UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA

RELAZIONE DEL PROGETTO D'ESAME

FONDAMENTI DI DATA SCIENCE E MACHINE LEARNING

Analisi Delle Tecniche Di Federated Learning Per l'Object Detection

Studenti:

Lorenzo Sorrentino
0522501849

Giovanni Borrelli
0522501807

Lorenzo Scorzelli
0522501811

ANNO ACCADEMICO 2023/2024

Sommario

1 INTRODUZIONE

1.1 Introduzione al problema e all'obiettivo del progetto

Lo scopo di questo progetto è implementare due differenti tipologie di Federated Learning, una che segue un approccio classico e una che segue un approccio innovativo, per confrontarle. La caratteristica fondamentale del federated learning è che, in questo processo di apprendimento, non si ha un solo modello, ma ogni dispositivo client addestra il proprio modello localmente e invia solo i pesi aggiornati a un modello centrale. Il modello centrale aggiorna le sue prestazioni basandosi su questi ultimi. Questo metodo non solo migliora le prestazioni del modello centrale, ma garantisce anche una maggiore privacy dei dati, poiché solo i pesi vengono trasferiti e non i dati grezzi.

I dataset utilizzati per l'addestramento contengono immagini di buche stradali.

1.2 Research questions individuate

RQ1: Quali sono i vantaggi in termini di privacy dei dati nell'utilizzo dell'apprendimento federato?

RQ2: Come varia la qualità del modello di rilevazione delle buche stradali in funzione del numero di immagini del dataset?

RQ3: Vale la pena utilizzare un approccio non casuale per la scelta dei client di cui combinare i pesi?

RQ4: Il federated learning può fornire risultati paragonabili ad un normale addestramento?

2 BACKGROUND E STATO DELL'ARTE

Una delle innovazioni più significative nel campo del machine learning decentralizzato è il concetto di **Federated Learning**, introdotto da McMahan et al. (2017). Questo approccio permette l'addestramento di modelli di deep learning utilizzando dati distribuiti su dispositivi mobili, senza la necessità di centralizzare i dati stessi. McMahan e colleghi hanno proposto l'algoritmo **FederatedAveraging**, che combina la discesa del gradiente stocastico locale su ciascun dispositivo con un server centrale che esegue la media degli aggiornamenti del modello. Gli esperimenti condotti hanno dimostrato che questo metodo è robusto a distribuzioni di dati squilibrate e non identicamente distribuite (non-IID) e può ridurre significativamente i round di comunicazione necessari per l'addestramento, fino a 100 volte rispetto ai metodi tradizionali di discesa del gradiente stocastico sincrono. Questo approccio non solo migliora l'efficienza della comunicazione, ma offre anche vantaggi significativi in termini di privacy e sicurezza dei dati, poiché i dati sensibili rimangono sui dispositivi locali e solo gli aggiornamenti del modello vengono trasmessi al server centrale. https://arxiv.org/abs/1602.05629

Un ulteriore sviluppo nel campo del Federated Learning è rappresentato dal lavoro di Li et al. (2020), che hanno introdotto il framework **FedProx** per affrontare le sfide della eterogeneità nei network federati. FedProx può essere visto come una generalizzazione e una ri-parametrizzazione di FedAvg, il metodo attualmente all'avanguardia per il federated learning. Questo framework

introduce un termine prossimale nell'obiettivo di ottimizzazione, che migliora la stabilità del metodo in presenza di eterogeneità sia a livello di sistema che a livello statistico. Teoricamente, FedProx offre garanzie di convergenza quando si apprendono dati da distribuzioni non identiche e consente a ciascun dispositivo partecipante di eseguire una quantità variabile di lavoro, in base alle proprie limitazioni di sistema. Praticamente, è stato dimostrato che FedProx consente una convergenza più robusta rispetto a FedAvg in una serie di dataset federati realistici, migliorando l'accuratezza dei test di un 22% in media in contesti altamente eterogenei.
https://arxiv.org/pdf/1812.06127v5

Recentemente, Deng et al. (2020) hanno proposto l'algoritmo **Adaptive Personalized Federated Learning (APFL)**, che mira a superare le limitazioni dei modelli globali standard nel contesto di dati altamente eterogenei. L'approccio APFL consente a ciascun cliente di allenare il proprio modello locale contribuendo contemporaneamente al modello globale. Gli autori derivano il bound di generalizzazione per la combinazione di modelli locali e globali e trovano il parametro di miscelazione ottimale. Inoltre, propongono un metodo di ottimizzazione efficiente in termini di comunicazione per apprendere collaborativamente i modelli personalizzati e ne analizzano la convergenza sia in contesti convessi fortemente lisci che non convessi. Gli esperimenti estensivi dimostrano l'efficacia dello schema di personalizzazione proposto, nonché la correttezza delle teorie di generalizzazione stabilite. Questo metodo permette di migliorare la generalizzazione rispetto ai modelli globali appresi da algoritmi come FedAvg e SCAFFOLD, specialmente in presenza di eterogeneità statistica tra i dati locali dei vari clienti. https://arxiv.org/pdf/2003.13461v3

2.1 Analisi dei principali approcci esistenti

Analizzando i principali approcci esistenti nel campo del Federated Learning, emerge un elemento comune: la scelta dei client per l'addestramento avviene in modo casuale. In particolare, come evidenziato nei lavori di McMahan et al. (2017), Li et al. (2020) e Deng et al. (2020), viene selezionato un sottoinsieme casuale di client a cui inviare i pesi del modello globale per l'aggiornamento. Questo approccio, sebbene semplice e facilmente implementabile, non tiene conto delle variazioni nella qualità dei dati e delle capacità computazionali dei diversi client. Nessuno di questi studi ha sperimentato metodi alternativi basati su metriche di precisione per la selezione dei client, come l'utilizzo di una o più metriche che possano meglio indirizzare il processo di addestramento.

Questo limite rappresenta un'opportunità per migliorare ulteriormente l'efficacia e l'efficienza del Federated Learning. Pertanto, abbiamo deciso di focalizzarci su questo aspetto, proponendo un metodo che utilizza una metrica di precisione specifica per la object detection, la mAP (mean Average Precision), come criterio per la selezione dei client. Questa metrica permette di valutare accuratamente le prestazioni dei modelli di object detection e può potenzialmente migliorare la qualità complessiva del modello globale; quindi, nel nostro caso, risulta essere la migliore candidata per tale scopo. I dettagli riguardo la mAP saranno discussi successivamente nel paragrafo 3.4.1, dove approfondiremo come quest’ultima viene calcolata e utilizzata nel contesto del nostro framework di Federated Learning.

3 METODOLOGIA

3.1 Descrizione del dataset

Il dataset utilizzato è reperibile al seguente link: Version (roboflow.com)

Il dataset proviene dal sito Roboflow ed è costituito da 655 immagini. Il dataset originale era destinato alla segmentazione delle immagini. Tuttavia, per il nostro progetto avevamo bisogno di un dataset per il rilevamento degli oggetti (Object Detection). Per trasformare il dataset di segmentazione in uno adatto per il rilevamento degli oggetti, abbiamo utilizzato la seguente strategia: per ogni oggetto segmentato, abbiamo calcolato la Bounding Box, prendendo come coordinate i valori più esterni della segmentazione. In questo modo, ogni oggetto segmentato è stato convertito in una Bounding Box che lo racchiude completamente.

3.2 TFRecord e YOLOv8

TFRecord è un formato di file semplice ma efficiente per memorizzare una sequenza di dati binari, usato comunemente con TensorFlow. Questo formato facilita l'archiviazione e il caricamento di grandi dataset in modo rapido e compatto, rendendo più agevole il processo di addestramento dei modelli di machine learning. I file TFRecord contengono esempi strutturati che possono includere vari tipi di dati, come immagini, testo o audio, e sono progettati per essere letti e scritti in modo sequenziale, ottimizzando così le prestazioni di input/output durante l'addestramento.

YOLOv8 è una versione avanzata dell'algoritmo YOLO (You Only Look Once) per il rilevamento di oggetti in tempo reale. Mantiene la filosofia di YOLO di essere veloce e accurato, migliorando ulteriormente le prestazioni e la precisione grazie a tecniche di ottimizzazione moderne e architetture di rete più sofisticate. YOLOv8 è in grado di rilevare e classificare oggetti in immagini e video con alta efficienza, rendendolo ideale per applicazioni che richiedono risposte rapide e accurate, come la sorveglianza, la guida autonoma e l'analisi video in tempo reale.

Per la fase di augmentation si è deciso di utilizzare il dataset in formato TFRecord, che risulta essere un formato idoneo per questo tipo di operazione, successivamente i dataset risultanti dalla fase di augmentation sono stati convertiti in formato YOLOv8 che risulta il più adatto per i modelli di Object Detection.

3.3 Augmentation

Poiché 655 immagini potrebbero non essere abbastanza per addestrare il modello, si è deciso di utilizzare tecniche di Augmentation, disponibili anche al seguente link: https://www.kaggle.com/code/parulpandey/overview-of-popular-image-augmentation-packages

Ad ogni immagine sono state applicate 4 tecniche di augmentation:

- **Flip orizzontale casuale (random flip left-right)**: Con una certa probabilità ribalta l'immagine in orizzontale.
- **Modifica casuale della luminosità (random brightness)**: Modifica la luminosità dell'immagine in modo casuale entro un intervallo definito.

- **Modifica casuale del contrasto (random contrast)**: Alterazione casuale del contrasto dell'immagine entro un intervallo specificato.
- **Modifica casuale della saturazione (random saturation)**: Variazione casuale della saturazione dell'immagine entro un intervallo definito.

Mentre le ultime 3 tecniche non hanno causato modifiche alla Bounding Box, si è dovuta fare più attenzione per la prima tecnica, poiché il ribaltamento causa inevitabilmente il cambio delle coordinate della Bounding Box. Tramite calcoli matematici, si è quindi fatto anche il ribaltamento della Bounding Box.
Sono stati generati due ulteriori dataset, uno da 1000 immagini e un altro da 1500 immagini che comprendono sia le immagini del dataset originale più quelle generate dalla fase di augmentation.

