mila persone di comunità di

utenti e un numero stimato di download superiore a 18 milioni. La libreria è ampiamente utilizzata in aziende, gruppi di ricerca e da enti governativi.

Insieme ad aziende ben consolidate come Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota che impiegano la libreria, ci sono molte startup come Applied Minds, VideoSurf e Zeitera, che fanno ampio uso di OpenCV. Gli usi di OpenCV spaziano dal rilevamento di intrusioni in video di sorveglianza in Israele, al monitoraggio di attrezzature da miniera in Cina, aiutando i robot a navigare e raccogliere oggetti al Willow Garage, al rilevamento di incidenti di annegamento in piscina in Europa, all'esecuzione di arte interattiva in Spagna e New York, al monitoraggio delle piste d'atterraggio per in Turchia, al controllo delle etichette sui prodotti nelle fabbriche di tutto il mondo, al rilevamento rapido dei volti in Giappone.

Ha interfacce C++, Python, Java e MATLAB e supporta Windows, Linux, Android e Mac OS. OpenCV si orienta principalmente verso applicazioni di visione in tempo reale e sfrutta le istruzioni MMX e SSE quando disponibili. Un'interfaccia CUDA e OpenCL completa è in fase di sviluppo attivo in questo momento. Ci sono oltre 500 algoritmi e circa 10 volte tante funzioni che compongono o supportano questi algoritmi. OpenCV è scritto nativamente in C++.

### 2.1.6 Raspberry Pi OS

Un Raspberry Pi è una scheda con processore ARM, progettata per ospitare sistemi operativi basati sul kernel Linux. Il sistema operativo utilizzato è Raspberry Pi OS [29] (ex Raspbian). Si tratta di un sistema operativo free basato su Debian, ottimizzato per l'hardware del Raspberry. È fornito con tutti i programmi di base e le utilities che ci si aspetta da un sistema operativo general purpose. Supportato dalla Raspberry foundation, è divenuto famoso per essere particolarmente veloce e performante. Il modo più facile per installare Raspberry Pi OS sulla propria Pi è quello di caricare il suo file immagine su una scheda micro SD. Esistono due versioni di Raspberry Pi OS, la prima dotata di interfaccia grafica user-friendly (versione with desktop) e un'altra priva di interfaccia grafica (versione lite) da usare con il protocollo SSH.

### 2.1.7 Protocollo SSH

In informatica e telecomunicazioni SSH [36] (Secure SHell, shell sicura) è un protocollo che permette di stabilire una sessione remota cifrata tramite interfaccia a riga di comando con un altro host di una rete informatica. È il protocollo che ha sostituito l'analogico, ma insicuro, Telnet. Il client SSH ha un'interfaccia a riga di comando simile a quella di telnet e rlogin, ma l'intera comunicazione (ovvero sia l'autenticazione - mutua - che la sessione di lavoro) avviene in maniera cifrata. Per questo motivo, SSH è diventato uno standard di fatto per l'amministrazione remota di sistemi UNIX e di dispositivi di rete, rendendo obsoleto il protocollo telnet, giudicato troppo pericoloso per la sua mancanza di protezione contro le intercettazioni.

Il client ed il server SSH sono installati o installabili su molte versioni di UNIX, GNU/Linux, macOS e Microsoft Windows. Inoltre è disponibile come strumento di amministrazione su alcuni apparati di rete. La IANA (Internet Assigned Numbers Authority) ha assegnato al servizio SSH la porta 22 TCP e UDP, anche se è comunque possibile implementare il servizio SSH su altre porte non definite dalla IANA.

La prima versione dell'SSH era completamente open source, mentre la seconda è diventata commerciale; esiste comunque una versione libera detta OpenSSH che si basa sulla prima versione, ma che fornisce supporto alla seconda versione. Il protocollo SSH si basa su un'architettura rappresentabile in tre distinti livelli: Transport Layer Protocol, User Authentication Protocol e Connection Layer Protocol. La divisione in tre livelli indipendenti garantisce un'estrema flessibilità del protocollo: l'indipendenza logica dei livelli permette di modificare la configurazione della connessione senza dover modificare i livelli superiori o inferiori del protocollo, per esempio si può modificare il protocollo di compressione delle informazioni senza dover cambiare il protocollo di cifratura o di autenticazione del sistema.

È importante specificare che i livelli sopra citati non hanno alcuna relazione con i livelli dello standard ISO/OSI essendo livelli interni al protocollo SSH che a sua volta è posizionabile subito sopra al livello di trasporto ISO/OSI. Di fatto i livelli interni al protocollo SSH coprono gli ultimi tre livelli della pila ISO/OSI applicando un'organizzazione diversa da quella prevista dallo standard per le reti. Inoltre, esistono implementazioni software di VNC per sistemi operativi diversi, tutte capaci di interoperate tra di loro.

### 2.1.8 Protocollo VNC

VNC [39] è un acronimo per Virtual Network Computing. Si tratta di un protocollo, in origine elaborato dai laboratori di ricerca di Olivetti ed Oracle e, in seguito, sviluppato da AT&T, che consente di visualizzare il desktop di un computer remoto e di interagire con esso. È una tecnologia che consente la condivisione in remoto di desktop grafici. Può trasmettere gli eventi generati da tastiera e mouse da un ambiente grafico ad un altro, attraverso la rete, permettendo così di controllare un altro computer a distanza.

È basato sul protocollo di rete RFB (Remote Frame Buffer) che scambia i dati tra due computer ed è realizzato su un modello client/server: nell'utilizzo tipico un VNC client (o viewer) invia gli eventi da interpretare e un VNC server li riceve e li trasforma in input locali; per questo motivo, i software VNC vengono spesso utilizzati per fornire assistenza remota a pc distanti nella rete. Uno dei punti di forza di VNC è il fatto di essere indipendente dalla piattaforma, esistono implementazioni software di VNC per sistemi operativi diversi, tutte capaci di interoperate tra di loro.

In quanto protocollo multicanale, dispone di svariati canali virtuali distinti per il trasporto dei dati di presentazione, delle comunicazioni con le periferiche seriali, delle informazioni di licenza e dei dati a elevato livello crittografico come ad esempio input da tastiera ed attività del mouse. RDP è inoltre semantico, ossia è a conoscenza di controlli, caratteri ed altre primitive grafiche simili. Ciò significa che quando si esegue il rendering di uno schermo su una rete, queste informazioni vengono utilizzate per comprimere il flusso di dati in modo

significativo rendendo il protocollo più veloce e reattivo. Infine può supportare più utenti remoti connessi alla stessa macchina che si ignorano completamente l'un l'altro.

Fra le caratteristiche rilevanti figurano anche la cifratura dei dati a 128 bit tramite l'algoritmo RC4, l'implementazione degli algoritmi di compressione video H.264/AVC per l'ottimizzazione della banda, la funzionalità di audio redirection, che permette di ascoltare l'audio del desktop remoto sulla postazione locale, di printer e port redirection, nonché il supporto multi monitor. Tutta l'attività del protocollo RDP è gestita dal driver di periferica di Terminal Server e dai suoi componenti, tra cui l'RDP driver che gestisce i trasferimenti, la crittografia e l'interfaccia utente. Per questo studio è stata utilizzata XRD [16], un'implementazione open source del protocollo RDP.

### 2.1.9 Protocollo RDP

Il protocollo RDP (Remote Desktop Protol) [31] è un protocollo di rete proprietario, sviluppato da Microsoft, che permette la connessione remota utilizzando di default la porta TCP e UDP 3389. I client RDP sono attualmente disponibili per Windows, MacOS, Linux, Unix, mentre il server è integrato di base in tutti i sistemi Windows. RDP è un protocollo di tipo multicanale, basato sulla famiglia ITUT.120, e compatibile con diverse topologie di rete, tra le quali TCP/IP, ISDN e diversi protocolli LAN.

### 2.1.10 YOLOv5

YOLO rappresenta lo stato dell'arte degli algoritmi per la risoluzione del task di object detection in real-time, e YOLOv5 è l'evoluzione delle precedenti versioni. Il modello è stato migliorato in maniera iterativa nel tempo, ciò gli ha permesso di raggiungere prestazioni elevate sui principali datasets per i benchmarks di object detection: Pascal VOC (visual object classes) [8] e MSCOCO (oggetti comuni nel contesto) [7], ed è ampiamente utilizzata nell'industria.

