UNIVERSITÀ
DEGLI STUDI
FIRENZE

Scuola di Scienze Matematiche, Fisiche e Naturali

Corso di Laurea Magistrale in Informatica
Curriculum: *Data Science*

TITOLO IN ITALIANO

TITLE IN ENGLISH

FEDERICO SCHIPANI

Relatore: Prof. *Marco Bertini*

Anno Accademico 2018-2019 (spero)

INDICE

ELENCO DELLE TABELLE

ELENCO DELLE FIGURE

INTRODUZIONE

1

OBJECT DETECTION

L'Object Detection è un *task* legato al mondo della *computer vision* che consiste nel rilevare e classificare istanze di oggetti in immagini o video.

Negli ultimi anni, grazie soprattutto all'avvento delle Graphics Processing Unit (GPU), c'è stato un incremento notevole del potere computazionale. Questo ha portato a sviluppare tecniche sempre più raffinate allo scopo di raggiungere prestazioni sempre migliori.

Sempre grazie allo sviluppo di hardware sempre più potente l'interesse sta sempre più virando verso il mondo del *Deep Learning*. In questo capitolo cercheremo di classificare le varie metodologie con cui si porta a compimento la *Object Detection*.

1.1 TIPOLOGIE DI DETECTOR

La letteratura sui detector è molto disomogenea e variegata, prenderemo quindi come riferimento i lavori di *Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng and Rong Qu* [1] e *Zhengxia Zou, Zhenwei Shi, Yuhong Guo and Jieping Ye* [2].

1.2 RETINANET

2

ESPERIMENTI

2.1 DATASET

Inizialmente i dataset usati per gli esperimenti sono due. Il primo è KAIST Multispectral Pedestrian Dataset (KAIST MPD) [3], per cui è disponibile un'ampia documentazione, il secondo è un dataset gratuito realizzato dalla FLIR [4] per cui è disponibile una documentazione molto stringata.

2.1.1 *KAIST Multispectral Pedestrian Dataset*

Il Korea Advanced Institute of Science and Technology (KAIST) propone in [3] un dataset che fornisce coppie di immagini termiche e a colori. La particolarità che offre questo dataset è che le due immagini sono allineate. Inoltre sono state raccolte sufficienti immagini sia diurne che notturne.

SPECIFICHE HARDWARE KAIST ha sviluppato una piattaforma basata su una camera a colori, una termica ed un *Beam Splitter*, oltre che ad un supporto a tre assi chiamato *camera jig*. Un *Beam Splitter* è un dispositivo ottico di forma cubica formato in molti casi da due prismi che divide la luce in due parti. In questo caso viene utilizzato per l'allineamento delle due immagini in quanto permette il passaggio dello spettro termico mentre quello visibile viene riflesso. Il dispositivo usato per la realizzazione del dataset è stato costrutito a partire da un wafer di silicio zincato.

Le telecamere utilizzate sono una *PointGrey Flea3* per la parte a colori ed una *FLIR-A35* per la parte termica. La prima acquisisce immagini ad una risoluzione di 640x480 pixels con un Field of View (FOV) di 103.6°, mentre la seconda ha una risoluzione di 320x256 con un un FOV di 39°. Come si può notare il campo visivo della telecamera visibile è più ampio di quello della telecamera termica, motivo per cui viene sacrificata parte dell'immagine visibile al fine di allineare i due fotogrammi. Il *framerate* è

di 20 FPS.