3.4 Federated Learning

Il Federated Learning è un approccio decentralizzato all'addestramento di modelli di machine learning che permette di mantenere i dati localmente sui dispositivi degli utenti, preservando la privacy e riducendo la necessità di trasferire grandi quantità di dati a un server centrale. Il processo di Federated Learning può essere riassunto in tre passaggi fondamentali: inizialmente, un modello globale viene inviato ai dispositivi dei client; successivamente, ogni client aggiorna il modello utilizzando i propri dati locali; infine, le modifiche ai modelli locali vengono inviate al server centrale, dove vengono aggregate per aggiornare il modello globale. Questo ciclo viene ripetuto fino a che il modello raggiunge le performance desiderate.

Nel nostro lavoro, ci siamo concentrati su un approccio differente per la selezione dei client, che solitamente avviene in modo casuale. Invece, abbiamo scelto di selezionare i client sulla base della metrica della mean Average Precision (mAP), utilizzando i migliori client in base a questa metrica per la combinazione dei pesi. La scelta della mAP come criterio di selezione mira a migliorare l'efficacia e l'efficienza dell'addestramento del modello, privilegiando i client che contribuiscono maggiormente alla qualità del modello globale.

3.4.1 Mean Average Precision

La mean Average Precision (mAP) è una metrica chiave utilizzata per valutare le performance di modelli di object detection, poiché fornisce un'indicazione precisa della capacità del modello di identificare correttamente e localizzare oggetti all'interno di immagini. La mAP è composta da due componenti principali: precision e recall.

- **Precision**: la precision è la frazione di previsioni corrette tra tutte le previsioni fatte. Si calcola come il numero di veri positivi (True Positives, TP) diviso per la somma dei veri positivi e dei falsi positivi (False Positives, FP):
$$Precision = \frac{TP}{TP + FP}$$
- **Recall**: il recall è la frazione di oggetti correttamente rilevati tra tutti gli oggetti presenti nell'immagine. Si calcola come il numero di veri positivi diviso per la somma dei veri positivi e dei falsi negativi (False Negatives, FN):
$$Recall = \frac{TP}{TP + FN}$$

La mAP combina questi due parametri per offrire una misura complessiva delle performance del modello. Il processo di calcolo della mAP comprende i seguenti passaggi:

1. **Calcolo della Precision-Recall Curve**: per ogni classe di oggetti, si calcola la precision e il recall a vari livelli di soglia di confidenza del modello. Questo genera una curva precision-recall per ciascuna classe.
2. **Calcolo dell'Average Precision (AP)**: l'AP per ogni classe viene calcolata come l'area sotto la curva precision-recall. Questo può essere fatto utilizzando diverse tecniche di interpolazione. Una comune è l'interpolazione dei 11 punti standard (dove la precisione viene calcolata in corrispondenza di 11 valori di richiamo standard: 0.0, 0.1, 0.2, ..., 1.0). L'AP è quindi la media delle precisioni massime a ciascun punto di richiamo:
$$AP = \frac{1}{11} \sum_{r \in \{0.0, 0.1, \ldots, 1.0\}} P_{interp}(r)$$
dove $P_{interp}(r)$ è la precisione interpolata a un dato livello di richiamo r.
3. **Calcolo della mean Average Precision (mAP)**: la mAP è la media degli AP calcolati per tutte le classi considerate. Se ci sono C classi, la mAP si calcola come:
$$mAP = \frac{1}{C} \sum_{c=1}^{C} AP_c$$

La mAP è una metrica estremamente utile nel campo della object detection perché offre una misura aggregata che considera sia la precision che il recall, fornendo una valutazione globale delle performance del modello in vari scenari di rilevamento. Questo la rende una scelta ideale per guidare la selezione dei client nel Federated Learning, assicurando che i client con le migliori capacità di rilevamento contribuiscano in modo più significativo all'aggiornamento del modello globale, migliorando così l'efficacia complessiva del sistema.

3.4.2 Implementazione

Per l'implementazione del Federated Learning, abbiamo utilizzato la piattaforma Google Colab, sfruttando il linguaggio di programmazione Python e il modello YOLOv8 per l'addestramento.

Abbiamo deciso di simulare i diversi client che partecipano al Federated Learning piuttosto che utilizzare macchine fisiche o virtuali separate. Questa decisione è stata presa per evitare l'overhead inutile di gestione e costi associati alla configurazione di più dispositivi. La simulazione dei client su una singola piattaforma ci ha permesso di mantenere il controllo completo sull'ambiente di addestramento e di concentrare le risorse computazionali, senza sacrificare la qualità e l'efficacia del progetto. Sebbene l'utilizzo di macchine separate avrebbe potuto ridurre il tempo complessivo di addestramento, la simulazione si è rivelata sufficiente per gli scopi della nostra ricerca.

Per l'addestramento del modello, abbiamo utilizzato una macchina virtuale su Google Cloud Engine dotata di GPU Nvidia L4, 8 vCPU e 32 GB di RAM. Questa configurazione ci ha fornito la potenza di calcolo necessaria per gestire il carico computazionale del modello YOLOv8, noto per la sua efficienza e accuratezza nella object detection.

4 VALUTAZIONE SPERIMENTALE

4.1 Experimental Settings

Sono state effettuate tre tipologie differenti di addestramento:

- Senza federated learning;
- Con federated learning, scegliendo randomicamente 3 client su 5 di cui combinare i pesi, seguendo la metodologia classica (denominata *random*);
- Con federated learning, in cui vengono scelti i 3 client migliori su 5 in base alla metrica di mAP (denominata *best mAP*).

Gli addestramenti senza federated learning sono stati effettuati su tre differenti dataset: il primo composto da 655 immagini, il secondo da 1000 e il terzo da 1500 immagini; questi ultimi sono stati ottenuti tramite tecniche di augmentation applicate al primo dataset che sono state illustrate nel paragrafo 3.3. Invece, gli addestramenti con federated learning sono stati fatti direttamente sui dataset da 1000 e 1500 immagini, dato che si è ritenuto che il dataset originale da 655 immagini non contenesse sufficienti dati da poter distribuire adeguatamente fra i client. Per gli addestramenti federati, i dataset sono stati divisi equamente fra i client; questi hanno a disposizione diverse partizioni del dataset di training, ma condividono lo stesso dataset di validazione, in modo tale da ottenere metriche di performance comparabili.

Gli addestramenti federati sono stati svolti in 5 epoche globali, in cui ciascun client si è addestrato per 20 epoche, mentre l'addestramento standard è stato fatto in 100 epoche.

Per ogni addestramento, fra i risultati vengono riportati:

- Le metriche di performance: Precision, mAP, Recall e F1 Score;
- I grafici contenenti: precision-confidence curve, recall-confidence curve, f1-confidence curve e precision-recall curve;
- Un insieme di immagini, facenti parte del validation set, che permettono di osservare le prestazioni di detection del modello.

Fra le metriche di performance dei modelli non viene riportata l'Accuracy, non essendo una metrica utilizzabile nel campo della object detection, data l'impossibilità di ottenere un valore di True Negative Count (TN), al suo posto viene utilizzata la metrica di mAP.

4.2 Caratteristiche dei dataset

Tramite i seguenti grafici possiamo ottenere diverse informazioni riguardanti la composizione dei dataset.

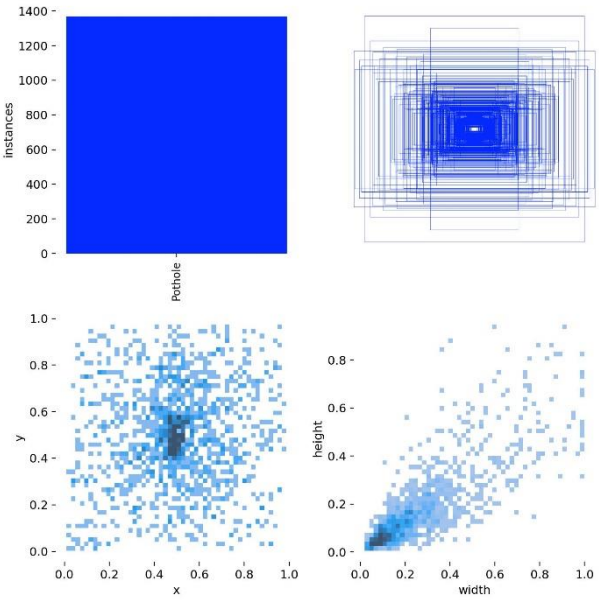


Figura 1. Dataset 655

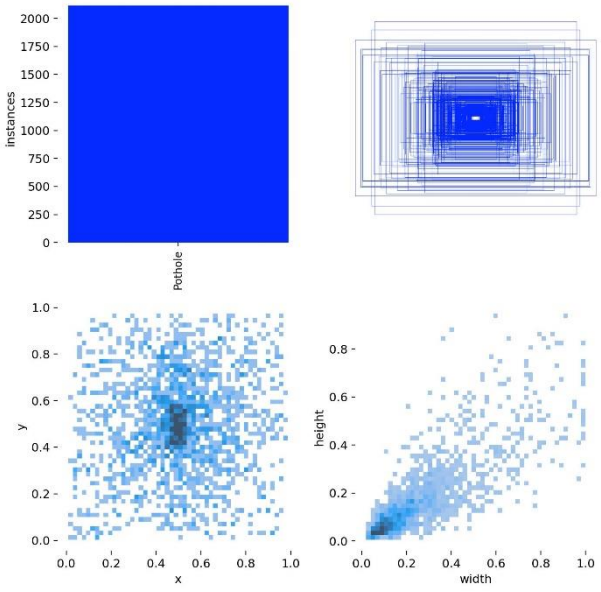


Figura 2. Dataset 1000

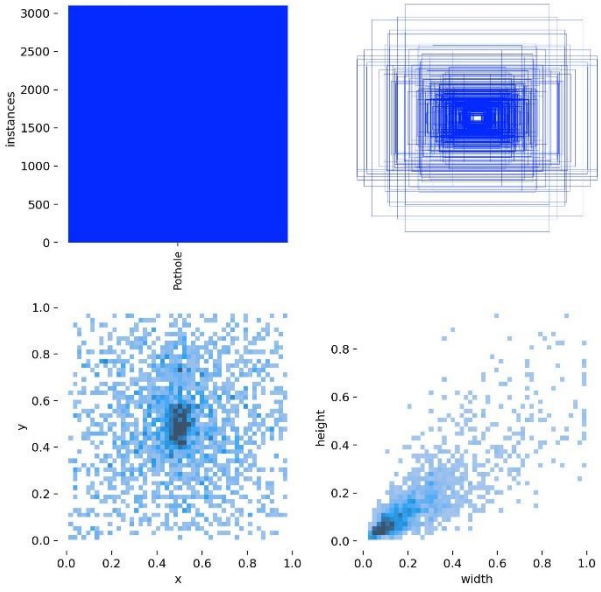


Figura 3. Dataset 1500

I grafici sopra riportati sono di quattro tipologie differenti:

- **Bar Plot (in alto a sinistra)**: Mostra la quantità di istanze della classe "Pothole". Questo ci permette di capire quante istanze di buche sono presenti nel dataset. Possiamo notare che, grazie all'augmentation, siamo passati dall'avere circa 1400 istanze nel dataset di partenza a circa 3000 nel dataset da 1500 immagini, ottenendo più del doppio delle istanze.

- **Box Plot (in alto a destra)**: Mostra la distribuzione delle coordinate di delimitazione (bounding box) delle buche secondo le loro dimensioni di larghezza e altezza. Da quello che possiamo notare, in tutti e tre i dataset sono più frequenti le buche di piccole dimensioni.
- **Scatter Plot di x vs y (in basso a sinistra)**: Mostra la distribuzione delle coordinate x e y delle bounding box, questo ci dice dove le buche sono più comunemente posizionate nell'immagine. Vediamo una distribuzione più densa verso il centro.
- **Scatter Plot di larghezza vs altezza (in basso a destra)**: Mostra la relazione tra la larghezza e l'altezza delle bounding box, indicando le dimensioni delle buche presenti. La maggior parte delle buche ha una larghezza e un'altezza tra 0.0 e 0.5. Possiamo notare, inoltre, che le bounding box tendono ad avere dimensioni simili fra larghezza e altezza, assumendo forme tendenti a dei quadrati.

4.3 Risultati e grafici

In questa sezione vengono riportati i risultati ottenuti dalle diverse tipologie di addestramento, organizzati sulla base delle Research Questions individuate nel paragrafo 1.2.

4.3.1 RQ1

Il federated learning offre dei vantaggi importanti in termini di privacy, infatti questo sfrutta un'architettura *privacy-by-design*. Per questo motivo ci siamo interrogati sull'efficacia che il federated learning abbia nel raggiungere questo scopo, chiedendoci:

Quali sono i vantaggi in termini di privacy dei dati nell'utilizzo dell'apprendimento federato?

Il nostro progetto non si concentra sul migliorare le funzionalità riguardanti la privacy, per questo abbiamo deciso di proporre una discussione riguardante un paper scientifico che tratta questo argomento nel dettaglio. Il paper, ideato da Jyoti Maurya e Shiva Prakash (https://ieeexplore.ieee.org/document/10266081), analizza le problematiche di privacy e sicurezza associate all'apprendimento federato, paradigma che promuove la collaborazione tra dispositivi senza la necessità di centralizzare i dati. Questo approccio si distingue dai modelli tradizionali, che raccolgono e aggregano i dati in un server centrale, sollevando preoccupazioni significative riguardo alla sicurezza e alla privacy.

Nel contesto del Federated Learning, la privacy degli utenti è preservata attraverso tecniche innovative che limitano la divulgazione di informazioni personali. Tra queste, la *secure aggregation* garantisce che solo aggiornamenti aggregati dei modelli vengano trasmessi al server centrale, piuttosto che i dati grezzi degli utenti. Inoltre, il *local training* consente di mantenere i dati sui dispositivi degli utenti, riducendo la necessità di trasferire dati sensibili attraverso la rete.

Fra le principali minacce alla privacy che possono insorgere in un ambiente federato vi sono: gli *inference attack* che possono permettere agli attaccanti di dedurre informazioni sensibili, analizzando le informazioni scambiate tra i dispositivi e il server; gli attacchi di *model extraction*, che possono compromettere la sicurezza del modello stesso; infine, gli attacchi di *data reconstruction* puntano ad ottenere informazioni sui dataset di training.

Per contrastare queste minacce, sono discusse diverse strategie difensive: la *Differential Privacy*, che introduce rumore ai dati per proteggere l'identità degli utenti, rendendo difficile la ricostruzione dei dati originali; la *Secure Multiparty Computation*, la quale consente la collaborazione tra

partecipanti senza rivelare informazioni private; la *Homomorphic Encryption*, invece, permette di eseguire operazioni sui dati criptati, mantenendo la privacy durante il calcolo; infine, il *Trusted Execution Environment* offre uno spazio sicuro per eseguire il codice, proteggendo i dati da accessi non autorizzati.

In conclusione, il federated learning offre diversi vantaggi rispetto all'apprendimento tradizionale, allo stesso tempo però, esistono diverse tecniche per comprometterne la sicurezza. Per questo motivo è un campo in continua evoluzione, per permettere l'addestramento di modelli di machine learning rispettando le sempre più stringenti leggi sulla privacy.

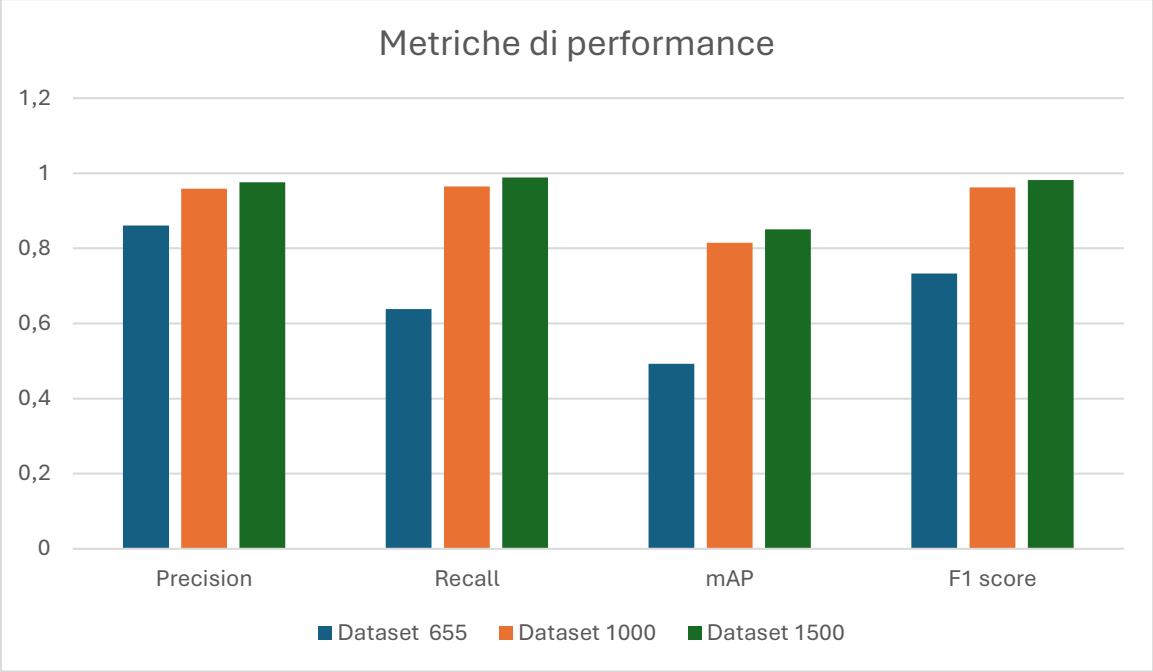
4.3.2 RQ2

In questo caso la domanda che ci poniamo è:

Come varia la qualità del modello di rilevazione delle buche stradali in funzione del numero di immagini del dataset?

Per rispondere a questa domanda andremo ad analizzare i risultati degli addestramenti effettuati sui tre differenti dataset, senza federated learning.

Metriche di performance



Dai risultati ottenuti, si osserva chiaramente che l'aumento del numero di immagini nel dataset porta a miglioramenti significativi nelle performance del modello di object detection. Il dataset con 1500 immagini mostra i migliori risultati in tutte le metriche di performance, suggerendo che un maggior numero di dati di addestramento permette al modello di generalizzare meglio e di essere più accurato e affidabile. Tuttavia, i benefici dell'aumento del numero di immagini tendono a ridursi man mano che il dataset diventa più grande, come indicato dagli incrementi relativamente minori tra il dataset di 1000 e quello di 1500 immagini.

