Università degli studi di
Milano-Bicocca

Advanced Machine Learning

Final Project

Classificazione multi-oggetto sul dataset PASCAL VOC 2012

Authors:
Riccardo Andena - 859643 - r.andena@campus.unimib.it
Marini Fabio - 851977 - f.marini14@campus.unimib.it

February 19, 2024

Abstract

Questo progetto si focalizza sulla classificazione di oggetti presenti nelle immagini del dataset Pascal VOC 2012. È stato adottato un approccio di suddivisione delle immagini in porzioni in base alla posizione degli oggetti al loro interno per l'addestramento dei modelli di machine learning. Successivamente, attraverso svariate operazioni di addestramento, sono stati valutati diversi modelli al fine di identificare quello più adatto al compito specifico. Un aspetto cruciale di questo task è stato il processo di predizione sulle immagini di test. È stato identificato un algoritmo ottimizzato per individuare e classificare tutti gli oggetti presenti nelle varie porzioni delle immagini, con l'obiettivo di massimizzare l'accuratezza e la completezza delle predizioni. Infine, i risultati ottenuti sono stati confrontati attraverso predizioni effettuate su dati reali, evidenziando le variazioni nelle predizioni dei modelli in base alla natura e alla quantità dei dati di addestramento utilizzati.

1 Introduzione

La realizzazione di questo progetto si basa sul topic riguardante "Object Category Recognition" sul dataset Pascal VOC 2012 che è composto da una grande varietà di immagini raffiguranti determinati oggetti in diversi contesti. L' obiettivo è stato quello di sviluppare un sistema di classificazione in grado di determinare la presenza o l'assenza di uno o più oggetti all'interno delle immagini di test, assegnandoli a una delle venti classi presenti nel dataset. Per realizzare questo compito è stato adottato un approccio basato su Convolutional Neural Network (CNN) per l'elaborazione delle immagini per via dell'efficiente capacità nell'estrazione di caratteristiche visive significative. In aggiunta a questo sono stati utilizzati i metodi di transfer learning per sfruttare reti complesse pre-addestrate su dataset di maggiori dimensioni. Infine, per lo svolgimento corretto del compito in questione, si è ragionato su come trattare la classificazione di più oggetti all'interno delle immagini di test. Si è quindi deciso di utilizzare delle finestre scorrevoli sulle immagini in maniera tale da esaminarne diverse porzioni e gestire le diverse sovrapposizioni.

2 Datasets

Il dataset Pascal VOC 2012 per il task di classificazione è composto da immagini e file contenenti le cosiddette annotazioni, questi non sono altro che file .xml associati ad ogni immagine che contengono le informazioni relative alle coordinate dei riquadri all'interno delle immagini che identificano gli oggetti, i cosiddetti "Bounding Boxes".
Le classi che fanno parte di questo dataset sono 20 e sono: *person, bird, cat, cow, dog, horse, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, tv/monitor*.

2.1 Operazioni sul dataset

Partendo dalle immagini e dalle corrispondenti annotazioni è stato creato un file .csv che contiene, per ogni oggetto di ogni immagine, le informazioni riguardanti: nome dell'immagine, coordinate x e y del Bounding Box dell'oggetto con riferimento all'immagine ed infine la classe di appartenenza di quell'oggetto.

Table 1: Esempio di righe nel CSV

fileName	**xmin**	**ymin**	**xmax**	**ymax**	**class**
2007_000032.jpg	104	78	375	183	aeroplane
2007_000032.jpg	26	189	44	238	person