CALIBRAZIONE L'idea per la realizzazione di questa architettura hardware è stata ripresa dal lavoro di Bienkowski *et al.* [5]. Sempre in questo lavoro però non si fa riferimento alla metodologia usata per la calibrazione. Parleremo in questo paragrafo dell'approccio utilizzato per la realizzazione di questo dataset. Innanzitutto è stata calcolata la traslazione fra le due telecamere, applicando la calibrazione stereo. Si può osservare che gli assi ottici delle telecamere al di là della divisione del fascio di luce sono paralleli a causa delle impostazioni tecniche. Di conseguenza, fra i due domini dell'immagine, è presente unicamente una traslazione ed è necessario solamente aggiustare la posizione tramite *camera jig* a tre assi finché la traslazione non diventa nulla. Dopo l'aggiustamento, i due domini sono rettificati fino ad avere la stessa distanza focale virtuale. Al termine di queste procedura, oltre alla focale, i domini condividono i punti principali e sono privi di baseline. Il dominio dell'immagine, virtualmente allineato, ha 640x512 pixel di risoluzione spaziale e un FOV di 39°, analogamente a quella umano. Visto che un pattern a scacchiera convenzionale non è osservabile con telecamera termica, viene invece utilizzata una tavola di calibrazione speciale, con un certo numero di buchi. Quando viene scaldata, si ottiene una differenza di temperatura fra la tavola e i buchi, che possono essere osservati nel termico.

CORREZIONE DEI COLORI Per via del passaggio all'interno dei prismi del *Beam Splitter* le immagini catturate, soprattutto nello spettro del visibile, mostrano distorsioni piuttosto evidenti dei colori. Per gestire questo problema è stato deciso di acquisire un fotogramma di riferimento completamente bianco che mostrava distorsioni di colore. Per motivi legati al sensore utilizzato all'interno della telecamera visibile la distorsione del colore può essere considerata come una funzione lineare. Quindi ogni pixel dell'immagine di riferimento può essere usato come coefficente di correzione per le altre immagini, dividendo il livello di intensità di queste immagini per questi coefficenti.

ACQUISIZIONE DEI DATI E ANNOTAZIONI Tutto il marchingegno composto dalle due telecamere, il *Beam Splitter* ed il *Camera jig* è stato montato sul tetto di un'automobile al fine di realizzare immagini egocentriche del traffico. In particolare, come già accennato in precedenza, sono state realizzate raccolte di dati sia di giorno che di notte.

Il numero totale delle coppie di immagini catturate sono 95328 le quali

sono state annotate manualmente con un totale di 103128 Bounding Box (BB). Per realizzare le annotazioni è stata usata una versione modificata del Piotr's Computer Vision Toolbox [6]. Le BB sono state annotate con quattro differenti label:

- `person`: individuo singolo ben individuabile
- `people`: individui non distinguibili
- `cyclist`: persone che stanno utilizzando una bicicletta
- `person?`: individuo non ben identificabile per via di fotogrammi molto densi

Inoltre, anche se per lo scopo della tesi non sono stati presi in considerazione, ogni BB ha una corrispondenza temporale che identifica il singolo individuo attraverso i vari frame.

TRAIN E TEST SET Per dividere tra *train set* e *test set* è stato usato un criterio ben definito:

- Il numero di pedoni nei due set è simile
- Il numero di frame notturni e diurni nei due set è simile
- I due set non si sovrappongono

PROPRIETA

- Attributo di scala: per ogni BB è associato un valore di scala. Questo valore dipende dalla distanza che ha il pedone dall'automobile, ed è giustificato dalla seguente affermazione. Supponendo che una vettura in area urbana viaggia ad una velocità compresa tra 30 e 50 km/h lo spazio di arresto varia tra gli 11 ed i 28 metri. Questo intervallo, scalato opportunamente rispetto alla risoluzione dell'immagine, e considerando anche che l'altezza media di un pedone è di 1.7 metri corrisponde ad un range che va da 45 a 115 pixel. All'interno di questo range vengono definite *medium*, al di sopra *far* ed al di sotto *near*.
- Occlusione: questo attributo associa ad ogni BB un valore che rappresenta l'occlusione del pedone. I valori possibili sono *no occlusion*, *partial occlusion*, *heavy occlusion*. I primi sono circa il 78.6%, i secondi circa il 12.6% e gli ultimi 8.8%.