L'architettura di rete di YOLOv5 è mostrata nella figura. Ci sono tre ragioni per cui abbiamo scelto YOLOv5 come algoritmo;

- In primo luogo YOLOv5 ha incorporato la rete parziale cross stage (CSPNet) [40] in Darknet, creando CSPDarknet come sua backbone. CSPNet risolve i problemi di informazioni ripetute sul gradiente in dorsali su larga scala, e integra i cambiamenti di gradiente nella mappa delle caratteristiche, diminuendo così i parametri e i FLOPS (operazioni in virgola mobile al secondo) del modello, che oltre a garantire la velocità e l'accuratezza dell'inferenza, permette di ridurre la dimensione del modello. Nel task di rilevamento degli animali selvatici, la velocità e accuratezza è imperativa, e la dimensione compatta del modello determina anche la sua efficienza su dispositivi di edge computing con poche risorse.
- In secondo luogo, YOLOv5 ha applicato il path aggregation (PANet) [41] come collegamento per aumentare il flusso di informazioni. PANet adotta un'innovativa

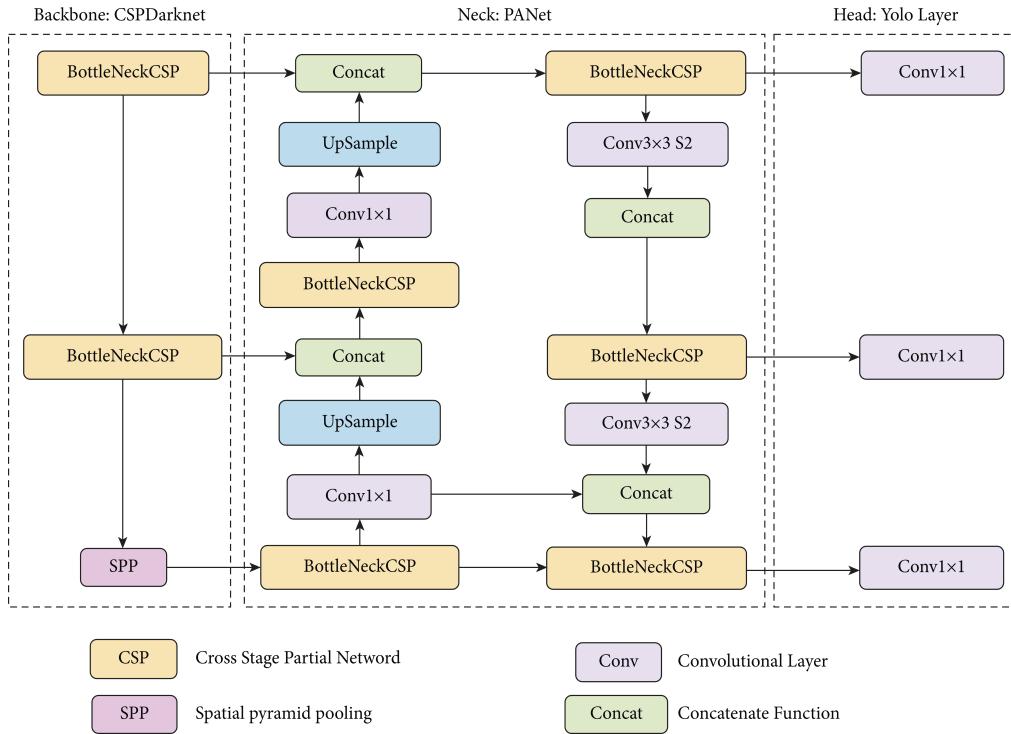


Figura 2.1: Architettura di YOLOv5.

struttura di rete piramidale (FPN) con un percorso bottom-up migliorato, che migliora la propagazione delle features di basso livello. Allo stesso tempo, l'adaptive feature pooling, che collega griglia di caratteristiche e tutti i livelli di features, viene utilizzato per rendere le informazioni utili in ogni livello di caratteristiche si propaghino direttamente alla rete successiva. PANet migliora l'utilizzo di accurati segnali di localizzazione nei livelli inferiori, il che può ovviamente migliorare la precisione di localizzazione dell'oggetto.

- Infine, il layer di output di YOLOv5, cioè il livello YOLO, genera 3 diverse dimensioni di mappe di caratteristiche per ottenere una previsione multiscala [30], permettendo al modello di gestire oggetti piccoli, medi e grandi. Solitamente gli animali possono trovarsi a differenti distanze. Il rilevamento multi-scala assicura che il modello possa seguire i cambiamenti dimensione degli animali a seconda della loro prossimità all'obiettivo

### 2.1.11 Transfer Learning

Il TL è un metodo di apprendimento automatico con la prospettiva di fornire una soluzione migliore e più veloce con meno sforzo per raccogliere i dati di allenamento necessari e ricostruire il modello [25]. Oltre ai suoi grandi progressi nei domini della classificazione dei documenti, del parlato e delle immagini [1], recenti studi neurofisiologici [2] hanno anche dimostrato l'efficacia della TL per migliorare le prestazioni di classificazione attraverso l'utilizzo/apprendimento dei dati/informazioni di altri individui. Prima di descrivere il quadro cTL proposto in questo studio, le notazioni e le definizioni di base del TL sono riassunte come segue su una revisione del TL.

Un dominio  $D$  consiste di due componenti: uno spazio di caratteristiche  $\mathcal{X}$  e una distribuzione marginale di probabilità  $P(X)$ , in cui  $X = \{x_1, \dots, x_n\} \in \mathcal{X}$ . Dato un dominio specifico  $D = \{\mathcal{X}, P(X)\}$ , un task consiste di due componenti: uno spazio di etichette  $\mathcal{Y}$  e una funzione predittiva oggettiva  $f(\bullet)$ , indicata con  $T = \{\mathcal{Y}, f(\bullet)\}$  che può essere appresa dalle coppie di dati di allenamento  $\{x_i, y_i\}$ , dove  $x_i \in \mathcal{X}$  e  $y_i \in \mathcal{Y}$ , la  $f(\bullet)$  può essere usata per predire l'etichetta di una nuova istanza  $x$ , che può essere riscritta dalla forma probabilistica della distribuzione di probabilità condizionata  $P(Y|X)$ . Un task può essere definito come  $T = \{\mathcal{Y}, P(Y | X)\}$ . Con le notazioni di dominio e compito, il TL è definito come segue: Dato un dominio sorgente  $D_S$  e un task di apprendimento  $T_S$ , e un dominio target  $D_T$  e un task di apprendimento  $T_T$ , il TL mira a migliorare l'apprendimento della funzione predittiva target  $f(\bullet)$  in  $D_T$  usando la conoscenza in  $D_S$  e  $T_S$ , dove  $D_S \neq D_T$ , o  $T_S \neq T_T$  [19].

### 2.1.12 mAP

La Mean Average Precision (mAP) [22] è la metrica più significativa per i rilevatori di oggetti. Incorpora il trade-off tra precision e recall, e così facendo, tiene conto di entrambi i tipi di errori, falsi positivi (FP) e falsi negativi (FN). Questa proprietà rende mAP applicabile alla maggior parte dei casi d'uso.

Poiché la mAP si basa su precision e recall, si basa anche sulla matrice di confusione. Per i rilevatori di oggetti, la matrice di confusione viene calcolata controllando se la classe della predizione corrisponde alla classe della ground truth e controllando se l'IoU è superiore a una certa soglia, la maggior parte delle volte superiore a 0,5.

Per calcolare la mAP è necessario calcolare la media della curva Precision-Recall per ogni classe.

La curva Precision-Recall descrive il conflitto tra precisione e recall. In genere,

- i modelli con alta precisione e basso recall producono previsioni molto sicure ma mancano una parte delle istanze.
- I modelli con bassa precisione e alto recall possono trovare la maggior parte degli oggetti, ma le previsioni sono falsi positivi in una certa misura, e la fiducia diminuisce.