Questo perché il training della rete viene fatto utilizzando come input i singoli oggetti delle immagini e non l'immagine intera, quindi diventa importante avere traccia dell'oggetto in questione. Risulterà poi comodo andare a salvare le porzioni di immagini che contengono i singoli oggetti. Ciò che viene fatto in aggiunta per migliorare le predizioni è l'introduzione di una classe aggiuntiva, ovvero la classe *background* che svolge il ruolo di "classe negativa". Questo viene fatto andando a ritagliare porzioni di immagini del dataset originale che non contengono oggetti. Questo serve per dare la capacità al modello di distinguere le aree "non pertinenti" andando ad assegnare esplicitamente per certe porzioni l'etichetta di "non oggetto", in questo modo si può facilmente andare a riconoscere quando un'immagine non contiene nessun oggetto appartenente alle classi previste senza che il modello forzi la predizione di qualche classe. Da questo dataframe sono state poi tolte 100 istanze per classe che verranno utilizzate successivamente per calcolare alcune metriche riguardanti la bontà delle predizioni del modello per ogni classe.

2.1.1 Bilanciamento del dataset

Dopo aver effettuato le operazioni per tenere traccia degli oggetti presenti nelle immagini, si ottiene un dataset che risulta essere altamente sbilanciato in favore di alcune classi.
Per via dell'alta disparità è stato effettuato un taglio del numero degli oggetti presi in considerazione, questo viene fatto anche per velocizzare il modello durante la fase di addestramento. Sono stati eseguiti diversi tagli per trovare il numero di istanze per classi ottimale. Inizialmente veniva utilizzato un modello che sfruttava transfer learning andando ad escludere solamente gli ultimi livelli completamente connessi, con esso si era stabilito che il numero di istanze per classe ottimale fosse di 400 in quanto anche portando il numero ad un tetto massimo di 1200 per classe, le performance del modello non miglioravano. Quando invece è stato effettuato un taglio più profondo si è osservato come l'aumento del numero di dati portava ad un conseguente aumento delle performance. Avendo poi adottato il modello con questo tipo di taglio si è stabilito di bilanciare il dataset utilizzando un limite massimo di 1200 istanze per classe arrivando quindi ad avere la seguente situazione:

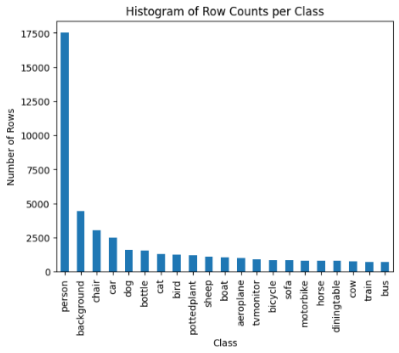

Figure 1: Distribuzione dataset originale

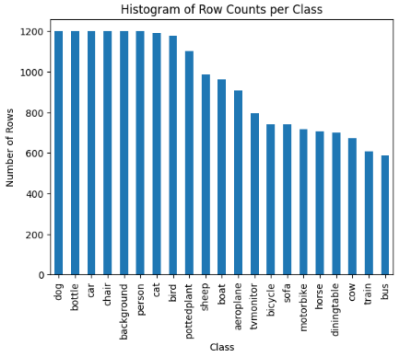

Figure 2: Distribuzione dataset bilanciato

Anche se le classi non sono perfettamente bilanciate, l'accuratezza per le classi con meno istanze, non risulta essere inferiore rispetto alle altre, inoltre anche avendo una situazione in cui tutte le classi hanno lo stesso numero di istanze è stato dimostrato che le performance del modello non ne beneficiano; perciò si è ritenuto di adottare questo tipo di bilanciamento.

2.1.2 Data augmentation

In aggiunta al dataset bilanciato, è stato creato un nuovo dataset su cui è fatta data augmentation sui dati andando a creare 100 nuove immagini per ogni classe. Il dataset è stato modificato, non con il bisogno di aumentare i dati, ma tenendo in considerazione la metodologia utilizzata durante la fase di testing che verrà analizzata nel dettaglio successivamente. Infatti, siccome la predizione viene fatta andando a prendere tante porzioni dell'immagine, si è scelto di creare nuovi dati cambiando solamente lo zoom dei campioni per cercare di capire se, così facendo, il modello fosse più sensibile a piccoli oggetti presenti nelle singole porzioni. Si è scelto solamente questo tipo di modifica anche considerando che, per ogni oggetto, sono già presenti numerosi campioni che lo rappresentano in svariate situazioni; perciò, ulteriori immagini con modifiche di rotazione, lumonistà etc. potrebbero risultare ridondanti.