- Posizione: l'impostazione dell'hardware rispecchia il più possibile quello di un essere umano, motivo per cui questo particolare setup concentra il rilevamento di pedoni nell'area centrale dell'immagine, in particolare nel lato destro. Questo è motivato dal fatto che nel paese dove sono stati acquisiti questi dati la guida è sulla destra. In figura **INSERIRE FIGURA** è possibile vedere questo fenomeno.
- Cambio d'aspetto: l'aspetto dei pedoni all'interno del dataset è molto variabile. In condizioni di pieno sole i pedoni sono ben visibili e con dei contorni ben definiti, mentre la differenza di temperatura tra l'ambiente circostante ed il pedone è meno marcata. Quindi nello spettro a colori sono presenti pedoni ben definiti, mentre nel termico no. Di notte invece, per via delle temperature ambientali più basse e per l'assenza di luce si verifica il contrario. In figura **INSERIRE FIGURA** è presente un esempio.

2.1.2 *FLIR Thermal Starter Dataset*

Come già accennato in precedenza la documentazione riguardante questo dataset è molto stringata, limitandosi dunque ad una sola pagina web molto riassuntiva. Cercheremo in questa sezione di parlare degli aspetti che caratterizzano questo dataset.

Il dataset in questione offre immagini termiche con annotazioni e l'equivalente a colori non annotato. A differenza di 2.1.1 le due immagini non sono allineate, quindi non è possibile portare le annotazioni delle immagini termiche sulle immagini a colori. Le immagini sono state acquisite tramite telecamere montate su una vettura e contiene un totale di 14453 immagini, di cui 10228 campionate da video di breve durata e 4224 provenienti da video di 144 secondi. Tutte le immagini sono state acquisite su strade ed autostrade a Santa Barbara, in California. L'arco temporale varia da Novembre a Maggio, nello stessa quantità di giorno e notte. Il meteo è generalmente buono.

Le immagini termiche sono state scattate con una *FLIR Tau2*, mentre quelle RGB con una *FLIR BlackFly*. Entrambi i device sono stati impostati in maniera tale da avere lo stesso FOV e per quanto riguarda il resto sono state lasciate entrambe alle impostazioni di default. Le videocamere sono state posizionate sullo stesso supporto a distanza di circa 1.9 pollici (circa 4.8 centimetri) l'una dall'altra. Il *framerate* è di 2 frame al secondo in scenari densi di annotazioni, mentre in scenari più tranquilli è stato deciso di scendere ad un frame al secondo.

Le annotazioni dove possibile ricalcano i codici adottati dal dataset COCO, ed hanno i seguenti codici:

- 1 People: esseri umani.
- 2 Bicycles: biciclette e moticicli. Questa è l'unica categoria non consistente con il formato adottato da COCO.
- 3 Cars: automobili e veicoli piccoli.
- 18 Dogs: cani
- 91 Other Vehicle: camion, rimorchi e imbarcazioni.

Le annotazioni sono state fatte manualmente da esseri umani ai quali è stato comunicato di fare BB il più piccole possibili e che omettessero piccole parti di oggetti, accessori personali e parti occluse. Inoltre è stato comunicato di non annotare oggetti di piccole dimensioni o molto occlusi e persone delle quali si vede solo braccia o gambe.

2.2 ADDESTRAMENTO INIZIALE DI RETINANET

In questa sezione presenteremo i risultati iniziali dell'addestramento di RetinaNet sui due dataset descritti in sezione 2.1. Per lo scopo è stata usata una versione di *RetinaNet* implementata tramite *Keras* reperibile in forma originale al seguente link. Durante lo sviluppo del lavoro di tesi le modifiche al codice originale sono state molteplici, tanto da aver richiesto un *fork* della *repository* originale reperibile al seguente link. Per tenere traccia dell'addestramento è stato usato *Weight & Biases*.

2.2.1 *Transfer Learning*

Inizialmente è stata usata la tecnica del *Transfer Learning*. Una rapida spiegazione del significato di questa espressione ce la fornisce il libro *Deep Learning* di *Goodfellow et al* [7]

> Transfer learning and domain adaptation refer to the situation where what has been learned in one setting is exploited to improve generalization in another setting.