Si costruisce calcolando i valori di precisione e recall per diverse soglie di fiducia. Cioè, se la soglia di fiducia è a 0,9, solo le previsioni in cui il classificatore ha una fiducia superiore al 90% sono contate come previsioni positive. La precisione sarà piuttosto alta. Viceversa, se la soglia è a 0,1, molte più previsioni saranno incluse in quelle positive e il recall sarà grande. L'Average Precision non è altro che l'area al di sotto della Precision-Recall curve. I modelli perfetti hanno un valore di AP vicino a 1.

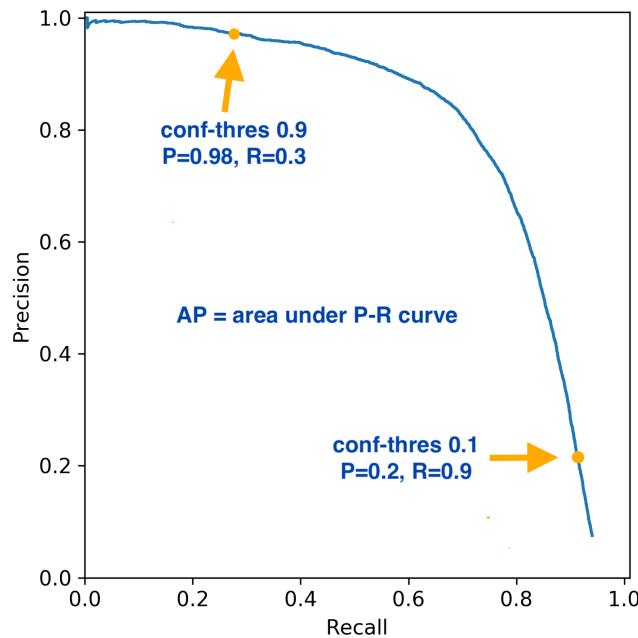


Figura 2.2: Esempio di curva Precision-Recall.

la mAP non è altro che la media dell'AP calcolata per tutte le classi, nel nostro esempio sono 2, 'goat' e 'boar'.

Spesso, si vede il mAP citato come mAP@0.5 o mAP@0.75. Questa notazione indica quale soglia di IoU è stata usata per calcolare la matrice di confusione. Questo è importante da notare perché la soglia di IoU può influenzare il mAP in modo sostanziale: un IoU basso aumenterà il tuo mAP. Come accennato sopra, il più delle volte si usa 0,5 come soglia IoU. A volte, per esempio, per il dataset COCO, il mAP viene valutato facendo la media su IoU da [.5,.95] in passi di .05.

## 2.2 Intersection over Union

IoU (Indice di Jaccard) [15] è usato per stimare quanto bene una bounding box corrisponda ai dati della ground truth. Un valore superiore corrisponde ad oggetti identificati in maniera rigorosa.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$



Figura 2.3: Esempio di Intersection over Union.

### 2.2.1 Roboflow

Roboflow [35] è un servizio cloud che permette di annotare un dataset. È stato scelto per l'elevata interoperabilità dei formati di labeling dei datasets. Inoltre permette la collaborazione in team per etichettare le immagini che compongono il dataset. Permette di dividere il dataset in training, test e validation e compiere task di data augmentation sulle immagini.

### 2.2.2 Weights & Biases

Weights & Biases [42] è una dashboard che permette di visualizzare le statistiche sugli addestramenti del modello. È stato scelto perché permette di svolgere un'operazione di debugging sulle bounding box usando diverse thresholds. Anche questo servizio offre una funzione di collaborazione, gratuita per gli studenti universitari.

## 2.3 Hardware

### 2.3.1 Raspberry Pi

Il Raspberry Pi [27] è un single board computer, cioè una scheda elettronica implementante un intero computer o quasi, dotato di processore ARM con un fattore di forma estremamente ridotto (pari a quello di una carta di credito) che integra funzionalità avanzate per molteplici utilizzi che spaziano dalle applicazioni domestiche a quelle industriali.

Fu sviluppato inizialmente dalla Raspberry Pi Foundation, un'organizzazione di beneficenza britannica, come strumento per promuovere lo studio dell'informatica nelle scuole

e nei paesi in via di sviluppo. Sebbene l’obiettivo iniziale fosse avvicinare i ragazzi alla programmazione, con un computer economico ma potente, ha finito per risvegliare la passione per la programmazione in adulti creativi e appassionati di tecnologia. Non offre la potenza di un computer desktop, ma può essere utilizzato per tutte quelle attività che non necessitano delle prestazioni di un PC costoso e ingombrante. Ad un prezzo irrisorio di circa 30-40€, questo Pi delle dimensioni di una carta di credito offre un processore ARM, RAM, funzionalità grafiche e tutte le porte hardware standard che si trovano di solito in un computer.

Il nome “Raspberry”, che in italiano significa lampone, è un omaggio alle prime aziende di computer che nacquero, il cui nome prendeva spunto da nomi di frutti, come Apple, Tangerine Computer Systems, Apricot Computers e Acorn (da cui fu preso spunto per il design del microcomputer). D’altro canto il nome “Pi” deriva dall’idea originaria di costruire un mini computer avente soltanto Python come linguaggio di programmazione.

Le ragioni per cui il Raspberry Pi ha riscosso un successo enorme in giro per il mondo sono riconducibili ai seguenti fattori:

- Costo contenuto.
- Dimensioni molto contenute.
- Bassissimo consumo di potenza elettrica (meno di 5W).

Bisogna sottolineare che il dispositivo di archiviazione è venduto separatamente. Ovviamente, oltre alla micro SD, è necessario procurarsi tutto ciò che si desidera connettere alla scheda, perché niente è incluso nel pacchetto. L’elenco essenziale per l’utilizzo generico è il seguente:

- Scheda micro SD.
- Alimentatore micro USB.
- Cavo micro HDMI a HDMI per connettere il Pi alla propria TV o monitor.
- Tastiera, Mouse e Monitor/TV.
- Case per il Pi.
- Cavo Ethernet (solo modello B/B+), se per qualche motivo non si vuole utilizzare il WiFi incorporato nei modelli più recenti.

È doveroso sottolineare, che non è necessario alcun monitor dedicato. Di solito, una volta completate le impostazioni, si collegano mouse e tastiera alle porte USB del Pi e il monitor via HDMI. Ciò provoca generalmente un avvio con il sistema operativo selezionato e la corrispondente GUI, interfaccia grafica utente. Alcuni progetti, però, non necessitano che il Raspberry Pi utilizzi una normale GUI. In questi casi, è possibile collegare il Pi alla rete e accedervi da remoto tramite il protocollo SSH (Secure Shell). Se il Raspberry Pi esegue

una variante di Linux, inoltre è molto probabile che l'opzione SSH sia disponibile come modalità predefinita. Se si utilizza un Mac, si può usare Terminal o Putty con Windows. Con questo metodo, è necessario effettuare l'accesso al Pi da un normale PC ed eseguire ogni operazione su di esso da remoto tramite linea di comando.

### 2.3.2 Specifiche tecniche Raspberry Pi Model 4B

- **SoC:** Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz.
- **GPU:** Broadcom VideoCore VI Dual.
- **RAM:** 2GB LPDDR4-3200 SDRAM.
- **Ethernet:** Gigabit Ethernet.
- **WiFi:** 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless.
- **Bluetooth:** Bluetooth 5.0, BLE
- **Porte:** microSD, micro HDMI , Jack 3.5 mm, 4×USB 2.0.
- **Interfacce:** CSI Camera Connector, DSI Display Connector, GPIO header 40-pin,
- **Consumo medio:** 3W.
- **Dimensioni:** 85 x 56 x 17 mm.
- **Peso:** 46 g.