3 Approccio metodologico

Dopo aver completato la fase di pre-elaborazione dei dati, si è proceduto con l'addestramento del modello. Numerosi esperimenti sono stati condotti al fine di ottenere performance soddisfacenti.

Inizialmente, è stata progettata e testata una rete neurale convoluzionale (CNN) personalizzata, attraverso una serie di iterazioni sperimentali, culminando in una accuracy del 52%. Data la performance non ottimale, si è optato per l'adozione di una strategia di transfer learning, valutando l'efficacia di architetture pre-addestrate note per la loro robustezza e capacità di generalizzazione, quali VGG19, ResNet50 e Inception V3.
L'applicazione di VGG19 non ha prodotto miglioramenti significativi, risultando in una precisione inferiore alle aspettative. ResNet50, nonostante la sua rinomata efficacia in numerosi contesti di visione artificiale, ha mostrato limitazioni in termini di velocità di elaborazione, rendendo impraticabile un'ampia sperimentazione data la sua onerosità computazionale.
L'integrazione di Inception V3, d'altro canto, ha segnato una svolta decisiva, elevando la precisione sul set di test al 87%. Questo notevole incremento evidenzia l'efficacia del transfer learning nel superare le limitazioni della CNN originariamente sviluppata, sfruttando la sofisticata architettura di Inception V3 che ha dimostrato di raggiungere un'accuratezza superiore al 78,1% sul

set di dati ImageNet e con l'integrazione dei moduli Inception, che aggregano convoluzioni di varie dimensioni, permette al modello di catturare dettagli a diverse scale, migliorando l'accuratezza della classificazione.

In una fase iniziale, è stato osservato che l'utilizzo di un dataset di training composto da 8400 istanze rispetto a uno da 19797 non comportava differenze significative in termini di prestazioni, ma solo in termini di tempo di elaborazione. Pertanto, considerando anche i limiti imposti dalla GPU su Colab, è stata privilegiata la scelta del dataset da 8400 istanze per l'addestramento. Durante lo sviluppo, il modello è stato modificato rimuovendo i fully connected layer esistenti e introducendo layer personalizzati. Per questi ultimi, sono state testate diverse configurazioni, variando il numero di layer, l'applicazione di dropout, la modifica del learning rate, e l'impiego di regolarizzatori. Dopo numerosi tentativi, è stata raggiunta una configurazione che ha permesso di ottenere una precisione del 77% sul set di test.
Successivamente, limitando l'intervento al solo layer di output di Inception V3 e sostituendolo con un layer personalizzato, si è ottenuto un incremento immediato della precisione, raggiungendo l'82% sul set di test. Ulteriori esperimenti con layer personalizzati non hanno tuttavia portato a miglioramenti significativi.

Un avanzamento notevole è stato conseguito modificando più radicalmente la struttura della rete, attraverso un taglio più profondo e l'aggiunta di un Average Pooling layer come primo layer personalizzato, replicando la configurazione proposta da Inception V3 nella sua sezione fully connected. È stato interessante notare che, in questo scenario, l'utilizzo del dataset ridotto a 8400 istanze ha leggermente abbassato le prestazioni rispetto al dataset più ampio da 19797 istanze a differenza di quanto osservato con le configurazioni precedenti. Questo fenomeno è stato attribuito alla necessità del modello, reso meno profondo, di disporre di un maggior numero di immagini per generalizzare e apprendere efficacemente. Pertanto è stato scelto di effettuare i training con il dataset da 19797 istanze. Dopo diverse iterazioni per ottimizzare la configurazione, è stata raggiunta una precisione dell'87% sul set di test, segnando il risultato finale del processo di sperimentazione.