Per un'introduzione più dettagliata su cos'è il *transfer learning* e sui vari tipi è stato preso spunto da *A survey on Transfer Learning* di *S. J. Pan* e *Q. Yang* [8].

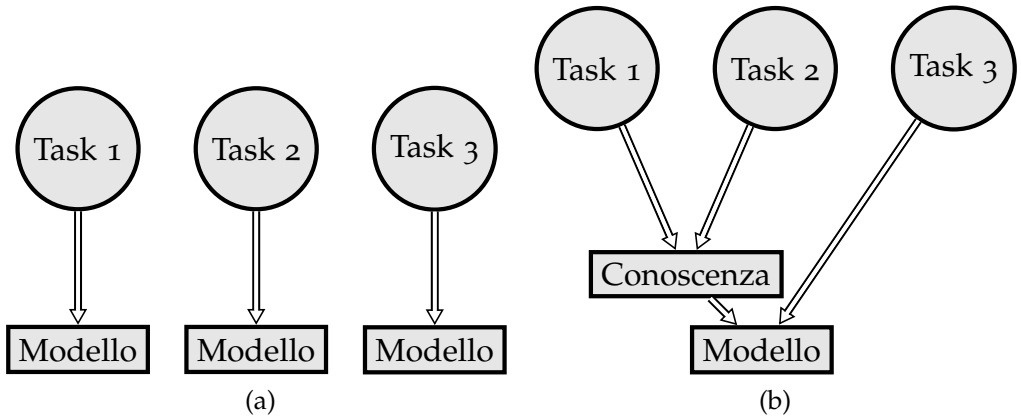

Figura 1: Differenze tra apprendimento tradizionale e transfer learning

La necessità di attuare tecniche di *transfer learning* deriva dal fatto che molti modelli di Machine Learning lavorano bene solo sotto determinate assunzioni, soprattutto quella che i dati di addestramento e di test derivino dallo stesso spazio delle *feature* e dalla stessa distribuzione. I problemi sorgono quando cambia la distribuzione, in quanto è necessario procedere ad una nuova fase di training.

Le casistiche in cui il *transfer learning* è applicabile sono molteplici, ad esempio l'analisi dei sentimenti, dove il compito è classificare le recensioni di un determinato prodotto in positive o negative. Per un compito del genere il primo passo da effettuare è la raccolta e l'annotazione di recensioni. Successivamente è necessaria una fase di addestramento di un modello usando come dati le recensioni precedentemente raccolte ed annotate. Lo scopo è poter usare lo stesso modello per vari prodotti, in questo caso però si va incontro al problema che le distribuzioni dei dati su diversi prodotti possono differire anche di molto. La soluzione sarebbe quindi di annotare altre recensioni, ma richiederebbe uno sforzo notevole. L'idea è quindi di adattare un modello di classificazione, addestrato su alcuni prodotti, per aiutare la fase di addestramento su articoli differenti. Le differenze tra processi di apprendimento tradizionali e *transfer learning* sono mostrate in Figura 1.

NOTAZIONE PRELIMINARE Per introdurre un po' più nello specifico le varie tipologie di *transfer learning* è necessario definire alcuni concetti. Il primo di questi è il *Dominio* $\mathcal{D}$, definito come una tupla $\mathcal{D} = \{\mathcal{X}, P(X)\}$, dove $\mathcal{X}$ è lo spazio delle *feature* e $P(X)$ con $X = \{x_1, x_2, \ldots, x_n\} \in \mathcal{X}$ è una distribuzione di probabilità marginale. In generale due domini posso-

no essere considerati differenti se hanno differenti spazi delle *feature* o distribuzioni differenti.