### 2.3.3 Modulo Camera

In prima istanza è stata valutata la possibilità di utilizzare una webcam USB, ma l'ipotesi è stata scartata per due ragioni: i consumi e l'impossibilità di avere un BUS (Binary Unit System) dedicato. Si è dunque optato per l'utilizzo della porta Camera Serial Interface (CSI). È stata selezionata una camera assemblata e distribuita dalla Raspberry Pi Foundation. Il Modulo Camera [28] del Raspberry Pi presenta le seguenti caratteristiche:

- **Sensore:** Sony IMX219PQ.
- **Risoluzione immagini:** 3280 x 2464 pixel (8 Megapixels).
- **Risoluzione video:** 1080p30, 720p60, 640x480p60/90.
- **Dimensioni:** 25mm x 23mm x 9mm.
- **Peso:** 3g.
- **Interfaccia:** Camera Serial Interface (CSI).

## Capitolo 3

# Addestramento dei modelli

### 3.1 Pre-processing e Labeling

Innanzitutto è stato necessario filtrare le immagini, selezionando solo quelle d'interesse per il nostro studio. Ovviamente la fototrappola può attivarsi anche senza che alcun animale sia presente nell'obiettivo (Falso Positivo), queste immagini spurie, non idonee per il nostro classificatore, sono state rimosse manualmente. In secondo luogo è stato necessario etichettare gli animali presenti nelle immagini del dataset. Alla fine della fase di etichettatura delle immagini sono stati estratti 1883 campioni di cinghiale selvatico e 2365 campioni di capra selvatica. Le immagini sono state ridimensionate a 416\*416 poichè la runtime gratuita di Google Colab possiede memoria centrale limitata. È stato applicato uno split del dataset 80-20-10 rispettivamente per training set, test set e validation set.

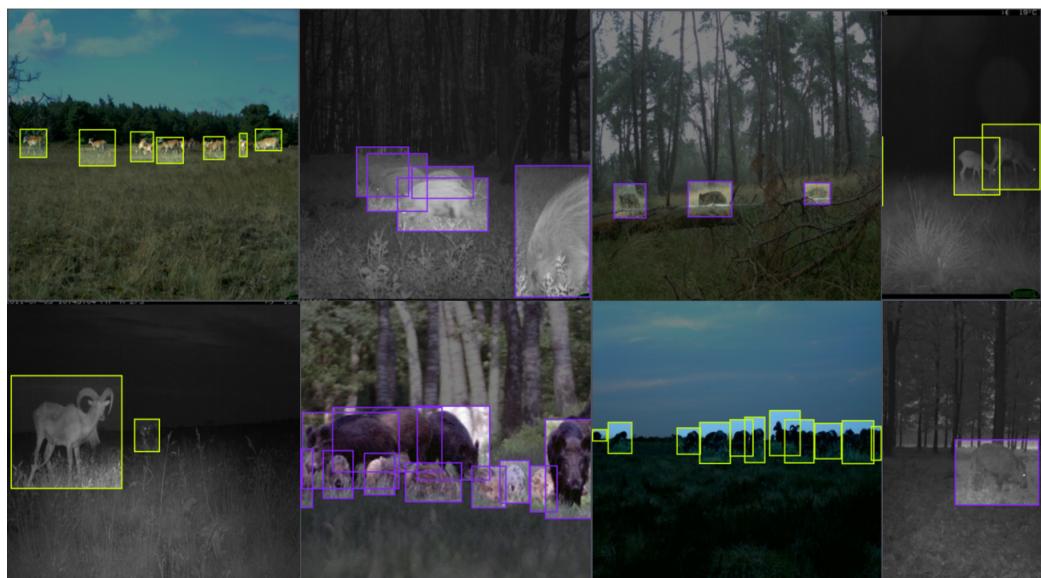


Figura 3.1: Alcuni campioni del dataset.

È interessante notare che sono state selezionate immagini ad infrarosso questo per permettere ad un dispositivo esterno di riconoscere gli animali anche nelle ore notturne. Una feature estremamente utile che offre la piattaforma Roboflow permette di convertire il dataset in diversi formati a seconda del modello che intendiamo addestrare

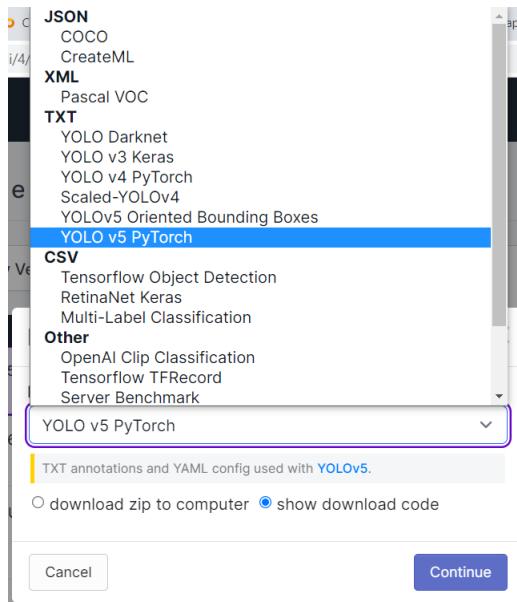


Figura 3.2: Formati di esportazione del dataset su Roboflow.

### 3.1.1 YOLOv5

Per addestrare il modello YOLOv5 su Colab è stato utilizzato il seguente comando:

```
$ !python 'train.py' --img '416' --batch '128' --epochs '100' --save-period '10' --freeze '10' --data '{dataset.location}/data.yaml' --weights 'yolov5s.pt' --cache
```

- ***img 416***: indica la risoluzione delle immagini in input;
- ***batch 128***: indica la batch size, questo parametro dipende dal task affrontato e dall'hardware, se abbiamo a disposizione molta memoria video nella nostra GPU possiamo aumentare questo parametro, che ci permette di avere un training più rapido, attenzione a non utilizzare valori troppo alti perché si rischia di saltare l'ottimo globale;
- ***epochs 100***: indica quante volte il modello osserverà gli esempi contenuti nel training set, un valore troppo basso ed il modello non riconoscerà gli oggetti in fase di validazione, un valore troppo alto tenderà all'overfitting. Per esempio se il training contiene solo animali di colore marrone, il modello non sarà in grado di identificare (generalizzare) animali della stessa specie ma di colore grigio;

- **save-period 10:** indica che sarà effettuato un checkpoint con relativo salvataggio dei valori ottimali dei pesi del modello ogni 10 epoche;
- **freeze 10:** indica il numero dei primi n layer che non saranno addestrati (i valori dei pesi di questi layers resteranno invariati) utilizzando la tecnica del Transfer Learning [37], ciò permette di ottenere un addestramento più rapido con meno risorse con il tradeoff di una precisione leggermente inferiore;
- **data:** indica il file YAML che contiene informazioni sul dataset;
- **weights:** utilizza i valori dei pesi pre-addestrati di YOLOv5 Small (yolov5s.pt);
- **cache:** rende il training più rapido salvando le immagini nella memoria cache;

### 3.1.2 Metriche di valutazione di YOLOv5

Per valutare le performance del classificatore è stato utilizzato il seguente comando:

```
$ !python 'val.py' --weights 'best.pt' --img '416' --conf '0.1'  
--data '{dataset.location}/data.yaml'
```

- **weights:** il file contenente i valori dei pesi addestrati in precedenza, che saranno in seguito esportati sul Raspberry;
- **img 416:** indica la risoluzione delle immagini in input;
- **conf 0.1:** indica la threshold (soglia) di confidenza minima per identificare un oggetto, una threshold di confidenza maggiore riduce i Falsi Positivi ma identifica meno oggetti, mentre un valore inferiore permette di identificare un maggior numero di oggetti ma anche i Falsi Positivi (vegetazione a forma di animale);
- **data:** indica il file YAML che contiene informazioni sul dataset;

dopo aver eseguito questo comando abbiamo ottenuto i risultati della validazione del modello nella cartella **runs/val/exp**.