Precision-Confidence curve

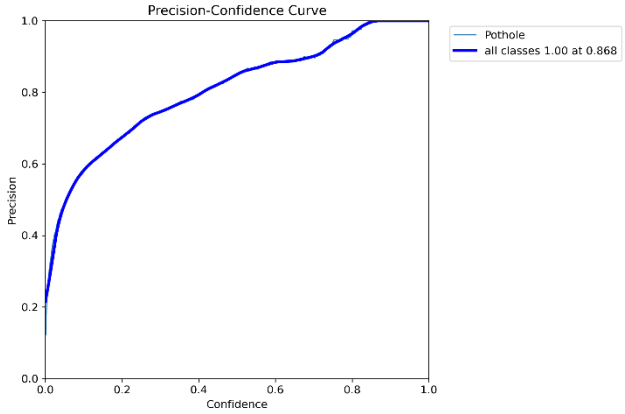


Figura 4. P-C curve dataset 655

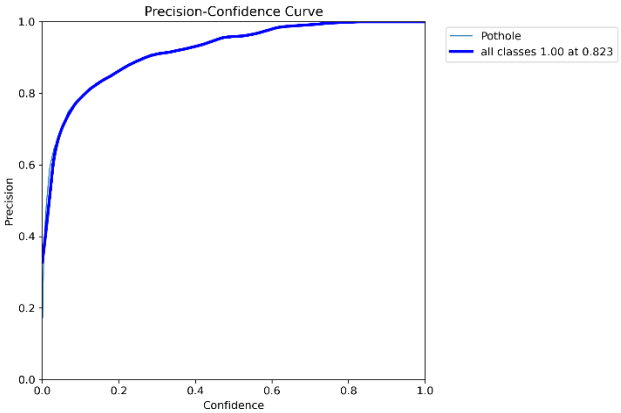


Figura 5. P-C curve dataset 1000

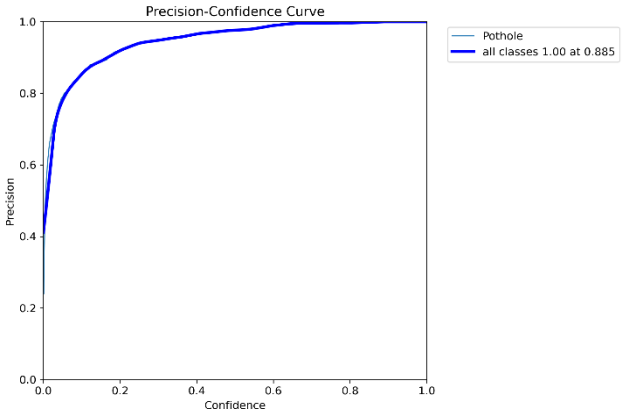


Figura 6. P-C curve dataset 1500

Le precision-confidence curve per i tre dataset mostrano come varia la precisione del modello in funzione del livello di confidenza. La capacità del modello di raggiungere una precisione perfetta a livelli di confidenza più bassi migliora con l'aumento del numero di immagini di addestramento. Il modello con 1000 immagini richiede una confidenza minore rispetto al modello con 655 immagini per raggiungere una precisione di 1.0. Tuttavia, con 1500 immagini, il livello di confidenza richiesto è leggermente più alto, ma l'andamento generale della curva suggerisce una maggiore affidabilità complessiva. Queste curve confermano le osservazioni fatte in precedenza sulle metriche di performance, mostrando che l'aumento del numero di dati di addestramento porta a un modello di object detection più accurato e affidabile.

Recall-Confidence curve

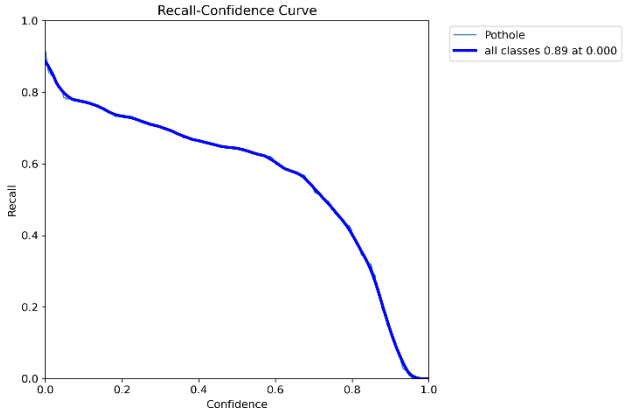


Figura 7. R-C curve dataset 655

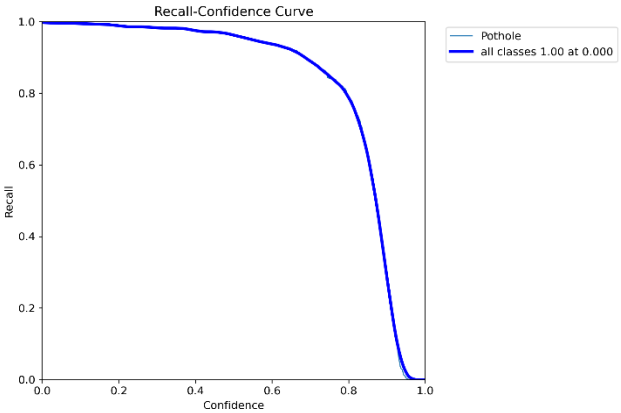


Figura 8. R-C curve dataset 1000

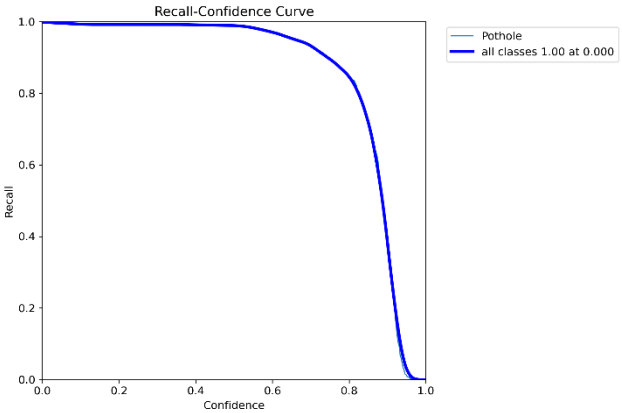


Figura 9. R-C curve dataset 1500

Le recall-confidence curve mostrano come varia il recall del modello in funzione del livello di confidenza. Il modello migliora nella capacità di mantenere un recall elevato con l'aumento del numero di immagini di addestramento. Con 1000 e 1500 immagini, il modello raggiunge un recall di 1.00 a confidenza 0, mostrando un miglioramento rispetto al dataset di 655 immagini (recall 0.89 a confidenza 0). Inoltre, il modello con 1000 e 1500 immagini riesce a mantenere un recall elevato per una gamma più ampia di livelli di confidenza rispetto al modello con 655 immagini. Questo indica una maggiore affidabilità e capacità di identificare correttamente gli oggetti anche con soglie di confidenza più elevate. Il modello addestrato con 1500 immagini è il più affidabile, mantenendo un recall elevato anche per valori di confidenza maggiori. Questo suggerisce che con un maggior numero di immagini di addestramento, il modello è più robusto e meno incline a perdere rilevamenti corretti; anche se le prestazioni cambiano di poco rispetto al modello addestrato con 1000 immagini.

F1-Confidence curve

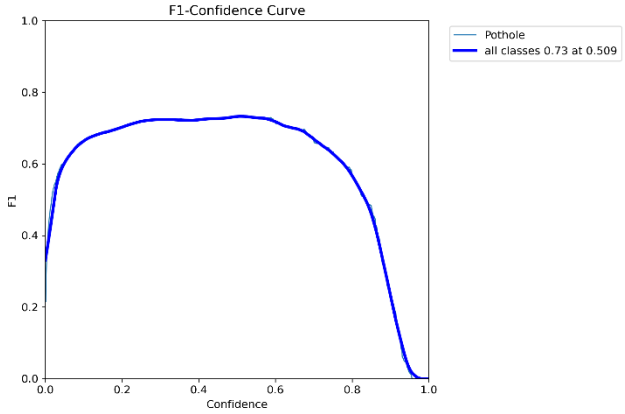


Figura 10. F1-C curve dataset 655

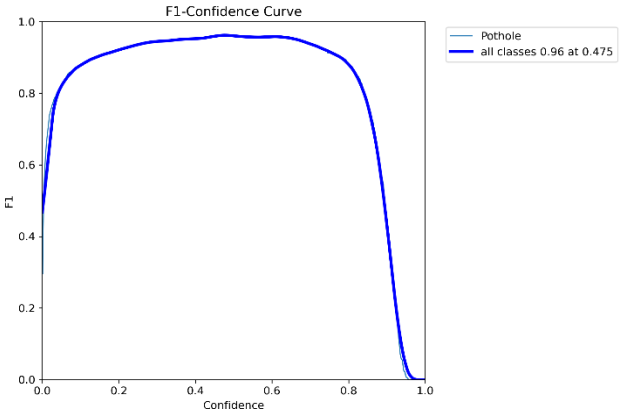


Figura 11. F1-C curve dataset 1000

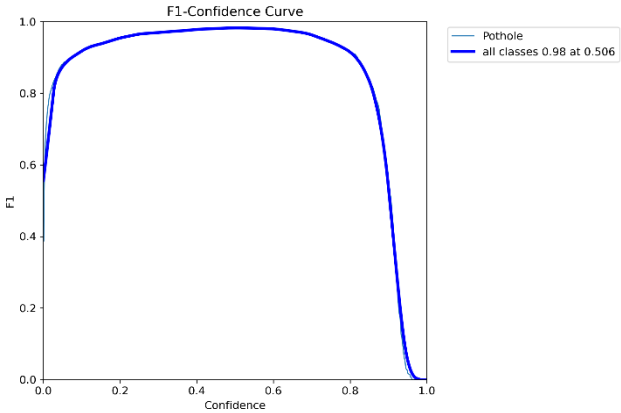


Figura 12. F1-C curve dataset 1500

Questa curva mostra l'F1-Score di un modello di classificazione al variare della soglia di confidenza. L'F1-Score è la media armonica di precision e recall. La curva consente di osservare come l'equilibrio tra precision e recall cambia al variare della soglia di confidenza. Un F1-Score elevato a valori di confidenza specifici indica che il modello bilancia bene precisione e richiamo per quella soglia. Tutti e tre i modelli presentano un valore di confidence ottimale intorno allo 0.5. Le prestazioni del modello migliorano significativamente con più dati di addestramento, suggerendo che quest’ultimo trae vantaggio da dataset più grandi.

