

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Corso di Laurea in Informatica per il management

**UNA PIATTAFORMA A SUPPORTO
DELLA RICERCA TRAMITE
RICONOSCIMENTO IMMAGINI**

Relatore:
Dott.
LUCA BEDOGNI

Presentata da:
CHIARA ZUFFI

Sessione III
Anno Accademico 2017-2018

Alla mia famiglia

Indice

Elenco delle Figure	1
1 Introduzione	3
2 Stato dell'arte	5
2.1 Background	5
2.2 DUEproject	7
2.2.1 Ricerca e catalogazione	7
2.3 iNaturalist	10
2.3.1 Funzionamento	10
2.4 Riconoscimento immagini	13
2.4.1 Classificazione delle immagini	13
3 Progettazione	15
3.1 Scopo e funzionalità	15
3.2 Architettura	16
3.2.1 Pagina Principale	16
3.2.2 Pagina delle predizioni	18
3.2.3 Pagina della mappa	20
4 Implementazione	21
4.1 Tecnologie utilizzate	21
4.1.1 Javascript e Node.js	21
4.1.2 HTML, CSS e Bootstrap	22

4.1.3	Instagram API	23
4.1.4	TensorFlow.js	23
4.1.5	Keras MobileNet	24
4.2	Scelte implementative	25
4.2.1	Caricamento del modello	25
4.2.2	Caricamento delle immagini	26
4.2.3	Elaborazione dell'immagine	27
4.2.4	Predizione	28
4.2.5	Web storage	29
4.2.6	Mappa	29
5	Conclusioni e lavori futuri	31
	Bibliografia	33

Elenco delle figure

2.1	Struttura di DUEproject [5]	8
2.2	Scheda con le fotografie [5]	9
2.3	Questionario valutativo [5]	9
2.4	Osservazioni iNaturalist [6]	11
2.5	Grafico osservazioni/specie di iNaturalist [8]	12
2.6	Assegnamento etichetta [10]	14
3.1	Pagina principale	16
3.2	Modal di Instagram	17
3.3	Predizione immagine scelta	18
3.4	Pagina delle predizioni	19
3.5	Mappa delle predizioni	20
4.1	Neuroni di output [20]	25

Capitolo 1

Introduzione

Al giorno d'oggi si sente sempre più spesso parlare di machine learning, o apprendimento automatico. Esso si concentra sulla capacità dei software di ricevere una serie di dati e di apprendere in modo autonomo man mano che ricevono più informazioni su quello che stanno elaborando. L'apprendimento automatico è, quindi, un modo per “educare” un algoritmo in modo che possa apprendere. E' possibile, ad esempio, educare un sistema a riconoscere e classificare immagini tramite un modello di apprendimento.

In particolare, nell'ambito della ricerca biologica, l'apprendimento automatico può essere di aiuto a ricercatori e biologi per velocizzare il lavoro di riconoscimento delle specie e, di conseguenza, essere un valido supporto per il monitoraggio delle biodiversità di una determinata area.

Esistono tantissimi progetti di ricerca in cui il riconoscimento delle specie, di piante e animali, viene fatto manualmente esaminando ogni singola foto precedentemente scattata e cercando di riconoscere ogni specie presente. Questo occupa un sacco di tempo prezioso ai ricercatori. L'idea di inserire l'apprendimento automatico in questo ambito nasce proprio da questo. Utilizzando un modello di apprendimento precedentemente addestrato è possibile non solo ridurre i tempi di riconoscimento, ma anche ottenere predizioni più precise in base alla specificità del modello utilizzato.

Questo lavoro propone una piattaforma software in grado di riconoscere e classificare le immagini caricate dall'utente. Inoltre, viene data la possibilità di salvare le predizioni in locale tramite Web storage in modo da rendere persistenti le informazioni ricavate e poterle visualizzare in un secondo momento.

Tutto questo viene reso possibile grazie a Keras, un'API (Application Programming Interface) dotata di modelli di apprendimento pre-addestrati.

Nei capitoli successivi verrà illustrato il tema dell'identificazione automatizzata in ambito biologico, attraverso la presentazione di due importanti progetti di monitoraggio delle biodiversità. Verrà, inoltre, descritta nel dettaglio la piattaforma realizzata. Verranno, quindi, mostrati i processi di progettazione e di implementazione della stessa e gli sviluppi che potranno essere apportati in futuro.