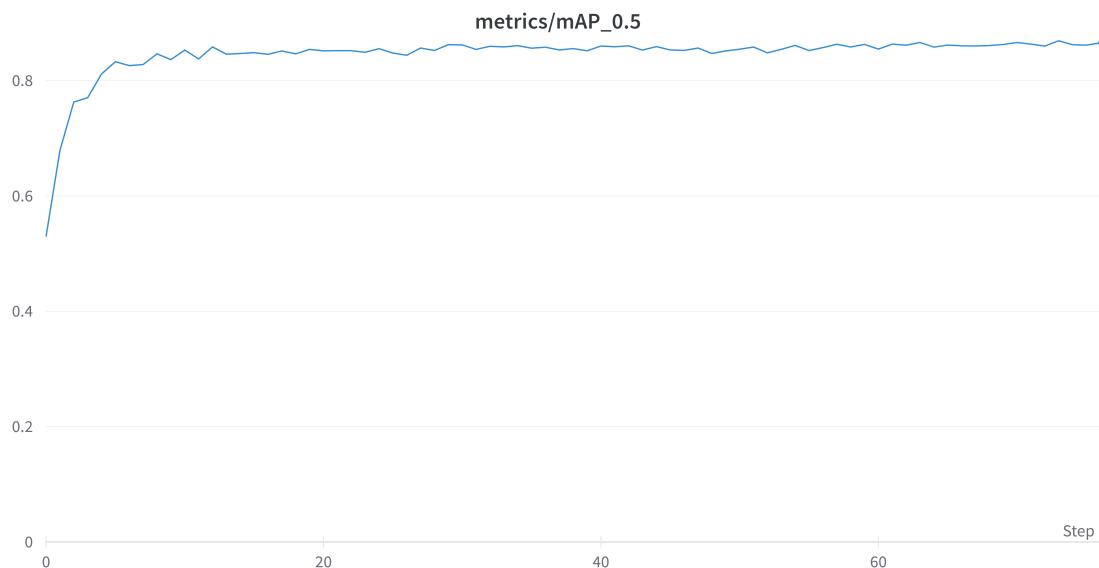


Figura 3.3: Mean Average Precision sul nostro dataset di YOLOv5.

Come è possibile notare nella figura 3.3, la Mean Average Precision converge nella settima epoca, e resta costante per tutta la fase di training.

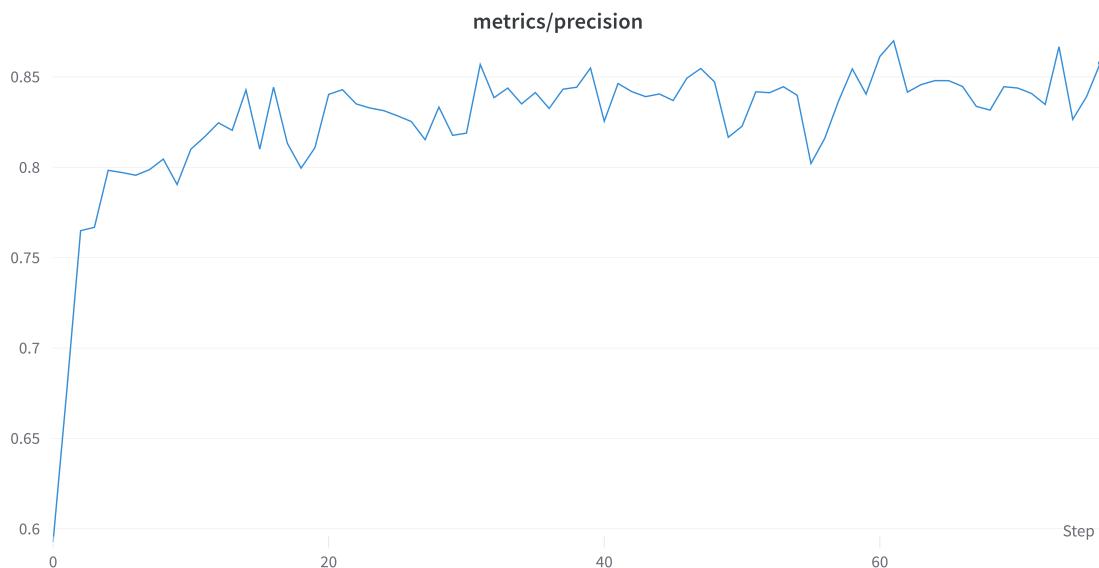


Figura 3.4: Precision sul nostro dataset di YOLOv5.

Come da figura 3.4 si nota che la precision raggiunge il valore di 0.8 anche se con una maggior fluttuazione dei valori nelle varie epoche.

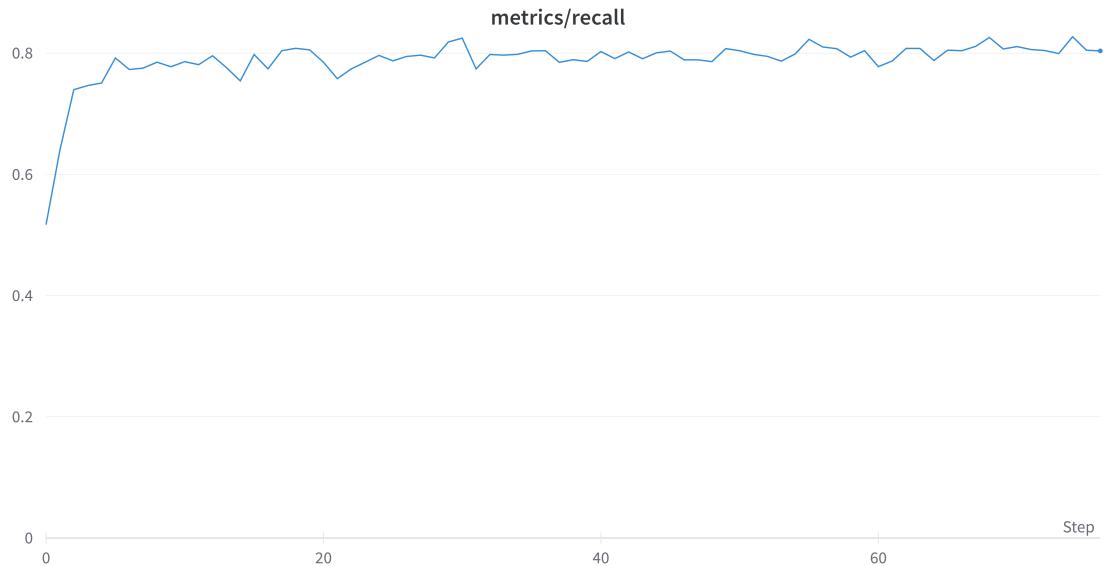


Figura 3.5: Recall sul nostro dataset di YOLOv5.

Come da figura 3.5 si nota che la recall raggiunge valori elevati (0,8) nonostante il trade-off con la precision [26].

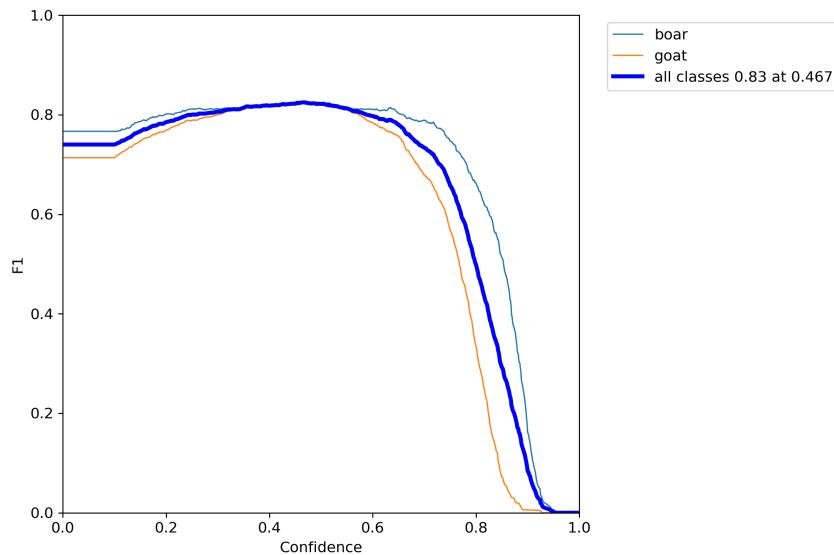


Figura 3.6: F1 score curve sul nostro dataset di YOLOv5.

Dalla figura 3.6 possiamo notare che il valore più adatto della threshold per identificare gli animali selvatici è compreso fra 0,4 e 0,6.