Precision-Recall curve

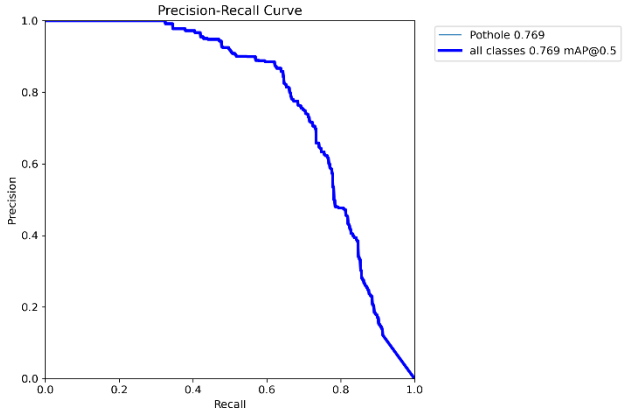


Figura 13. P-R curve dataset 655

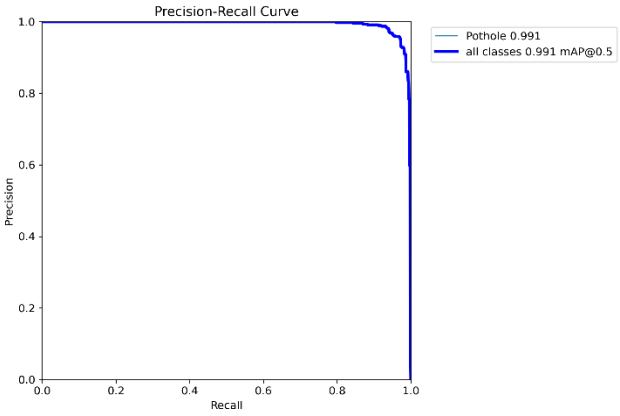


Figura 14. P-R curve dataset 1000

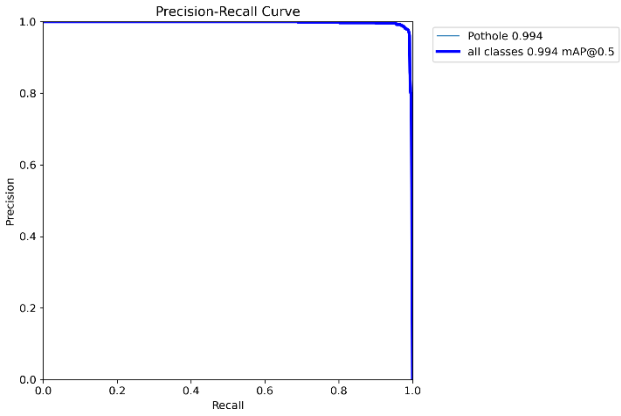


Figura 15. P-R curve dataset 1500

La curva di precision-recall è un grafico che mostra il compromesso tra precision e recall per diversi valori di soglia: come la precision diminuisce quando il recall aumenta. Una curva che si avvicina all'angolo superiore destro indica un buon modello (alta precision e alto recall). Come possiamo osservare, i modelli addestrati su set di dati più grandi mantengono una precision più elevata su tutti i livelli di recall. Date le prestazioni così elevate sui dataset da 1000 e 1500 immagini, non si esclude la possibilità di overfitting.

Esempi di detection

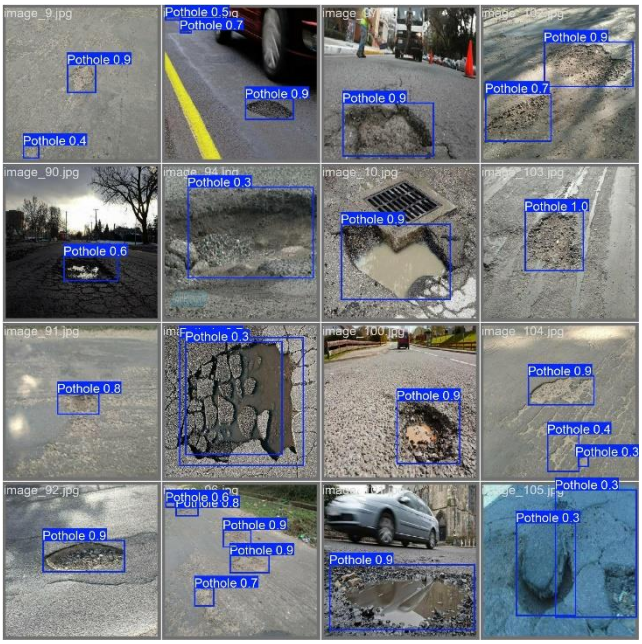


Figura 16. Detection dataset 655

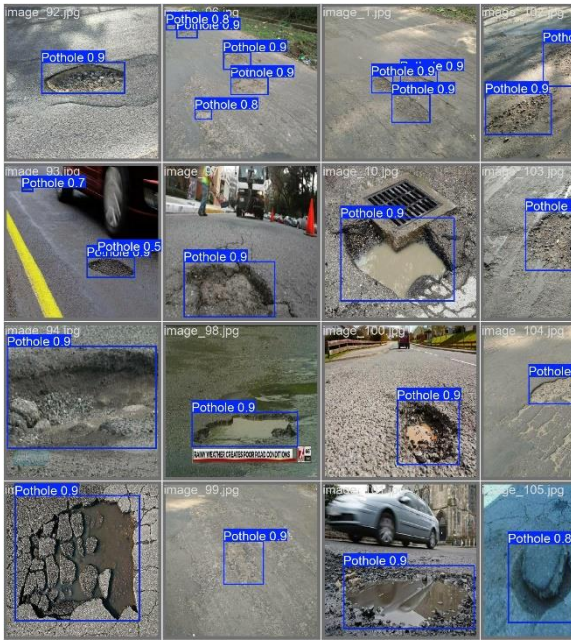


Figura 17. Detection dataset 1000

Figura 18. Detection dataset 1500

Dalle immagini d'esempio possiamo osservare ciò che è stato chiaro lungo il corso di tutta l'analisi, ovvero che le prestazioni del modello addestrato con dataset da 1500 immagini risultano le migliori.

Conclusioni

La qualità del modello varia notevolmente applicando tecniche di augmentation al dataset originale. Queste ultime permettono al modello di fare addestramento su più immagini, ottenendo metriche di valutazione molto più elevate, quindi delle prestazioni nettamente migliori nella detection. Nei casi di apprendimento standard analizzati, c'è poca differenza di prestazioni fra i dataset da 1000 e 1500 immagini; questo risultato cambia col federated learning, come si può osservare successivamente.

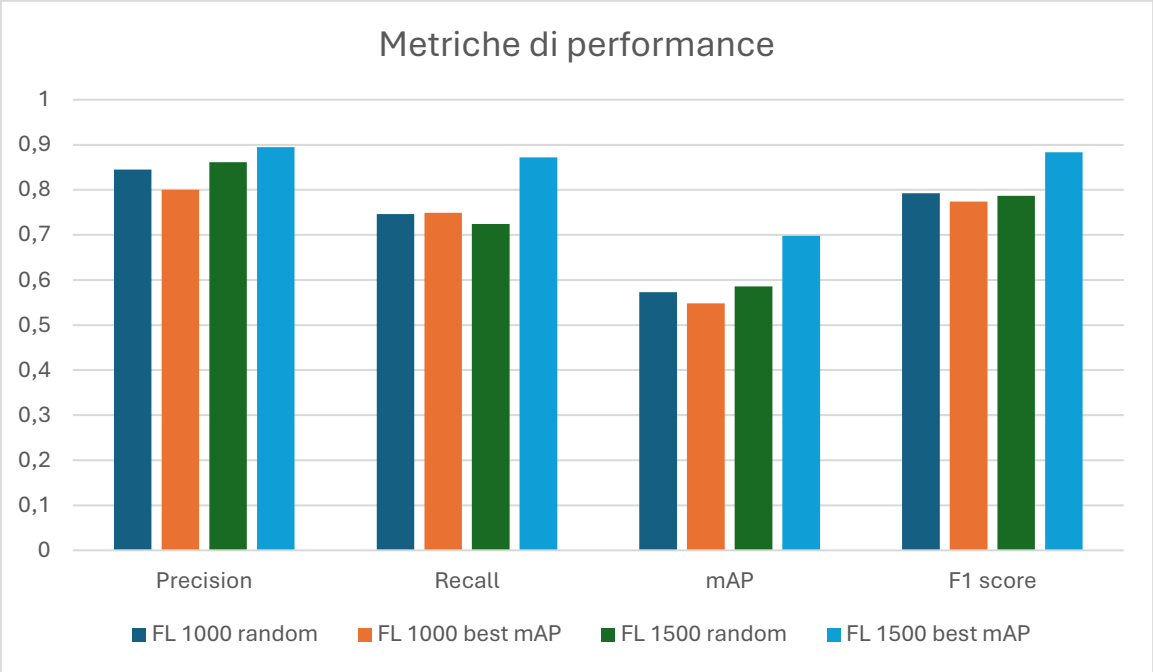
4.3.3 RQ3

I risultati che andremo ad analizzare in questo paragrafo sono presentati per rispondere alla seguente domanda:

Vale la pena utilizzare un approccio non casuale per la scelta dei client di cui combinare i pesi?

Per rispondere a questa domanda, andremo ad analizzare i risultati ottenuti da due differenti tipologie di addestramento federato: la prima segue il criterio di scelta dei client in base alle mAP, mentre la seconda utilizza la strategia randomica. Entrambi gli approcci sono stati testati sui dataset da 1000 e 1500 immagini, per un totale di quattro addestramenti federati.

Metriche di performance



Analizzando le metriche di performance dei diversi modelli, possiamo notare che non vi sono notevoli differenze fra i primi tre, in qualsiasi metrica considerata. Quello che possiamo chiaramente notare è che, in tutte le metriche, il modello *FL 1500 best mAP* supera chiaramente gli altri, indicando che un dataset più grande e la selezione ottimale dei client contribuiscono a performance superiori. Da questa prima analisi, quindi, non possiamo affermare che il metodo *best mAP* performa meglio del *random* in tutte le situazioni, ma nel caso si abbiano a disposizione un gran numero di dati mostra evidenti vantaggi; a differenza di *FL 1500 random* che non riesce a migliorare i suoi risultati nonostante il maggior numero di dati a disposizione.