Una considerazione finale riguarda la scelta della funzione di attivazione per il layer di output, valutando tra l'uso della sigmoide e della softmax. Tipicamente, per compiti di classificazione multi-label, dove più oggetti possono

essere identificati all'interno di un'unica immagine, la sigmoide sarebbe la scelta preferibile, consentendo di classificare più etichette contemporaneamente. Tuttavia, data la natura del processo di addestramento, che si concentra su singole istanze, e il metodo adottato per la valutazione e il testing del modello, è stata selezionata la funzione softmax. Questa decisione si basa sull'approccio di predizione che impiega una finestra scorrevole, analizzando solo porzioni dell'immagine di varie dimensioni, il che facilita la distinzione tra le diverse istanze. Questa metodologia, che sarà illustrata più dettagliatamente in seguito, permette di identificare con precisione anche le istanze sovrapposte nelle immagini.
Di seguito vengono mostrati i layer aggiuntivi rispetto al modello di inception V3 con taglio effettuato al layer 'mixed9':

Table 2: Elenco dei layer utilizzati nel modello

Layer	**Attivazione**
GlobalAveragePooling2D()	-
Dense(500)	ReLU
Dropout(0.5)	-
Dense(400)	ReLU
Dropout(0.5)	-
Dense(300)	ReLU
Dropout(0.5)	-
Dense(200)	ReLU
Dropout(0.5)	-
Dense(21)	Softmax

Una volta effettuato il training si è dovuto pensare ad un algoritmo per effettuare la rilevazione di più oggetti in un'immagine e per farlo si è scelto di usare la Sliding Window.

Questo metodo implica il movimento sequenziale di una finestra attraverso l'intera area di un'immagine. A ogni passo, la finestra cattura una porzione dell'immagine in cui viene effettuata la predizione degli oggetti. L'efficacia della Sliding Window si basa sulla sistematicità con cui esplora l'immagine: inizia dall'angolo superiore sinistro e si muove orizzontalmente fino al bordo destro, per poi scendere verticalmente di un passo e ripetere il processo. Questo assicura che ogni parte dell'immagine venga ispezionata. La dimen-

sione della finestra e la grandezza del passo sono parametri cruciali, poiché influenzano sia la risoluzione dell'analisi che l'efficienza computazionale del processo. Una finestra più grande può catturare oggetti più grandi, ma potrebbe perdere quelli più piccoli e richiedere più tempo per l'analisi; al contrario, una finestra più piccola aumenta la sensibilità ai piccoli oggetti, ma può portare a un numero eccessivo di analisi e a una maggiore probabilità di falsi positivi.

Si è però andati incontro al problema di rilevare oggetti di diverse dimensioni e siccome la dimensione della finestra è decisa a priori, c'era il rischio che alcuni non venissero presi in considerazione. Si è quindi scelto di utilizzare la tecnica della Pyramid Sliding Window. Quest'ultima invece di affidarsi a una singola finestra di dimensioni fisse che scorre attraverso l'immagine introduce un approccio gerarchico. Si inizia costruendo una serie di immagini ridimensionate, note come piramide d'immagini, ciascuna rappresentante l'immagine originale a una scala diversa. Questo insieme di immagini scalate viene poi esplorato utilizzando la metodologia della sliding window tradizionale rendendo in questo modo la dimensione della sliding windows più dinamica permettendo di considerare oggetti di diverse dimensioni.

4 Risultati e Valutazione

In questa sezione verranno mostrati i risultati del modello che ha ottenuto le performance migliori utilizzando il dataset di training bilanciato con 19797 istanze totali. In particolare verranno mostrati i risultati delle metriche di training e i risultati sulla predizione delle immagini a cui viene applicata la sliding window.