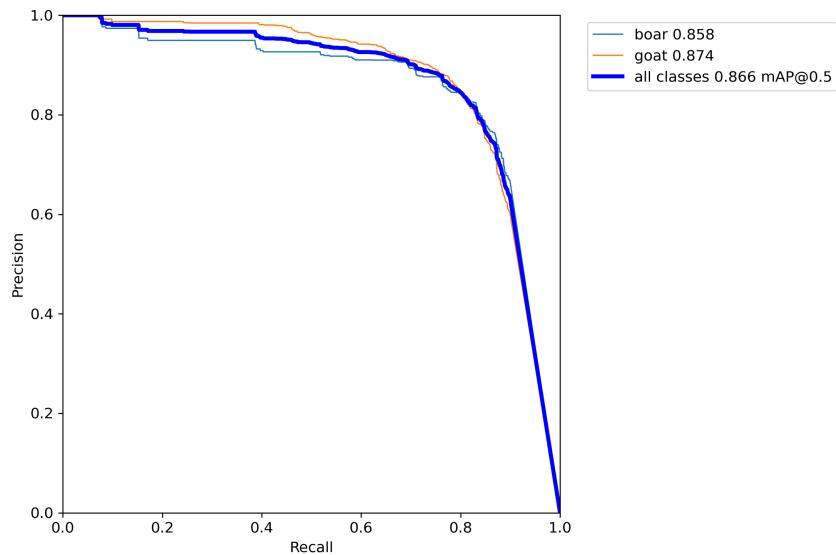


Figura 3.7: PR curve sul nostro dataset di YOLOv5.

Notiamo dalla figura 3.7 la tipica "curva a gomito" segno che il nostro classificatore ha appreso come distinguere le due classi di animali.

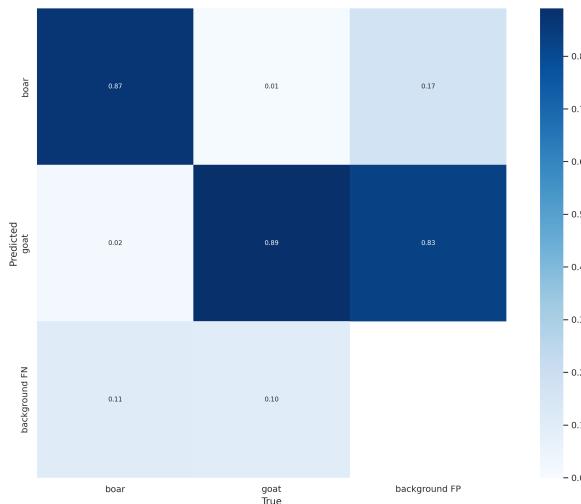


Figura 3.8: Confusion matrix sul nostro dataset di YOLOv5.

È possibile notare in figura 3.8 che la confusion matrix possiede valori elevati sulla diagonale, ciò implica una corretta classificazione delle immagini di test.

## 3.2 Tensorflow Lite

Prima di addestrare il modello è stato necessario convertire il dataset nel formato Pascal VOC in seguito è stata utilizzata la seguente riga di codice:

```
model = object_detector.create(train_data, model_spec.get('efficientdet_lite0'),
                               validation_data='val_data', epochs='20')
```

il metodo **create** della classe **object\_detector** carica il dataset e addestra il modello, nel nostro caso il modello è '**efficientdet\_lite0**' addestrato per **20** epoche. Per validare il modello è stato utilizzato il metodo **evaluate** della classe **model**

```
model.evaluate(val_data)
```

Infinie per salvare i valori dei pesi dopo l'addestramento utilizziamo:

```
model.export(export_dir='.', tflite_filename='animals_20_epochs.tflite')
```

### 3.2.1 Metriche di valutazione di efficientdet\_lite

Confrontare le metriche dei due frameworks è stato un compito impegnativo poichè utilizzano metriche di performance differenti. Le metriche in comune sono la Average Precision e la Average Recall che riportiamo confrontate nella seguente tabella 3.1 insieme al numero di parametri di ciascun modello.

|                    | efficientdet | YOLOv5 |
|--------------------|--------------|--------|
| AP50               | 0.606        | 0.866  |
| AP (boar)          | 0.271        | 0.543  |
| AP (goat)          | 0.263        | 0.432  |
| AR                 | 0.485        | 0.807  |
| Parameters (m)     | 3.2          | 7      |
| Inference time (s) | 0.3          | 1.6    |

Tabella 3.1: Tabella comparativa dei due modelli

È possibile notare che il modello YOLOv5 è in grado di riconoscere gli animali selvatici del nostro studio in maniera più accurata (AP50: 0.866), ciò si riflette anche negli scores delle singole classi (0.543, 0.432) e nella recall (0.807). Il vero vantaggio di efficientdet come si può osservare anche dal nome è il minor tempo di inferenza (0.3 sec) e la metà dei parametri necessari (3.2 mln) per il suo funzionamento. L'efficienza del modello è data dalle tecniche di quantizzazione a 8 bit, che riducono l'accuratezza aumentandone la velocità [24]. Questa tecnica è particolarmente utile per gli edge devices che non possiedono GPU altamente performanti.

### 3.2.2 Learning Curve di efficientdet\_lite

La seguente figura indica 3.9 che la loss inizia a decrementare sensibilmente nella terza epoca per poi raggiungere un plateau nella ventesima. Ciò indica che il modello non riesce ad apprendere ulteriormente dai dati per questo è stato deciso di interrompere l'addestramento utilizzando la tecnica dell'Early Stopping [13].

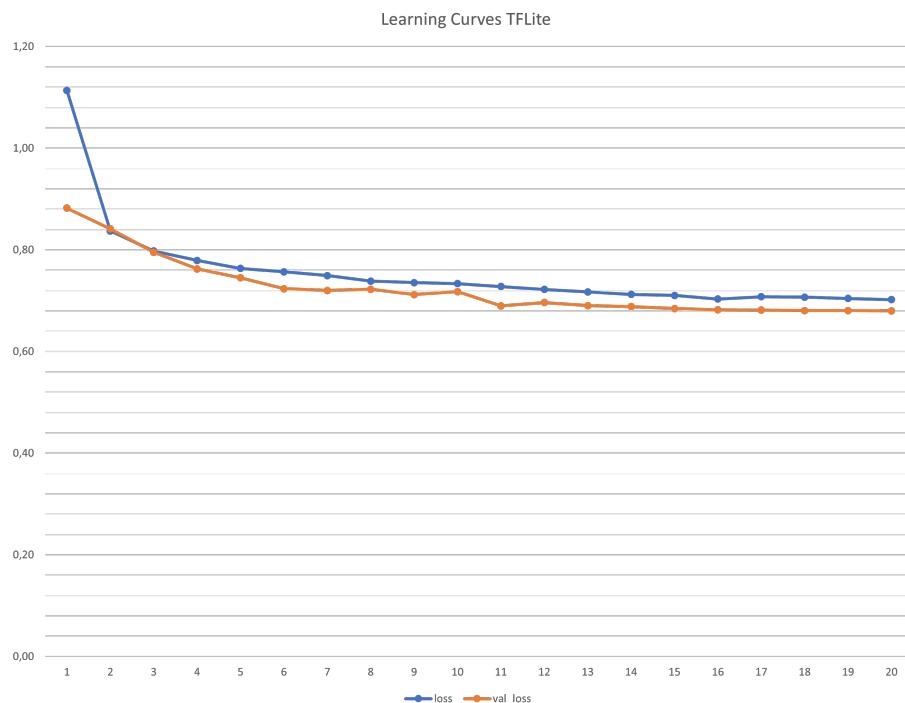


Figura 3.9: Learning Curve di TFLite.

## Capitolo 4

# Esecuzione degli scripts sul Raspberry

Per eseguire il deployment sul Raspberry Pi è stato necessario installare i due framework di deep learning con le relative librerie [11] [14] [32] [33] [34]. Da notare che per il Raspberry bisogna scaricare i wheel in per architettura ARM.

È caldamente consigliato l'utilizzo di un ambiente virtuale Python [38], oppure Anaconda [12], per evitare il conflitto di versione con eventuali librerie installate in precedenza.