Capitolo 2

Stato dell'arte

In questo capitolo verrà illustrata l'identificazione automatizzata delle specie attraverso due progetti di ricerca naturalistica.

Il paragrafo 2.2 illustrerà il progetto di monitoraggio delle biodiversità DUE-project, il paragrafo 2.3 il progetto di identificazione iNaturalist e il paragrafo 2.4 tratterà del riconoscimento delle immagini.

2.1 Background

Al giorno d'oggi ci sono milioni di individui, scienziati e non, che si impegnano in progetti di raccolta, analisi e trascrizione di dati scientifici. Essi contribuiscono alla ricerca scientifica e formano quella che può essere definita una scienza dei cittadini: Citizen Science.

Esistono tipi differenti di Citizen Science in base al coinvolgimento dei cittadini. Nei crowd-sourcing projects, ad esempio, i volontari mettono a disposizione dei ricercatori i propri dispositivi mobili per ricercare, estrarre dati e informazioni e analizzarli. In altri progetti invece i volontari raccolgono attivamente informazioni in svariati campi come la biologia, ecologia, astronomia e così via. Infine ci sono progetti dove i volontari contribuiscono ad analizzare i dati o collaborano con i ricercatori nella ricerca.

Uno dei vantaggi della Citizen Science è, quindi, quello di sviluppare monitoraggi su larga scala raccogliendo una notevole massa di dati con un costo notevolmente inferiore rispetto ai monitoraggi operati dai professionisti. Un altro vantaggio è il procuramento di informazioni difficilmente ottenibili attraverso i metodi tradizionali. Inoltre, il coinvolgimento dei volontari li rende consapevoli dell'importanza e delle difficoltà della ricerca scientifica, rendendoli più rispettosi e interessati all'ambiente e alle tematiche scientifiche. E', però, importante specificare precisamente cosa chiedere ai volontari e valutare se l'attività è fattibile per loro poiché questo influisce considerevolmente sull'affidabilità dei dati. [1]

La Citizen Science permette di aprire le porte della scienza anche ai cittadini. Tuttavia condividere una così grande quantità di dati con il pubblico, a volte, può portare notevoli rischi. Ad esempio per quanto riguarda gli animali in via di estinzione. Questa accessibilità alle informazioni, ad esempio l'esatta posizione, espone gli animali al rischio di essere facilmente individuabili da bracconieri o appassionati, come birdwatchers, fotografi o collezionisti, che a volte possono essere fonte di problemi per le specie che osservano.

Come si evince da uno studio pubblicato recentemente su *Nature Ecology and Evolution* [2], da una parte la condivisione di informazioni su una specie in pericolo ha enormi potenzialità, dall'altra il cattivo utilizzo di queste informazioni procura molti rischi. Per questo motivo gli scienziati hanno sviluppato un protocollo decisionale che possa aiutare nella valutazione di rischi e benefici della condivisione di dati riguardanti le specie in pericolo.

L'obiettivo è di capire, per le diverse specie, se la condivisione può essere davvero utile. In alcuni casi ad esempio sarebbe utile tenere nascoste tutte le informazioni o rivelare solo dati parziali. [3]

2.2 DUEproject

DUEproject (Divers United for the Environment) è un progetto di *monitoraggio della biodiversità del Mar Mediterraneo* creato da un team di ricercatori del Dipartimento di Scienze biologiche, geologiche e ambientali dell'Università di Bologna.

La biodiversità in questo contesto rappresenta la diversità a livello di specie, la quale è fortemente influenzata dai cambiamenti ambientali.

Lo scopo di questo progetto è quello di proteggere la vita del Mediterraneo attraverso monitoraggi e osservazioni. Grazie al contributo di migliaia subacquei ricreativi è possibile valutare la salute ambientale registrando la presenza e l'abbondanza delle principali specie mediterranee.

2.2.1 Ricerca e catalogazione

Una caratteristica importante di DUEproject è il fatto di essere *Citizen Science*, ossia di coinvolgere i cittadini per la ricerca scientifica. Questo permette di aumentare la conoscenza e la consapevolezza ambientale e ottenere moltissimi dati in poco tempo.

Il coinvolgimento dei volontari riduce i tempi di raccolta e i costi di ricerca, ma allo stesso tempo richiede maggior valutazione dell'affidabilità dei dati raccolti in quanto gli operatori sono non professionisti.