Di seguito, in figura 3 viene mostrato l'andamento delle metriche di loss e di accuracy durante il training. Per l'addestramento, il dataset bilanciato mostrato in precedenza è stato diviso tra training set e validation set rispettivamente dell'80% e 20%. In particolare alla fine delle 10 epoche si è ottenuta una training loss uguale a 0,371 e una validation loss uguale a 0,507. Per quanto riguarda l'accuracy è stata ottenuta una training accuracy uguale a 0,89 e una validation accuracy uguale a 0,87.

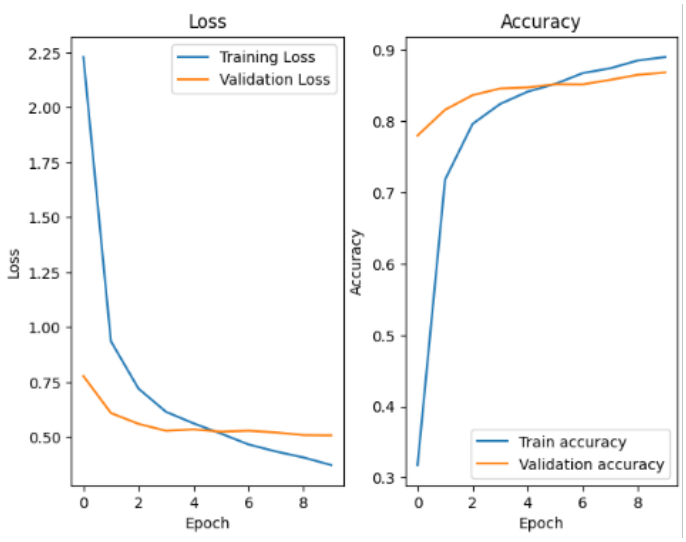

Figure 3: Grafico di loss e accuracy

In figura 4 sono state calcolate poi le metriche di precision, recall e f1 utilizzando un dataset composto da 100 istanze per classe estrapolate in una fase iniziale dal dataset originale.

	precision	recall	f1-score	support
aeroplane	0.97	0.89	0.93	100
background	0.94	0.92	0.93	100
bicycle	0.93	0.87	0.90	100
bird	0.96	0.89	0.92	100
boat	0.91	0.86	0.88	100
bottle	0.93	0.88	0.90	100
bus	0.93	0.89	0.91	100
car	0.92	0.78	0.84	100
cat	0.95	0.95	0.95	100
chair	0.71	0.75	0.73	100
cow	0.95	0.75	0.84	100
diningtable	0.94	0.67	0.78	100
dog	0.89	0.90	0.90	100
horse	0.92	0.92	0.92	100
motorbike	0.95	0.86	0.90	100
person	0.83	0.80	0.82	100
pottedplant	0.94	0.92	0.93	100
sheep	0.81	0.87	0.84	100
sofa	0.87	0.71	0.78	100
train	0.94	0.88	0.91	100
tvmonitor	0.98	0.90	0.94	100
micro avg	0.91	0.85	0.88	2100
macro avg	0.91	0.85	0.88	2100
weighted avg	0.91	0.85	0.88	2100
samples avg	0.85	0.85	0.85	2100

Figure 4: Metriche di classificazione

Infine, come ultima valutazione, in figura 5 viene mostrato il risultato della matrice di confusione ottenuto sul dataset usato per calcolare le metriche di classificazione. Questo permette di mostrare per una specifica "ground truth", le classi che il modello predice.