Dato uno specifico dominio $\mathcal{D} = \{\mathcal{X}, P(X)\}$ è possibile definire il *task*. Un *task* $\mathcal{T}$ è una tupla $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$ con $\mathcal{Y}$ spazio dei *label* e $f(\cdot)$ funzione di predizione. La particolarità del *task* è il fatto che non è osservabile ma può essere appreso dai dati di addestramento, che consistono di una coppia $\{x_i, y_i\}$ con $x_i \in X$ e $y_i \in \mathcal{Y}$. Una volta completata la fase di addestramento dovrebbe essere possibile usare la funzione $f(\cdot)$ per predirre il label $f(x)$ corrispondente ad una nuova istanza x.

Chiameremo il dominio sorgente $\mathcal{D}_S$ ed il dominio target $\mathcal{D}_t$. In particolare avremo a disposizione anche i dati D_S del dominio sorgente, definiti come $D_S = \{(x_{S_1}, y_{S_1}), \ldots, (x_{S_n}, y_{S_n})\}$ tali che $x_{S_i} \in \mathcal{X}_S$ è l'istanza del dato e $y_{S_i} \in \mathcal{Y}_S$ è il corrispondente label. In maniera similare definiamo anche i dati del dominio target $D_T = \{(x_{T_1}, y_{T_1}), \ldots, (x_{T_n}, y_{T_n})\}$ tali che $x_{T_i} \in \mathcal{X}_T$ è l'istanza del dato e $y_{T_i} \in \mathcal{Y}_T$ è il corrispondente label.

Dire che due domini $\mathcal{D}_S$ e $\mathcal{D}_T$ implica che o $\mathcal{X}_S \neq \mathcal{X}_T$ oppure $P_X(X) \neq P_T(X)$. In maniera del tutto analoga è possibile definire la differenza tra due *task* $\mathcal{T}_S$ e $\mathcal{T}_T$. Nel caso in cui i due domini ed i due task sono uguali ci si riconduce ad un tradizionale problema di apprendimento. Quando invece c'è una relazione, implicita o esplicita, tra gli spazi delle *feature* dei due domini si dice che il dominio sorgente e target sono in relazione tra di loro.

TIPOLOGIE DI TRANSFER LEARNING Prima di introdurre le varie tipologie di *transfer learning* è necessario definire formalmente questo concetto e fare alcune premesse.

Definizione 2.2.1. (Transfer Learning) Dato un dominio sorgente $\mathcal{D}_S$, un task di apprendimento $\mathcal{T}_S$, un dominio target $\mathcal{D}_T$ ed un task di apprendimento $\mathcal{T}_T$, il transfer learning tenta di migliorare l'apprendimento di $f(\cdot)_T \in \mathcal{D}_T$ usando la conoscenza in $\mathcal{D}_S$ e $\mathcal{T}_S$, con $\mathcal{D}_S \neq \mathcal{D}_T$ o $\mathcal{T}_S \neq \mathcal{T}_T$.

Quando si parla di *transfer learning* bisogna mettere in conto tre problemi: *cosa*, *come* e *quando* trasferire.

- Cosa trasferire: quale parte della conoscenza bisogna trasferire tra domini o task sorgenti e target. In particolare possiamo dire che alcune conoscenze sono comuni tra i diversi domini, mentre altre sono specifiche.

- Come trasferire: a questo problema pone una soluzione l'algoritmo di trasferimento della conoscenza.

- Quando trasferire: ci chiediamo in quali situazioni è realmente necessario applicare tecniche di *transfer learning* ed in quali non è assolutamente necessario. Ad esempio in casi in cui i due domini sorgente e target non sono in relazione tra di loro il *transfer learning* potrebbe non portare ad alcun risultato positivo. Si tende quindi sempre a parlare di *transfer learning* dando per scontato che i due domini siano in qualche modo relazionati tra di loro, in quanto altrimenti non avrebbe senso andare oltre l'apprendimento tradizionale.