Precision-Confidence curve

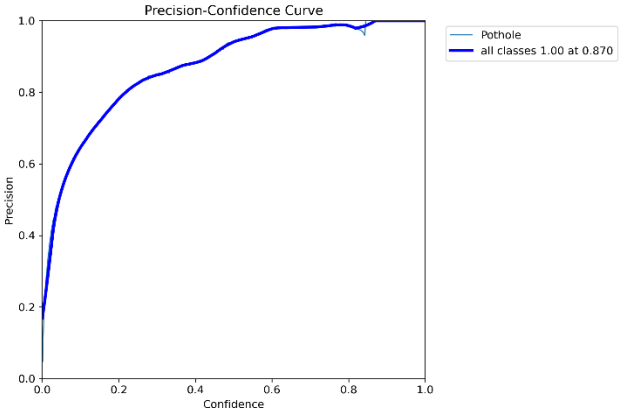


Figura 19. P-C curve FL 1000 random

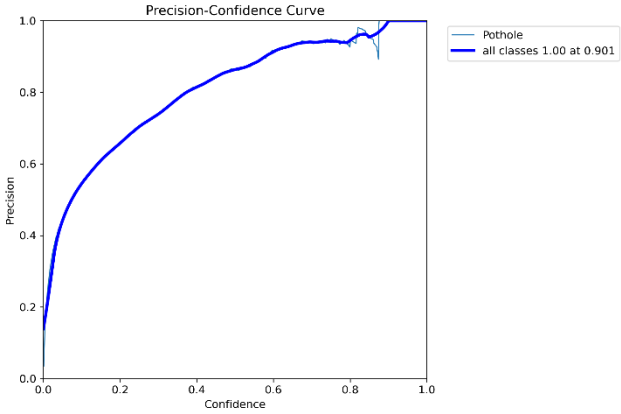


Figura 20. P-C curve FL 1000 best mAP

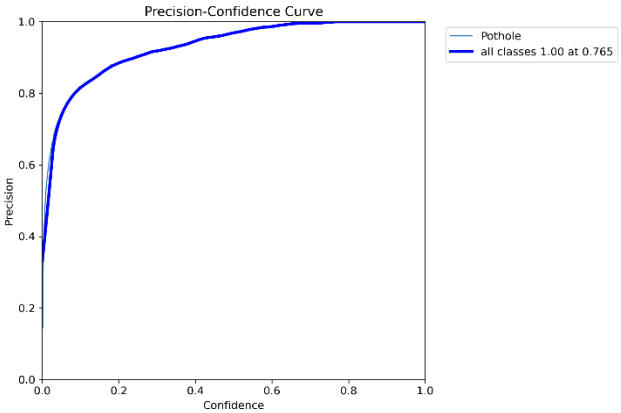


Figura 21. P-C curve FL 1500 random

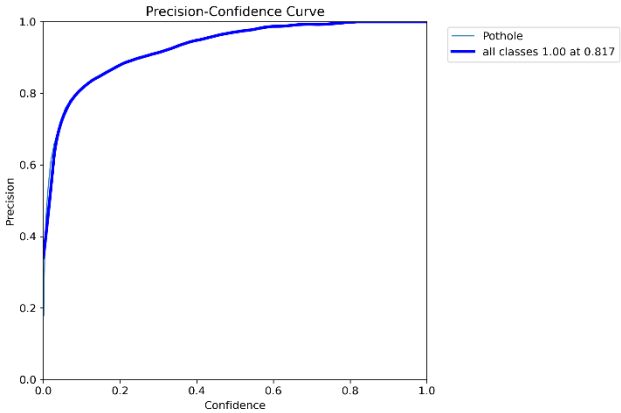


Figura 22. P-C curve FL 1500 best mAP

I grafici sopra riportati ci permettono di analizzare più a fondo la precisione dei modelli. I risultati descritti sono analoghi a quelli che abbiamo potuto già osservare nel grafico sintetico, però vengono evidenziate meglio le differenze. Tutti e quattro i modelli forniscono buone prestazioni e, a parità di dataset, le curve risultano essere molto simili. Da ciò che si può evincere, nessuna delle due metodologie prevale sull'altra e vi è un risultato anche leggermente controverso, ovvero che, secondo questa metrica, il metodo *best mAP* fornisce sia il modello che performa meglio sia quello che performa peggio. Inoltre, in linea con ciò che abbiamo potuto osservare rispondendo alla RQ2, notiamo un netto miglioramento delle curve all'aumentare del dataset.

Recall-Confidence curve

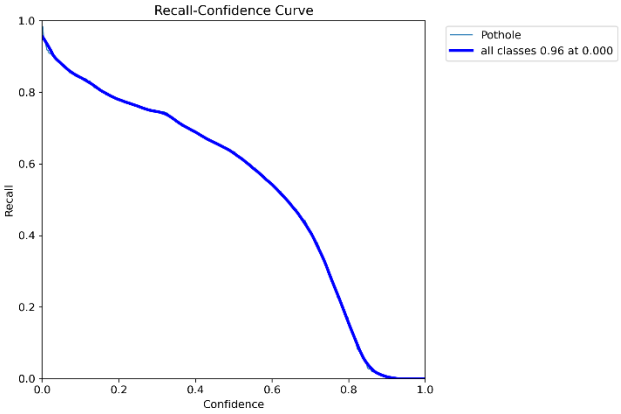


Figura 23. R-C curve FL 1000 random

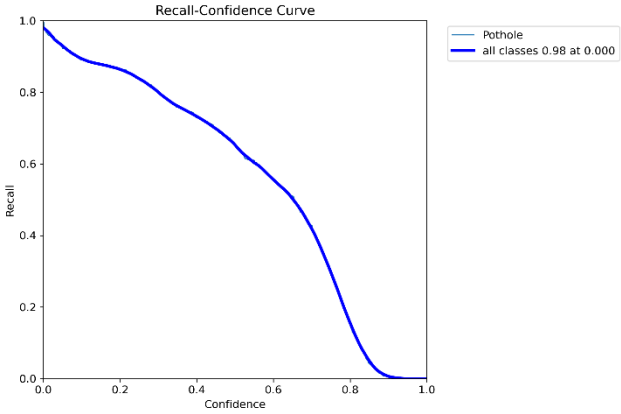


Figura 24. R-C curve FL 1000 best mAP

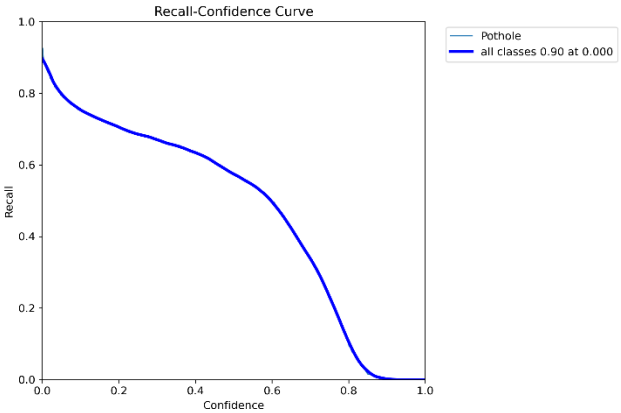


Figura 25. R-C curve FL 1500 random

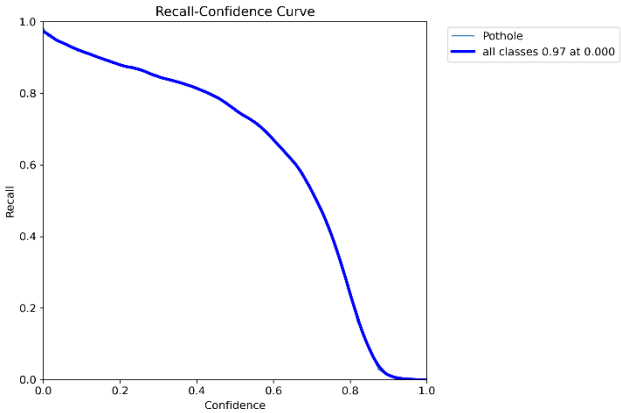


Figura 26. R-C curve FL 1500 best mAP

Analizzando i grafici di recall-confidence, la prima cosa che salta all'occhio è che tutti e quattro i modelli forniscono buone prestazioni. Osservando più in dettaglio, però, possiamo notare che i primi tre modelli hanno prestazioni molto simili fra loro. Questo denota una differenza poco marcata fra i metodi *random* e *best mAP* addestrando il modello sul dataset da 1000 immagini, ma allo stesso tempo denota una capacità nulla del modello *FL 1500 random* di sfruttare il dataset più grande; quest'ultimo, infatti, ha anche un calo di prestazioni rispetto agli altri due. Il modello *FL 1500 best mAP* è quello che fornisce il risultato migliore, discostandosi di molto dagli altri. Da questa analisi, quindi, si evince che il metodo *best mAP* fornisce prestazioni migliori.