DUEproject è supportato da PADI, la principale organizzazione di addestramento subacqueo del mondo e da SCUBAPRO, azienda leader nella produzione di attrezzature per la subacquea. Grazie a questi enti DUEproject si può aprire a una ricerca in una più ampia area geografica.

La figura 2.1 mostra la struttura e l'organizzazione di DUEproject.



Figura 2.1: Struttura di DUEproject [5]

La raccolta dei dati avviene distribuendo una particolare *scheda/questionario* ai volontari. Il questionario ha lo scopo di riconoscere le specie chiave del Mediterraneo e raccogliere dati sulla loro presenza e abbondanza.

Gli organismi rappresentati nel questionario sono 3 piante e 41 animali e comprendono un ampio spettro di specie dalle caratteristiche comuni e facili da riconoscere. Questo permette alle attività di DUEproject di essere realizzabili per chiunque voglia prendervi parte.

La compilazione del questionario avviene al termine dell'immersione e permette ai subacquei di confrontarsi e apprendere cose nuove.

La scheda/questionario comprende 2 parti principali:

- fotografie degli organismi da segnalare se avvistati, come mostra la figura 2.2;
- un questionario scientifico per la segnalazione degli avvistamenti, mostrato nella figura 2.3.

Tutti i questionari, compilati manualmente, sono poi inviati all'Università di Bologna. [4] [5]



Figura 2.2: Scheda con le fotografie [5]

In which habitat did you spend most of your dive? (indicate only one) sandy rocky other

Please mark with a cross the organisms that you have seen, giving an estimation of their abundance. Your diving instructor can help you!

	RARE	FREQUENT	ABUNDANT
1- VEGETALS			
1/A - mermaid's wine glass (<i>Acetabularia acetabulum</i>)	<input type="checkbox"/> up to 100 specimens	<input type="checkbox"/> up to 1000	<input type="checkbox"/> more than 1000
1/B - sea rose (<i>Peyssonnelia squamaria</i>)	<input type="checkbox"/> up to 10 specimens	<input type="checkbox"/> up to 100	<input type="checkbox"/> more than 100
1/C - Neptune grass (<i>Posidonia oceanica</i>)	<input type="checkbox"/> one hortus, up to 100 m ²	<input type="checkbox"/> one meadow, up to 1000	<input type="checkbox"/> one prairie, more than 1000
other vegetals	<input type="checkbox"/> up to 50 specimens	<input type="checkbox"/> up to 500	<input type="checkbox"/> more than 500
2 - SPONGES (PORIFERA)			
2/A - chicken liver sponge (<i>Chondrilla nucula</i>)	<input type="checkbox"/> up to 10 specimens	<input type="checkbox"/> up to 100	<input type="checkbox"/> more than 100
2/B - stony sponge (<i>Petrosia ficiformis</i>)	<input type="checkbox"/> up to 2 specimens	<input type="checkbox"/> up to 5	<input type="checkbox"/> more than 5
other sponges	<input type="checkbox"/> up to 6 specimens	<input type="checkbox"/> up to 50	<input type="checkbox"/> more than 50
3 - COELENTERATA, ANTHOZOA, OCTOCORALLIA			
3/A - precious red coral (<i>Corallium rubrum</i>)	<input type="checkbox"/> up to 10 colonies	<input type="checkbox"/> up to 100	<input type="checkbox"/> more than 100
3/B - violetescent sea-whip (<i>Paramuricea clavata</i>)	<input type="checkbox"/> up to 3 colonies	<input type="checkbox"/> up to 10	<input type="checkbox"/> more than 10
3/C - red dead men's fingers (<i>Alcyonium palmatum</i>)	<input type="checkbox"/> up to 1 colony	<input type="checkbox"/> up to 3	<input type="checkbox"/> more than 3
other octocorals	<input type="checkbox"/> up to 5 colonies	<input type="checkbox"/> up to 40	<input type="checkbox"/> more than 40
4 - COELENTERATA, ANTHOZOA, HEXACORALLIA			
4/A - snakelocks anemone (<i>Anemonia sulcata</i>)	<input type="checkbox"/> up to 15 specimens	<input type="checkbox"/> up to 40	<input type="checkbox"/> more than 40
4/B - yellow cluster anemone (<i>Parazoanthus axinellae</i>)	<input type="checkbox"/> up to 100 specimens	<input type="checkbox"/> up to 1000	<input type="checkbox"/> more than 1000
4/C - cylinder anemone (<i>Cerianthus membranaceus</i>)	<input type="checkbox"/> up to 2 specimens	<input type="checkbox"/> up to 5	<input type="checkbox"/> more than 5
other hexacorals	<input type="checkbox"/> up to 10 specimens	<input type="checkbox"/> up to 30	<input type="checkbox"/> more than 30