### 4.1 YOLOv5

Per avviare l'inferenza di YOLOv5 sul Raspberry aprire un terminale sul Raspberry, attivare l'ambiente virtuale configurato in precedenza e digitare:

```
python detect.py --source 'data/images' --weights 'best.pt' --conf '0.25'
```

- **source:** indica la cartella in cui sono presenti le immagini simulate degli animali;
- **weights:** indica i pesi addestrati nella precedente fase di training
- **conf:** indica la threshold (soglia) di confidenza minima per identificare un oggetto, una threshold di confidenza maggiore riduce i Falsi Positivi ma identifica meno oggetti, mentre un valore inferiore permette di identificare un maggior numero di oggetti ma anche i Falsi Positivi (vegetazione a forma di animale), questo parametro può variare nell'intervallo di valori [0,1];

Una volta avviato il comando verrà creata una cartella ***runs/detect/exp*** al cui interno troveremo le immagini annotate da YOLOv5, il tempo medio per l'inferenza di una singola immagine è di circa 1,6 secondi.

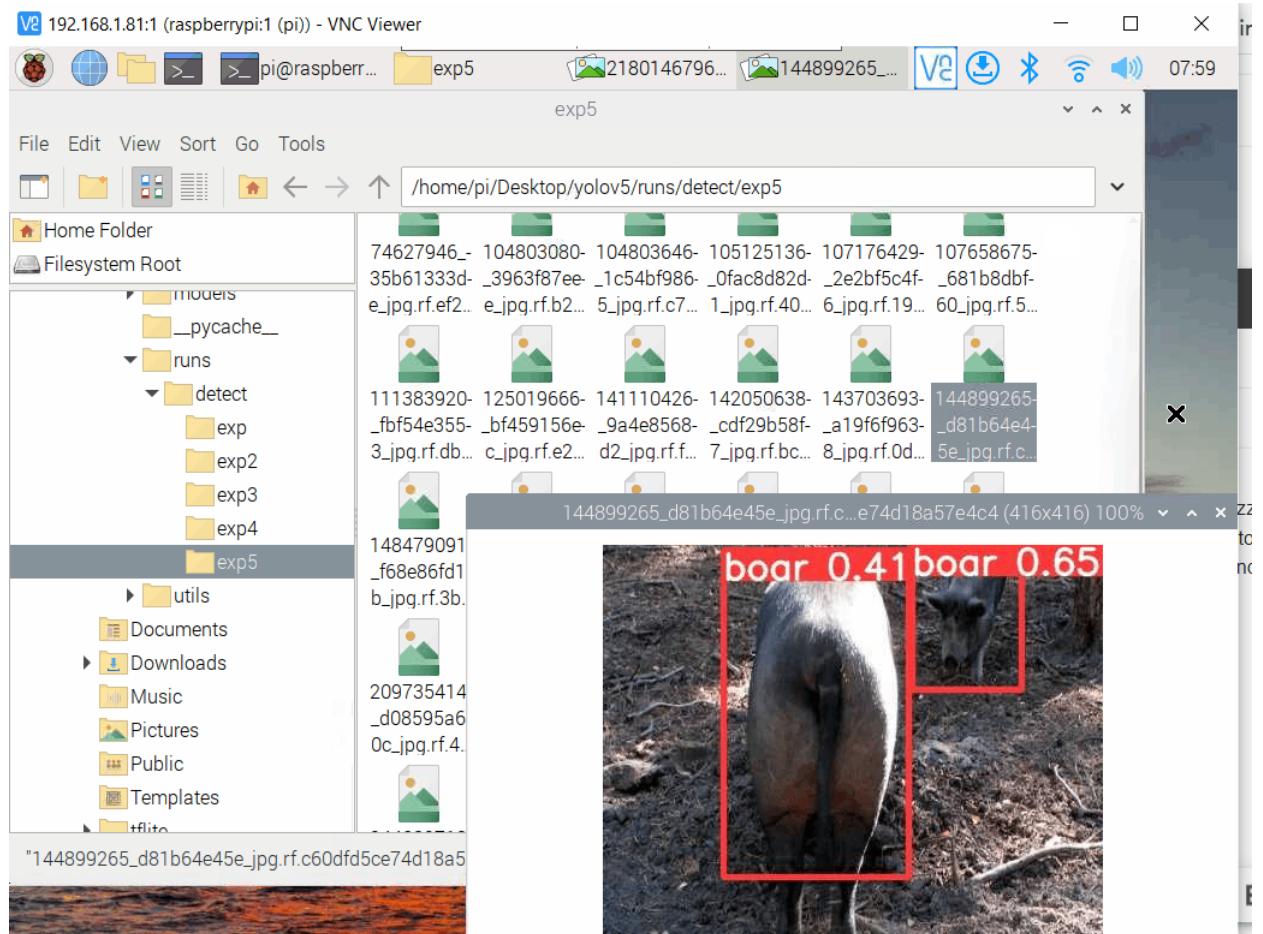


Figura 4.1: Screenshot demo di YOLOv5 sul Raspberry Pi.

## 4.2 Tensorflow Lite

Per avviare l'inferenza di efficientdet\_lite sul Raspberry aprire un terminale sul Raspberry, attivare l'ambiente virtuale configurato in precedenza e digitare:

```
python detect.py --model 'animals_20_epochs.tflite'
```

- **model:** indica i pesi addestrati nella precedente fase di training;

All'avvio del comando si aprirà una finestra con delle immagini che cambiano nel tempo, si tratta delle immagini inferite dal modello efficientdet\_lite.

Il tempo di inferenza del modello efficientdet\_lite è di 0,3 secondi in media.