Possiamo ora descrivere le tre categorie di *transfer learning*:

- *Transfer Learning Induttivo*: il task target è differente dal task sorgente e non importa se il dominio sorgente ed il dominio target sono uguali o meno. Per indurre un modello predittivo oggettivo, da usare nel dominio target, sono necessari alcuni dati etichettati nel dominio sorgente. A seconda della tipologia ed alla quantità di annotazioni possiamo dividere questo tipo di *transfer learning* in ulteriori due sottocategorie.
  - Molti dati annotati nel dominio sorgente: ci si riconduce al caso del *multitask learning*, tuttavia mentre lo scopo di quest'ultimo è operare bene in entrambi i domini, lo scopo del *transfer learning* induttivo è operare bene solamente sul dominio obbiettivo.
  - Nessun dato annotato nel dominio sorgente: le similarità in questo caso ci portano a pensare al *self-taught learning*, dove le annotazioni tra il dominio sorgente ed obbiettivo sono totalmente differenti, e quindi non direttamente utilizzabili.
- *Transfer Learning Trasduttivo (si traduce così "transductive"?)*: in questo caso il task obbiettivo ed il task sorgente sono i medesimi, mentre i domini sono differenti. Abbiamo quindi molti dati annotati nel dominio sorgente e nessuna annotazione nel dominio obbiettivo. Possiamo a sua volta dividere questa categoria in ulteriori due sottocategorie.
  - Gli spazi delle feature tra sorgente e obbiettivo sono diffrenti, più formalmente abbiamo $\mathcal{X}_S \neq \mathcal{X}_T$
  - Gli spazi delle feature tra sorgente e obbiettivo sono gli stessi, ma cambia la distribuzione di probabilità marginale, quindi $P(X_S) \neq P(X_T)$. Quest'ultimo caso è chiamato anche *Domain Adaptation*.

Tipologia	Similarità	Annotazioni Sorgente	Annotazioni Target	Campo di applica
Induttivo	Multitask Learning	SI	SI	Regressione e Classif
	Self-taught Learning	NO	SI	Regressione e Classif
Trasduttivo	Domain adaptation	SI	NO	Regressione e Classif
n supervisionato	46501	NO	NO	Clustering, dimensionali

Tabella 1: Schema riassuntivo delle categorie di Transfer Learning

- *Transfer Learning non supervisionato*: il task obbiettivo è differente dal task sorgente, ma hanno una qualche tipo di relazione tra di loro. Tuttavia l'attenzione si focalizza sul risolvere compiti di apprendimento non supervisionati nel dominio obbiettivo, come possono essere il *clustering*, *dimensionality reduction* o *density estimation*. Non abbiamo quindi annotazioni né nel dominio sorgente, né nel dominio obbiettivo.

In Tabella 1 sono riassunte tutte le caratteristiche principali delle varie categorie di *transfer learning*. **SISTEMARE TABELLA 1**

2.2.2 *Addestramento sulle immagini RGB*

Per il primo esperimento è stato deciso di effettuare un training di *RetinaNet* partendo dai pesi della rete precedentemente addestrata sul dataset di COCO. Il dataset utilizzato è KAIST MPD, descritto precedentemente in 2.1.1. Il motivo è legato al fatto che è l'unico dataset a nostra disposizione a disporre di annotazioni sulle immagini RGB.

Questa prima fase di addestramento è durata circa 40 ore, e come si può vedere dal grafico in Figura 2 è proceduta senza particolari problemi fino ad arrivare a convergenza intorno all'epoca 45. Le classi usate per l'addestramento sono solamente *person* e *cyclist*. È stato deciso di non prendere in considerazioni le rimanenti classi in quanto sono persone non ben distinguibili. Tutti gli esperimenti sono stati eseguiti su una macchina remota dotata di una GPU Nvidia Titan X.

Dopo la fase di addestramento sono stati eseguite le valutazioni sulla parte di dataset adibita ai test. Inizialmente le classi utilizzate per i test sono le stesse usate per l'addestramento. I risultati complessivi vengono riassunti in Tabella 2. La Mean Average Precision (mAP) mostrata nell'ultima riga della tabella è stata calcolata come media pesata secondo il numero di esempi all'interno del *test set*.