Figura 2.3: Questionario valutativo [5]

2.3 iNaturalist

iNaturalist è un social network online con lo scopo di *monitorare la diffusione delle specie animali viventi e i mutamenti degli ecosistemi* di tutto il mondo.

Si tratta di un progetto Citizen Science che consente a chiunque di scoprire nuove specie, registrare le proprie osservazioni e condividerle con la comunità.

L'utente può accedervi sia tramite sito web *iNaturalist.org* sia attraverso l'omonima applicazione. Questo permette di raccogliere una grande mole di dati utile sia a fini di ricerca sia a scopi ricreativi per chiunque.

2.3.1 Funzionamento

Sfruttando l'intelligenza artificiale, iNaturalist è in grado di identificare piante e animali da una semplice foto.

L'utente non deve far altro che registrare la propria fotografia e il luogo in cui la foto è stata scattata e automaticamente la piattaforma farà il resto riconoscendo la specie presente nell'immagine.

Come mostra la figura 2.4, iNaturalist possiede una mappa del mondo in cui vengono mappate le fotografie caricate dagli utenti in base alla localizzazione inserita. Questo permette di effettuare ricerche in base a specie e/o localizzazione in modo da ottenere tutte le informazioni desiderate.

Ogni utente può, inoltre, interagire con gli altri per confermare o migliorare l'identificazione di una determinata foto laddove il riconoscimento risulta impreciso.

L'identificazione delle varie specie avviene attraverso strumenti di apprendimento automatico come la visione artificiale.

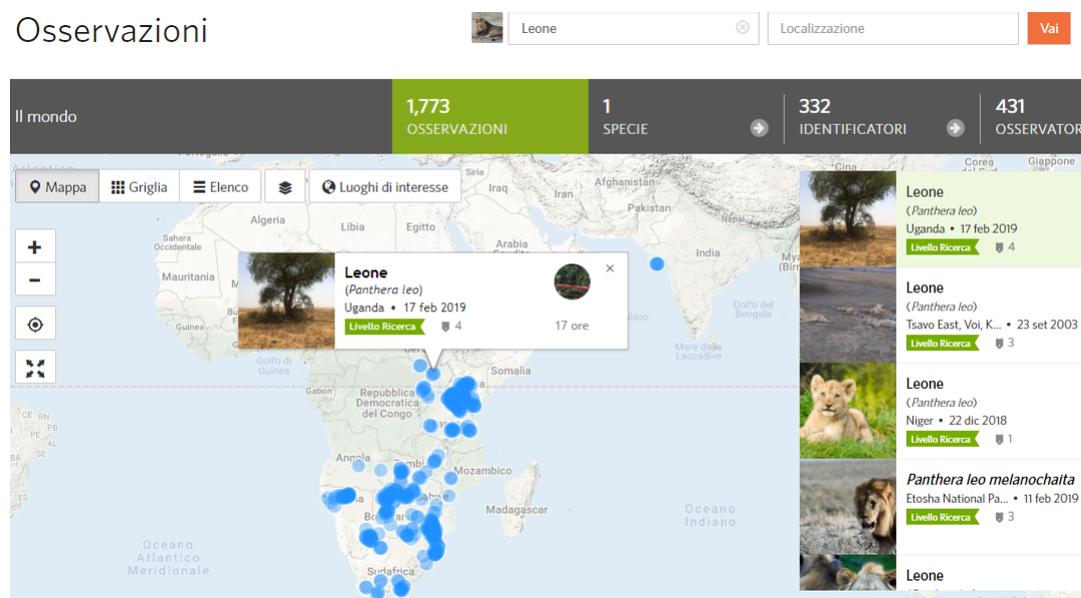


Figura 2.4: Osservazioni iNaturalist [6]

Utilizzando la libreria software open source TensorFlow è possibile formare modelli di classificazione delle immagini sottoponendoli all'analisi di grandi serie di immagini etichettate.