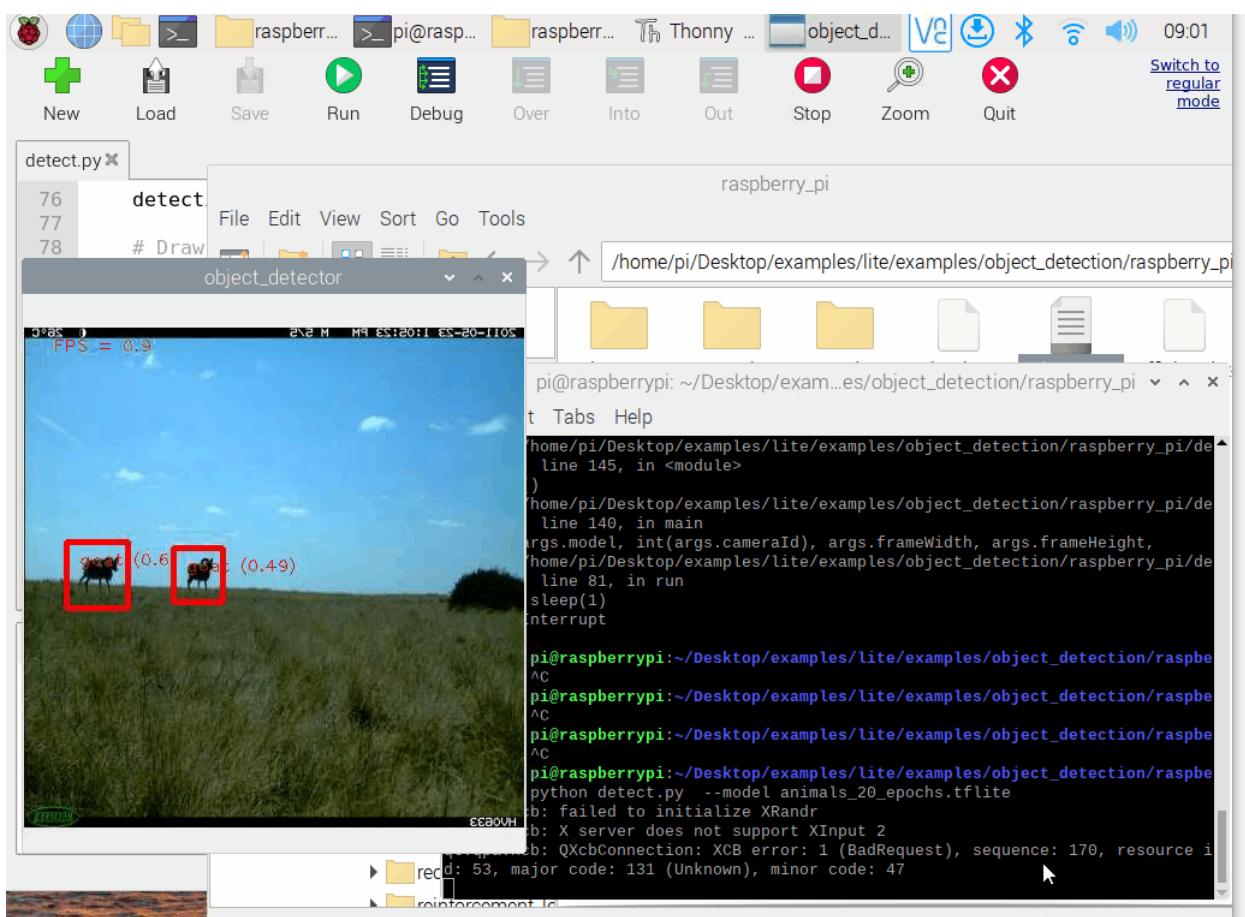


Figura 4.2: Screenshot demo di TFLite sul Raspberry Pi.

## **Capitolo 5**

### **Conclusioni**

In questo lavoro di tesi è stato evidenziato come la tecnologia può influire e Confrontare i 2 framework dire ke si doveva metterla la telecamera all aperto ma non è stato dfatto ecc

# Bibliografia

- [1] Dai W. et al. «Boosting for transfer learning». In: *Proceedings of the 24th International Conference on Machine learning* (2007).
- [2] Kang H. et al. «Composite common spatial pattern for subject-to-subject transfer». In: *IEEE Signal Process* (2009).
- [3] Ambiente di Sviluppo Integrato PyCharm. URL: <https://www.jetbrains.com/pycharm/>.
- [4] Cinghiali danneggiano azienda agricola: Regione Umbria condannata al risarcimento. URL: <http://www.umbria24.it/cronaca/cinghiali-danneggiano-azienda-agricola-regione-umbria-condannata-al-risarcimento>.
- [5] Coldiretti. URL: <https://lombardia.coldiretti.it/news/cinghiali-danni-per-mezzo-milione-nel-lodigiano-rota-regole-certe-per-contenimento/>.
- [6] Compensation For Traffic Accidents Caused By Wild Animals In Italy - Transport - Italy. URL: <https://www.mondaq.com/italy/rail-road-cycling/879306/compensation-for-traffic-accidents-caused-by-wild-animals-in-italy>.
- [7] Dataset MSCOCO. URL: <https://cocodataset.org/#home>.
- [8] Dataset VOC. URL: <http://host.robots.ox.ac.uk/pascal/VOC/>.
- [9] Google Colab. URL: <https://colab.research.google.com/>.
- [10] Marcel P Huijser et al. *The Comparison of Animal Detection Systems in a Test-Bed: A Quantitative Comparison of System Reliability and Experiences with Operation and Maintenance*. Rapp. tecn. Montana. Department of Transportation, 2009.
- [11] Install PyTorch on Raspberry Pi 4. URL: <https://qengineering.eu/install-pytorch-on-raspberry-pi-4.html>.
- [12] Installing conda packages, Anaconda documentation. URL: <https://docs.anaconda.com/anaconda/user-guide/tasks/install-packages/>.
- [13] Introduction to Early Stopping. URL: <https://towardsdatascience.com/early-stopping-a-cool-strategy-to-regularize-neural-networks-bfdeca6d722e>.
- [14] Introduction to object detection on Raspberry Pi. URL: <https://www.youtube.com/watch?v=mNjXEybFn98>.
- [15] IoU (Intersection over Union) - Hasty visionAI Wiki. URL: <https://wiki.hasty.ai/metrics/iou-intersection-over-union>.

- [16] *Jupyter Notebook*. URL: <https://jupyter.org/>.
- [17] *Libreria OpenCV*. URL: <https://opencv.org/>.
- [18] *LILA BC*. URL: <https://lila.science/>.
- [19] Lin. «Improving EEG-Based Emotion Classification Using Conditional Transfer Learning». In: *Frontiers in Human Neuroscience* (2017).
- [20] *Linguaggio Python*. URL: <https://www.python.org/>.
- [21] *Link Dataset Finale*. URL: <https://drive.google.com/file/d/1aseSzoDPdjycmDjEP6ntbhjhAgZESKA/view?usp=sharing>.
- [22] *mAP (mean Average Precision) - Hasty visionAI Wiki*. URL: <https://wiki.hasty.ai/metrics/map-mean-average-precision>.
- [23] *Missouri Camera Traps - LILA BC*. URL: <https://lila.science/datasets/missouricameratraps>.
- [24] *Ottimizzazione del modello / TensorFlow Lite*. URL: [https://www.tensorflow.org/lite/performance/model\\_optimization](https://www.tensorflow.org/lite/performance/model_optimization).
- [25] S. J. Pan e Q. A. Yang. «A survey on transfer learning». In: *IEEE Trans. Knowledge and Data Eng* (2010).
- [26] *Precision/Recall Tradeoff*. URL: <https://medium.com/analytics-vidhya/precision-recall-tradeoff-79e892d43134>.
- [27] *Raspberry Foundation*. URL: <https://www.raspberrypi.com/>.
- [28] *Raspberry Pi Official Camera Module V2 - 8MP - Melopero Electronics*. URL: <https://www.melopero.com/shop/raspberry-pi/boards/raspberrypiofficialcameramodulev28mp/>.
- [29] *Raspberry Pi OS – Raspberry Pi*. URL: <https://www.raspberrypi.com/software/>.
- [30] Farhadi A. Redmon J. «A. YOLOv3: An incremental improvement». In: *arXiv* (2018).
- [31] *Remote Desktop Protocol - Wikipedia*. URL: [https://en.wikipedia.org/wiki/Remote/Desktop\\_Protocol](https://en.wikipedia.org/wiki/Remote/Desktop_Protocol).
- [32] *Repository di GitHub per il codice del lavoro di Tesi*. URL: <https://github.com/labGISS/wild-animal-recognition-raspberry>.
- [33] *Repository GitHub di TFLite*. URL: [https://github.com/tensorflow/examples/tree/master/lite/examples/object\\_detection/raspberry\\_pi](https://github.com/tensorflow/examples/tree/master/lite/examples/object_detection/raspberry_pi).
- [34] *Repository GitHub di YOLOv5*. URL: <https://github.com/ultralytics/yolov5>.
- [35] *Roboflow*. URL: <https://roboflow.com/>.
- [36] *Secure Shell - Wikipedia*. URL: [https://en.wikipedia.org/wiki/Secure\\_Shell](https://en.wikipedia.org/wiki/Secure_Shell).
- [37] *Transfer Learning with Frozen Layers - YOLov5 Documentation*. URL: <https://docs.ultralytics.com/tutorials/transfer-learning-froze-layers/>.

## BIBLIOGRAFIA

---

- [38] *Virtual Environments in Python, documentation.* URL: <https://docs.python.org/3/tutorial/venv.html#introduction>.
- [39] *Virtual Network Computing - Wikipedia.* URL: [https://it.wikipedia.org/wiki/Virtual\\_Network\\_Computing](https://it.wikipedia.org/wiki/Virtual_Network_Computing).
- [40] C.Y. Wang. «CSPNet: A new backbone that can enhance learning capability of cnn». In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2020)* (2020).
- [41] K. Wang. «Few-shot image semantic segmentation with prototype alignment». In: *IEEE Conference on Computer Vision (ICCV 2019)* (2019).
- [42] *Weights & Biases.* URL: <https://wandb.ai/site>.
- [43] *Wildlife Detection System.* URL: [https://www.canadianconsultingengineer.com/cce/awards/2017/F7\\_WildlifeDetectionSystem.pdf](https://www.canadianconsultingengineer.com/cce/awards/2017/F7_WildlifeDetectionSystem.pdf).