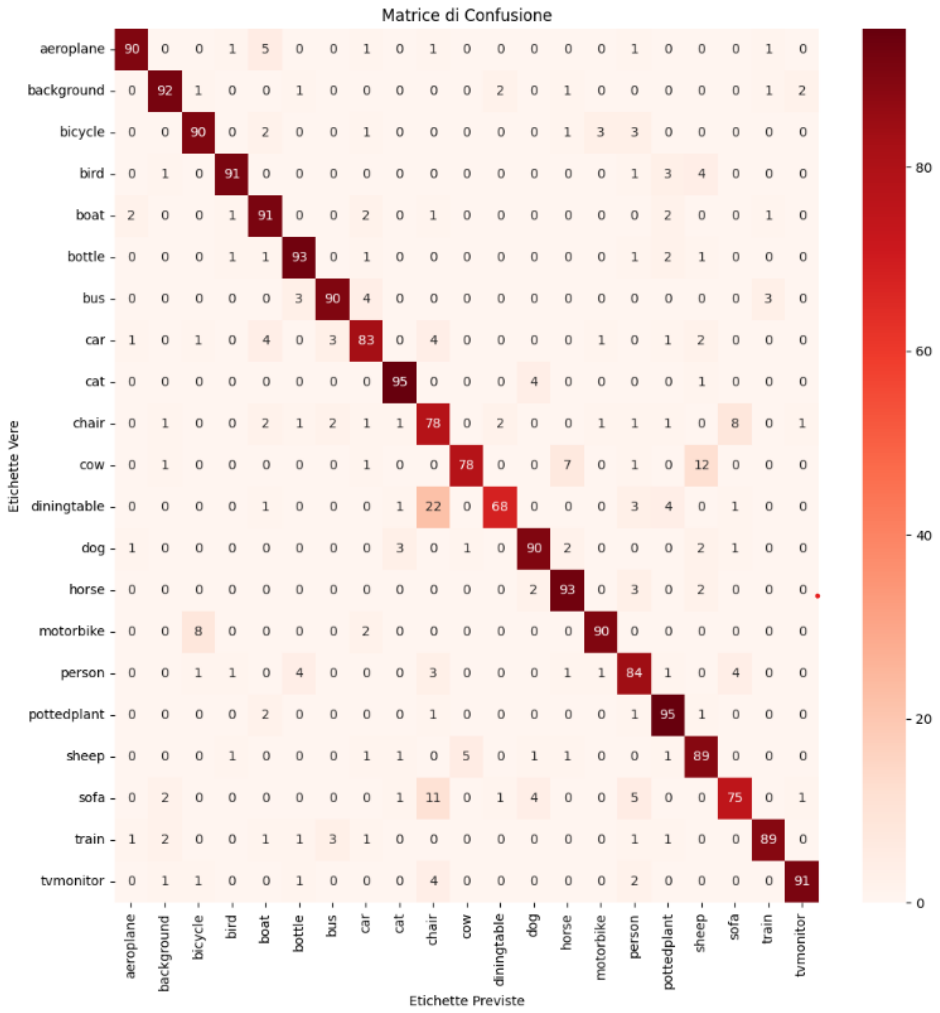

Figure 5: Matrice di confusione

4.1 Predizioni sulle immagini

Di seguito sono mostrate delle immagini su cui è stata effettuata l'output della predizione applicando la pyramid sliding window.

Figure 6: Cane su una sedia
Predizioni: dog, chair, sofa

Figure 7: Persone che mangiano
Predizioni: person, pottedplant, diningtable

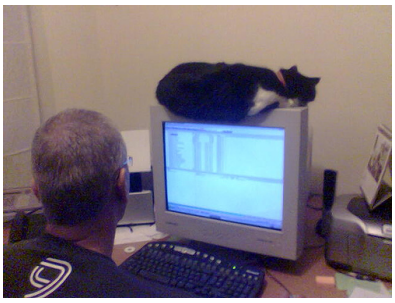

Figure 8: Persona davanti a monitor
Predizioni: person, tvmonitor, cat

Figure 9: Persona in macchina
Predizioni: car, boat, person

4.2 Predizioni con data augmentation

In questa sottosezione sono mostrati dei risultati utilizzando lo stesso modello descritto precedentemente ma utilizzando il dataset con data augmentation e valutare le differenze con il modello precedente a cui non viene applicata la data augmentation.