Nel caso di iNaturalist, le immagini sono le foto delle osservazioni e le etichette sono le loro identificazioni a livello di specie.

Una volta addestrato, il modello può essere utilizzato per identificare le immagini ricevendo un'immagine senza etichetta e assegnando a quest'ultima l'etichetta corrispondente.

Allenando il sistema al riconoscimento è dunque possibile identificare le caratteristiche di ogni specie.

iNaturalist possiede 5.000.000 milioni di osservazioni verificabili, ossia idonee a diventare "osservazioni di ricerca". Le osservazioni di ricerca sono quelle che sono state controllate dalla comunità e che sono quindi piuttosto precise. Tutte queste osservazioni rappresentano circa 100.000 specie distinte.

La figura 2.5 mostra un grafico del 2017 in cui viene mostrato il rapporto tra numero delle osservazioni e numero di specie. Come si evince dal grafico, il

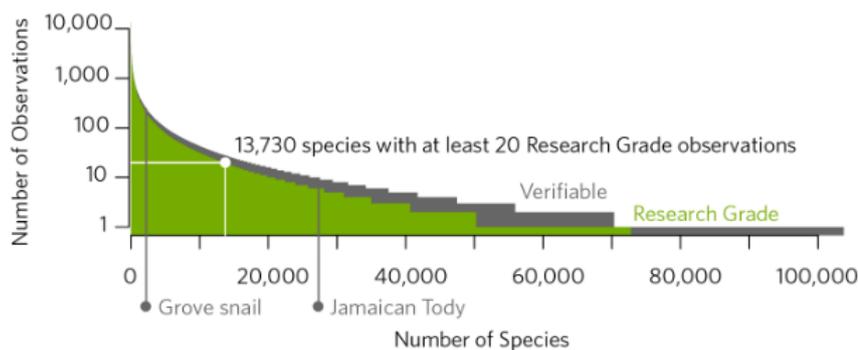


Figura 2.5: Grafico osservazioni/specie di iNaturalist [8]

numero di osservazioni per specie non è uniforme: ci sono specie che hanno molte osservazioni e altre che ne hanno molte meno. Questo incide sul grado di affidabilità dell'identificazione.

Ogni osservazione che viene pubblicata o identificata è, quindi, utile per migliorare il modello. Utilizzando gli strumenti di visione artificiale iNaturalist riesce a fornire identificazioni di qualità superiore sempre più velocemente man mano che continua a crescere.

Nonostante questo gli sviluppatori di iNaturalist non nascondono che il sistema non sia infallibile, l'hanno infatti definito "accurato ma non preciso". [7] [8]

2.4 Riconoscimento immagini

Il riconoscimento dell'immagine, nel contesto della visione artificiale, è la capacità del software di identificare oggetti, luoghi, persone, scrittura e azioni nelle immagini.

La capacità delle macchine di riconoscere immagini e forme aumenta a velocità esponenziale e viene utilizzata per eseguire un gran numero di attività visive basate su macchine, da etichettare il contenuto delle immagini, a eseguire ricerca di contenuti di immagini, fino ad arrivare a guidare robot autonomi. [9]

I software per il riconoscimento delle immagini richiedono un profondo apprendimento automatico. Devono quindi essere sottoposti all'analisi di migliaia di immagini con decine di foto per ciascun tipo di oggetto presente, costruendo i cosiddetti modelli. Questi modelli verranno poi utilizzati per riconoscere le immagini.

2.4.1 Classificazione delle immagini

L'occhio umano percepisce un'immagine come un insieme di segnali che vengono elaborati dal cervello. Il riconoscimento dell'immagine tenta di imitare questo processo.

Il computer percepisce una foto come una sequenza di pixel con valori numerici per ogni colore. Per analizzare le immagini la codifica dei pixel viene trasformata in costrutti che rappresentano caratteristiche e oggetti fisici. L'organizzazione dei dati comporta la classificazione e l'estrazione delle caratteristiche.

Una volta addestrato l'algoritmo di classificazione mostrandogli migliaia di immagini di un determinato oggetto, esso è pronto per riconoscere immagini test e assegnargli un'etichetta.