F1-Confidence curve

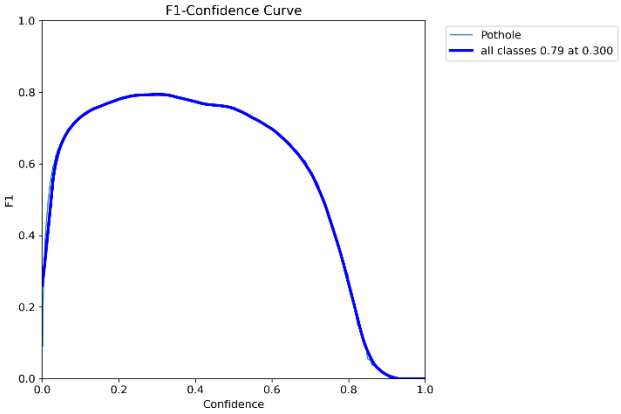


Figura 27. F1-C curve FL 1000 random

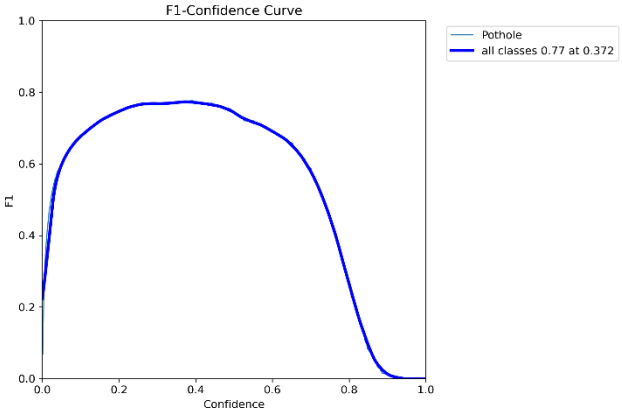


Figura 28. F1-C curve FL 1000 best mAP

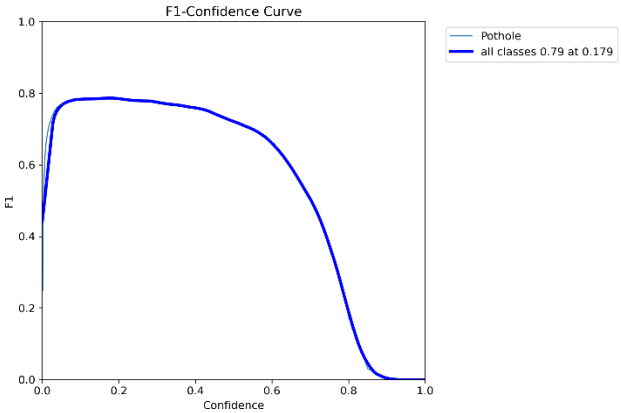


Figura 29. F1-C curve FL 1500 random

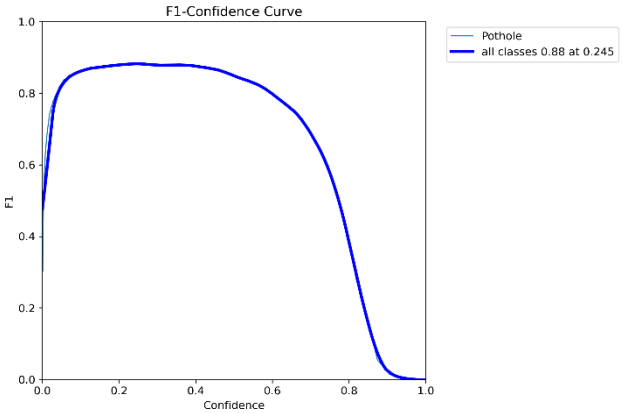


Figura 30. F1-C curve FL 1500 best mAP

I grafici di F1-confidence riportati sopra descrivono una situazione molto simile a quelli di recall-confidence. Tutti e quattro i modelli presentano buone prestazioni, ma l'ultimo si discosta molto, presentando prestazioni nettamente migliori rispetto agli altri. I primi tre modelli hanno prestazioni molto simili fra loro, denotando una differenza poco marcata fra i metodi *random* e *best mAP* su dataset da 1000 immagini, ma una scarsa capacità di *FL 1500 random* di sfruttare a pieno il dataset più grande. Quest'ultimo presenta prestazioni di poco migliori rispetto agli altri due, infatti ha un picco leggermente più basso rispetto a *FL 1000 random*, ma presenta valori migliori a livelli di confidence bassi. Come detto prima, il modello *FL 1500 best mAP* è quello che presenta il risultato migliore, discostandosi di molto dagli altri, perciò si evince che il metodo *best mAP* fornisce prestazioni migliori e riesce a sfruttare al meglio il dataset più grande.

Precision-Recall curve

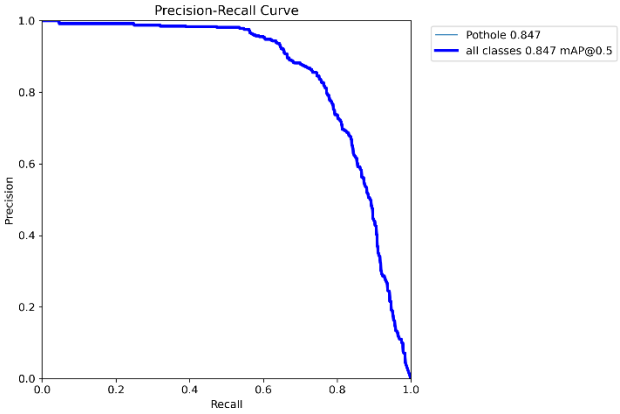


Figura 31. P-R curve FL 1000 random

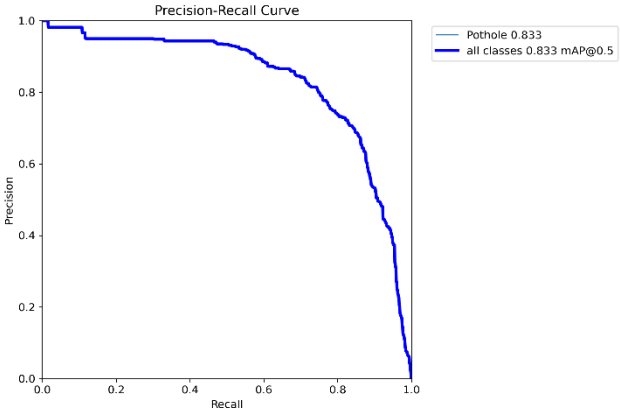


Figura 32. P-R curve FL 1000 best mAP

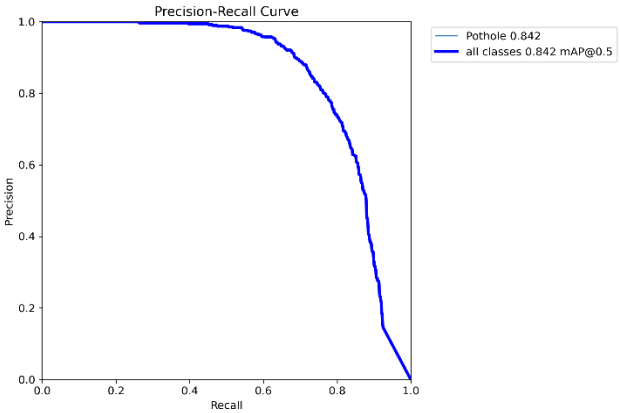


Figura 33. P-R curve FL 1500 random

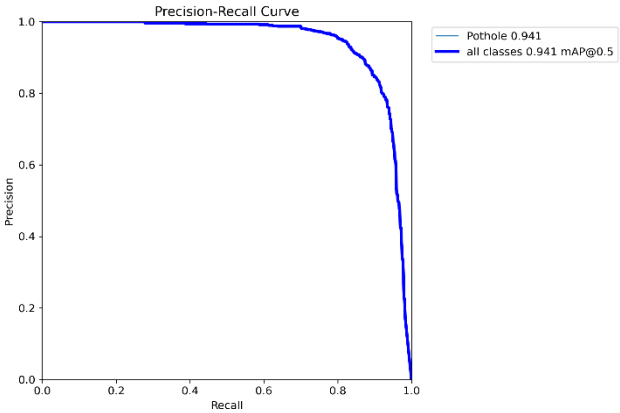


Figura 34. P-R curve FL 1500 best mAP

I grafici di precision-recall sopra riportati confermano il trend che abbiamo già osservato in precedenza, ovvero i primi tre modelli hanno prestazioni simili, mentre il terzo è nettamente migliore. Anche in questo caso tutti e quattro i modelli forniscono buone metriche di performance. Come già osservato in precedenza, quindi, non si evincono nette differenze fra i primi due modelli; perciò, i due metodi hanno prestazioni simili sul dataset da 1000 immagini (il primo ha una partenza migliore, ma il secondo mantiene valori più alti di precision a valori più alti di recall). *FL 1500 random* mostra, ancora una volta, di sfruttare male il dataset di dimensioni maggiori, ottenendo delle performance anche leggermente inferiori rispetto a *FL 1000 random*. *FL 1500 best* si conferma definitivamente il migliore dei quattro, mostrando delle performance di gran lunga superiori agli altri tre, confermando la bontà del metodo *best mAP*, soprattuto nello sfruttare il dataset da 1500 immagini.