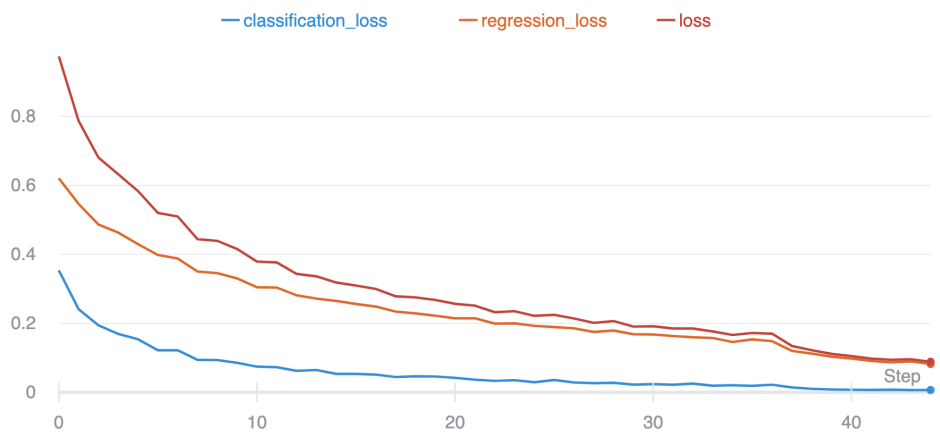

Figura 2: Transfer learning da COCO a KAIST

	# Istanze	mAP
Person	45195	0.4184
Cyclist	1396	0.1154
Complessivo	46501	0.4093

Tabella 2: Test complessivo delle performance dopo l'addestramento

La valutazione è stata anche effettuata in maniera separata sia sulla parte di *test set* diurna che notturna. I risultati sono mostrati in Tabella 3a e Tabella 3b. Come lecito aspettarsi, avendo visibilità limitata in notturna si ottengono risultati mediamente peggiori. In Figura 3 è possibile vedere un esempio di predizioni fatte sul *test set*.

Il successivo step della valutazione è stato effettuato su più classi, per questa comparazione delle performance è stata presa in considerazione anche la classe *people*, ma rinominandola in *person* in maniera tale da farla digerire **rivedere questo periodo** alla rete già addestrata. **INSERIRE**

	# Istanze	mAP
Person	33688	0.4493
Cyclist	818	0.2015
Complessivo	34506	0.4434

(a) Giorno

	# Istanze	mAP
Person	11507	0.3281
Cyclist	578	0.0029
Complessivo	12085	0.3125

(b) Notte

Tabella 3: Risultati della valutazione separata

Figura 3: Esempio di predizioni, in verde la *ground truth* in rosso le predizioni.

VALUTAZIONE EFFETTUATA CON TUTTE LE CLASSI

ACRONIMI

mAP Mean Average Precision

KAIST MPD KAIST Multispectral Pedestrian Dataset

GPU Graphics Processing Unit

KAIST Korea Advanced Institute of Science and Technology

FOV Field of View

BB Bounding Box

BIBLIOGRAFIA

[1] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *CoRR*, abs/1907.09408, 2019. (Cited on page 9.)

[2] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *CoRR*, abs/1905.05055, 2019. (Cited on page 9.)

[3] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1037–1045. IEEE Computer Society, 2015. (Cited on page 11.)

[4] Free flir thermal dataset for algorithm training. `https://www.flir.com/oem/adas/adas-dataset-form/`. Accessed: 2019-11-20. (Cited on page 11.)

[5] L. Bienkowski, C. Homma, K. Eisler, and Christian Boller. Hybrid camera and real-view thermography for nondestructive evaluation. 01 2012. (Cited on page 12.)

[6] Piotr Dollár. Piotr's Computer Vision Matlab Toolbox (PMT). `https://github.com/pdollar/toolbox`. (Cited on page 13.)

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`. (Cited on page 15.)

[8] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct 2010. (Cited on page 15.)