L'immagine test viene inserita nel modello per essere riconosciuta. Prendendo le caratteristiche come input, il classificatore darà come risultato un'etichetta. [10]

La figura 2.6 mostra il processo di assegnamento di un'etichetta.

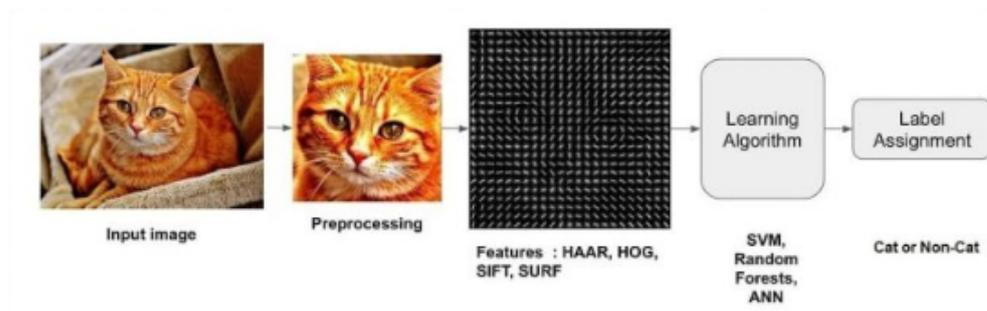


Figura 2.6: Assegnamento etichetta [10]

Esistono modelli di apprendimento automatico già addestrati a riconoscere il contenuto di determinate immagini. Una di essi è, ad esempio, MobileNet. Questo modello è in grado di fare predizioni su immagini che appartengono alle categorie ImageNet, un database con migliaia di categorie etichettate.

Capitolo 3

Progettazione

In questo capitolo verrà trattata la progettazione della piattaforma realizzata. Nel paragrafo 3.1 verranno spiegati il suo scopo e le sue funzionalità, e nel paragrafo 3.2 la sua architettura.

3.1 Scopo e funzionalità

Il progetto consiste nella progettazione e l'implementazione di una piattaforma software per il caricamento, il riconoscimento e la classificazione di immagini provenienti dal proprio computer o da profili pubblici di Instagram. L'obiettivo della piattaforma è quello di fornire all'utente uno strumento che permetta di analizzare le immagini caricate, riconoscendo il soggetto o i soggetti presenti nell'immagine, e successivamente classificarle secondo l'accuratezza della predizione.

La piattaforma consente all'utente di scegliere la modalità con cui caricare l'immagine: dal proprio PC o da Instagram, tramite la ricerca di hashtag o profili pubblici.

Il riconoscimento delle immagini caricate avviene per mezzo di un modello preimpostato Keras MobileNet il quale è in grado di effettuare predizioni su immagini appartenenti alle categorie ImageNet.

La classificazione delle immagini si basa sull'accuratezza della predizione e

viene definita in tre categorie: VERIFIED, PENDING e UNCERTAIN. Ogni immagine sulla quale è stata fatta una predizione appartiene a una di queste categorie.

All'utente viene data la possibilità di salvare l'immagine con la relativa predizione in locale tramite Web storage.

Sulla piattaforma è inoltre presente una mappa sulla quale sono riportati i marker relativi alle immagini caricate da Instagram che possiedono la localizzazione.

3.2 Architettura

Si può suddividere la piattaforma in due parti: un *front-end*, che si occupa dell'interazione con l'utente e della visualizzazione delle immagini con le relative informazioni, e un *back-end*, che gestisce il caricamento del modello e delle immagini, le predizioni e il salvataggio dei dati sul browser dell'utente.