Esempi di detection

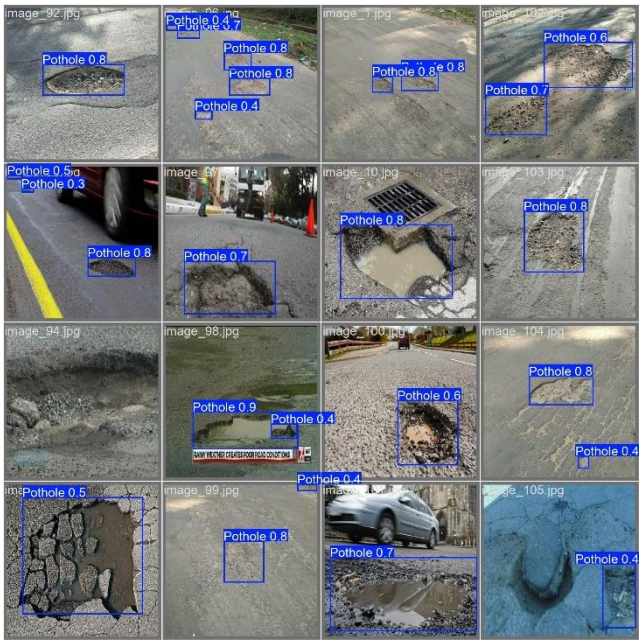


Figura 35. Detection FL 1000 random

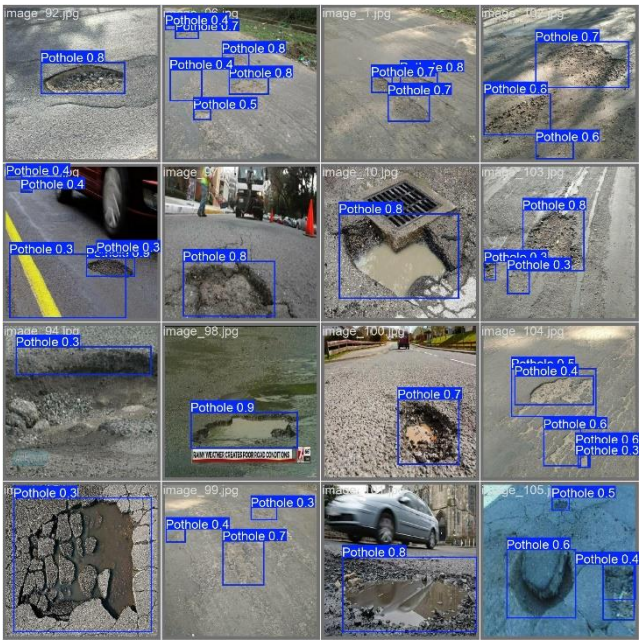


Figura 36. Detection FL 1000 best mAP

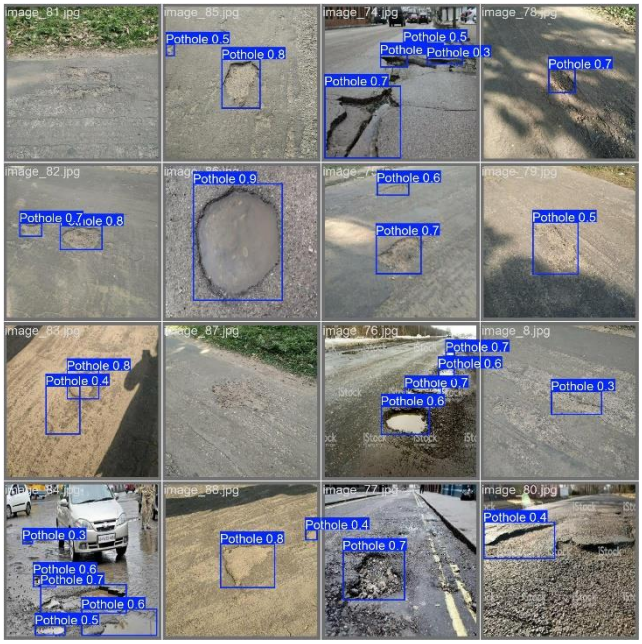


Figura 37. Detection FL 1500 random

Figura 38. Detection FL 1500 best mAP

Questi esempi di detection ci permettono di osservare che, fra i primi due, *FL 1000 random* presenta meno falsi positivi, però riconosce meno buche rispetto a *FL 1000 best mAP*. Fra i due addestrati sul dataset da 1500, invece, notiamo che *FL 1500 best mAP* non presenta falsi positivi e identifica correttamente più buche rispetto a *FL 1500 random*. Queste immagini confermano ciò che abbiamo già potuto osservare dai grafici, ovvero che i primi tre hanno prestazioni paragonabili, mentre l'ultimo è nettamente migliore.

Conclusioni

Dai risultati ottenuti possiamo affermare che scegliere i client sulla base dei valori di mAP risulta in un modello mediamente migliore rispetto a sceglierli su base casuale. Infatti, è emerso che l'addestramento col dataset da 1000 immagini ha portato a risultati paragonabili dei due modelli. Invece, nel caso del dataset da 1500 immagini, il modello risultante ha prestazioni nettamente migliori con scelta basata sulla mAP. Quindi, possiamo affermare che, se l'architettura lo permette, è meglio scegliere sulla base delle mAP; perché può fornire risultati paragonabili, se non nettamente migliori, soprattutto con dataset di dimensioni maggiori.

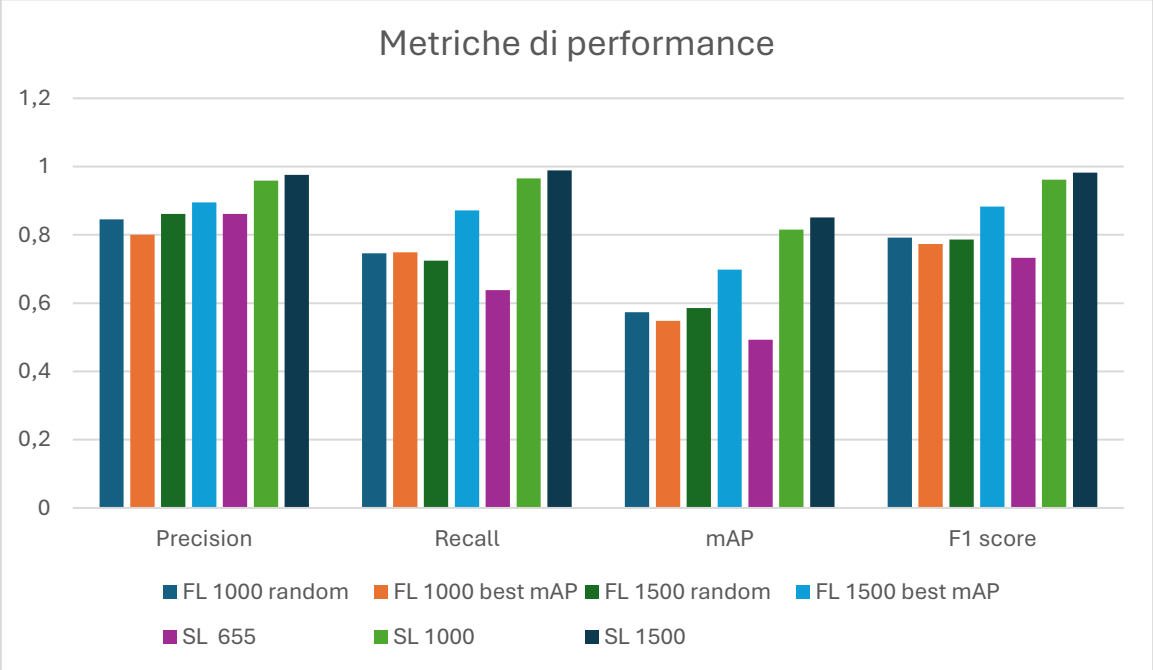
4.3.4 RQ4

Come ultima Research Question, abbiamo deciso di mettere a confronto l'apprendimento classico e il federated learning, quindi ci chiediamo:

Il federated learning può fornire risultati paragonabili ad un normale addestramento?

In questo caso, a differenza delle precedenti Research Question, abbiamo deciso di affidarci solamente al grafico sintetico delle metriche di performance, per il confronto. Questo perché gli altri grafici sono stati ampiamente discussi in precedenza e non aggiungerebbero informazioni significative per rispondere a quest'ultima domanda. Il confronto sarà effettuato fra tutti i differenti addestramenti, sia federati che non. Con FL si identificano gli addestramenti effettuati con federated learning, con SL si intendono gli addestramenti standard.

Metriche di performance



Dai risultati ottenuti, il federated learning risulta avere delle prestazioni abbastanza inferiori rispetto all'apprendimento classico, a parità di dataset e con una durata di addestramento simile. Infatti, se confrontiamo i valori delle mAP possiamo osservare che mediamente c'è uno scarto in negativo del 23% rispetto all'addestramento classico. Detto ciò, il federated learning rimane lo stesso una

tipologia di addestramento valida, dato che riesce comunque a fornire dei buoni risultati, al contempo soddisfando anche i requisiti di privacy come riportato nella RQ1.

Per rendere questo confronto plausibile si è scelto di effettuare un addestramento federato in 5 epoche globali in cui ciascun client si è addestrato per 20 epoche, mentre l'addestramento standard è stato fatto in 100 epoche.

Inoltre, possiamo notare che il federated learning ottiene un grande aumento di prestazioni passando dal dataset da 1000 a quello da 1500, utilizzando il metodo *best mAP*. Questo sottolinea che beneficia di un dataset più grande molto più di quanto non faccia l'addestramento normale. Da quello che possiamo osservare, quindi, è facile dedurre che il modello addestrato col federated learning possa beneficiare di un dataset ancora più grande, a differenza del modello addestrato col metodo standard che già presenta delle metriche di performance molto elevate.

4.4 Conclusioni e sviluppi futuri

In conclusione, si è rivelato fondamentare fare augmentation del dataset, poiché questo ha fatto sì che l'addestramento del modello raggiungesse delle metriche di qualità più elevate. Soprattutto il dataset da 1500 immagini si è rivelato fondamentale per confrontare le performance delle due tecniche di Federated Learning *random* e *best mAP*, mostrando le maggiori potenzialità della seconda.

Confrontando i risultati ottenuti dai diversi approcci al federated learning, possiamo affermare che scegliere randomicamente i 3 client si è rivelato il peggiore, invece, scegliere i 3 client migliori in base alla metrica di mAP si è rivelata una strategia più efficace. In termini di performance, a parità di dataset e con una durata di addestramento simile, l'apprendimento federato ha mostrato risultati peggiori rispetto a quello standard. Bisogna, però, sottolineare che nonstante le performance inferiori, abbiano comunque ottenuto delle performance soddisfacenti e che il federated learning offre notevoli miglioramenti riguardo gli aspetti di privacy.

Come possibili sviluppi futuri, si potrebbe migliorare ulteriormente il modello federato con l'ampliamento del dataset, avendo a disposizione più immagini o immagini più variegate; si potrebbero, inoltre, sperimentare ulteriori tecniche di scelta dei client sulla base di criteri diversi. Infine, sarebbe interessante approfondire la differenza fra le due metodologie con dataset non equamente distribuiti fra i client.