Figure 10: Strada con veicoli

Figure 11: Pecora con un cane

Table 3: Confronto predizioni

	Figura 10	**Figura 11**
No data augmentation	person, bicycle	dog, sheep
Data augmentation	person, bicycle, car, motorbike	dog, sheep, bird, cow

5 Discussion

Come illustrato nel quarto capitolo, le prestazioni del modello sono state valutate positivamente, con un'accuratezza sui dati di test che raggiunge l'87%. L'analisi delle metriche di accuracy e loss mostrata nella figura 3 rivela un incremento rapido dell'accuratezza durante la fase di addestramento, mentre l'accuratezza sulla validazione, sebbene in aumento nelle varie epoche, tende a stabilizzarsi dopo la quinta, suggerendo che ulteriori sessioni di addestramento potrebbero non migliorare in modo significativo la capacità del modello di generalizzare. La matrice di confusione in figura 5, evidenzia una corretta classificazione delle istanze nel set di test con una precisione superiore al 90% per diverse classi. Permette anche di osservare, ad esempio, che la classe *cow* viene identificata correttamente nel 78% dei casi, ma confusa con la classe *sheep* nel 12% dei casi, probabilmente a causa della somiglianza visiva tra i due oggetti.

L'analisi delle predizioni su immagini, mediante l'applicazione di finestre scorrevoli, mostra una discreta capacità di riconoscimento degli oggetti, nonostante occasionalmente vengano identificate istanze non presenti, come nel caso della figura 9, dove, oltre a *car* e *person*, viene erroneamente prevista anche una barca. Tuttavia, in situazioni con oggetti sovrapposti, come ad

esempio nella figura 6, il modello è in grado di distinguere correttamente elementi quali cane e sedia, anche utilizzando la funzione di attivazione softmax, grazie alla diversa prospettiva offerta dalle finestre scorrevoli che permette di analizzare separatamente gli oggetti.
Nel sottocapitolo 4.2 viene infine effettuato un confronto tra i risultati ottenuti addestrando il modello con un dataset non arricchito da immagini modificate e uno che include tali immagini, con particolare attenzione a quelle ingrandite per migliorare il riconoscimento degli oggetti durante l'uso delle finestre scorrevoli. Sebbene le metriche di base rimangano simili tra i due dataset, la fase di predizione mostra alcune differenze: il modello addestrato con anche data augmentation tende a identificare un numero maggiore di oggetti presenti nelle immagini, come mostrato nella figura 10, ma talvolta predice classi inesistenti, come evidenziato nella figura 11. Nonostante ciò, il modello addestrato con il dataset senza augmentation è stato preferito per la sua maggiore precisione nelle predizioni.

6 Conclusioni

Grazie alle metriche ottenute e alle predizioni fatte sulle immagini in analisi, è possibile dire che l'approccio utilizzato si è rivelato piuttosto efficacie per lo svolgimento del task in questione. Tuttavia i diversi tentativi di addestramento su modelli diversi hanno evidenziato quanto, per questi tipi di compito, sia complicato trovare modelli in grado di generalizzare bene oltre una certa soglia senza il rischio di overfitting. Oltre ad un fattore legato ai parametri di addestramento sono anche emerse sfide in merito alla scelta delle soglie e della dimensione delle finestre scorrevoli ottimale da utilizzare. Inoltre, dalle valutazioni emerge anche che le predizioni possono variare in maniera più o meno significativa in base alla natura dei dati di training e questo rende il task ancor più complesso in termini di scelte da effettuare. Un altro dettaglio evidenziato dai risultati è che ci sono determinate classi su cui il modello incontra maggiori difficoltà rispetto ad altre, una delle sfide future infatti sarà quella di rendere il modello più robusto per quei tipi di istanze senza che ciò intacchi le performance sulle altre. Infine si è osservato che l'algoritmo per effettuare le predizioni finali può non essere completamente affidabile, perciò un altro dei lavori futuri sarà quello di affinare maggiormente questa procedura per fare in modo che sempre più dettagli delle immagini siano classificati correttamente.