3.2.1 Pagina Principale

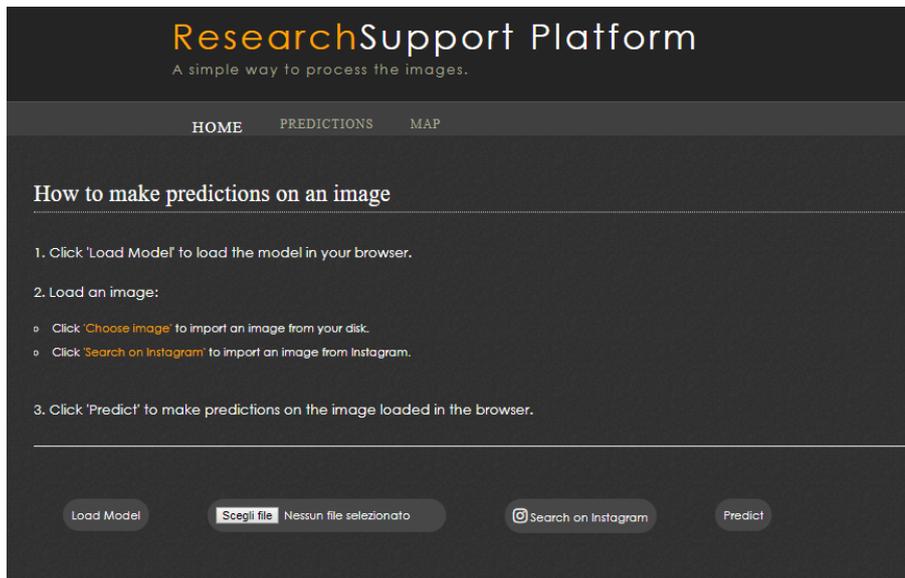


Figura 3.1: Pagina principale

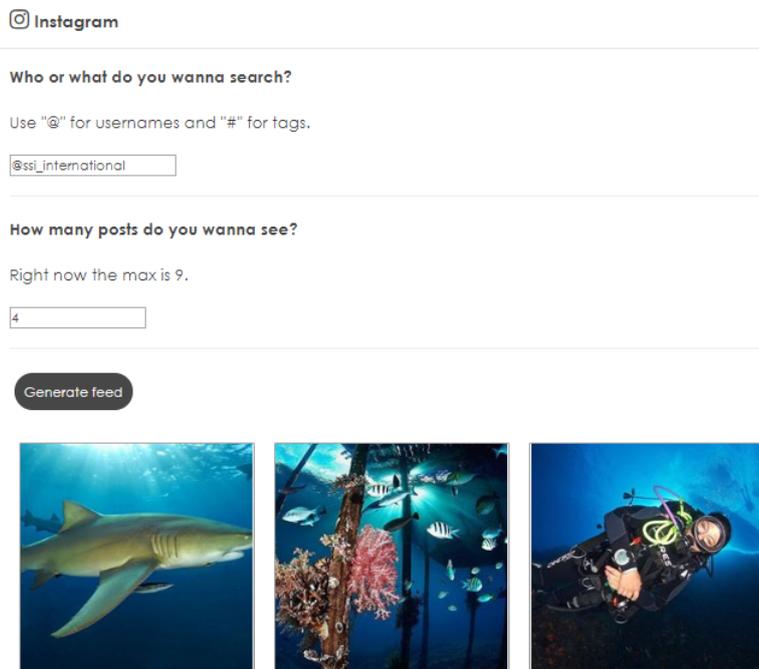
Nella figura 3.1 è mostrata la pagina principale della piattaforma in cui è presente una piccola guida per il suo funzionamento e i relativi pulsanti.

L'utente che intende fare predizioni sulle immagini deve innanzitutto caricare il modello tramite l'apposito pulsante *Load Model* e, successivamente, può scegliere il modo in cui caricare l'immagine.

Il pulsante *Scegli file* consente all'utente di scegliere un file proveniente dall'archivio del dispositivo.

Tramite il pulsante *Choose on Instagram* all'utente viene invece mostrata una finestra pop up in cui è possibile effettuare la ricerca di un hashtag, utilizzando il carattere “#” o di un profilo pubblico di un utente, utilizzando il carattere “@”. E' inoltre necessario inserire il numero di immagini che si vogliono visualizzare inserendo un numero da 1 a 9 nell'apposito campo.

La figura 3.2 mostra la finestra pop up di Instagram appena descritta.



Instagram

Who or what do you wanna search?

Use "@" for usernames and "#" for tags.

@ssi_international

How many posts do you wanna see?

Right now the max is 9.

4

Generate feed



Figura 3.2: Modal di Instagram

Una volta caricata l'immagine l'utente può procedere con la predizione tramite il pulsante *Predict*.

Accanto all'immagine caricata appariranno, dunque, i risultati della predizione i quali comprendono il nome della classe e la probabilità percentuale che sia giusta. Vengono considerati soltanto i 5 risultati con la probabilità più alta.

Come mostrato in figura 3.3, è possibile salvare la predizione per poterla rivedere in futuro tramite il pulsante *Save Prediction*.

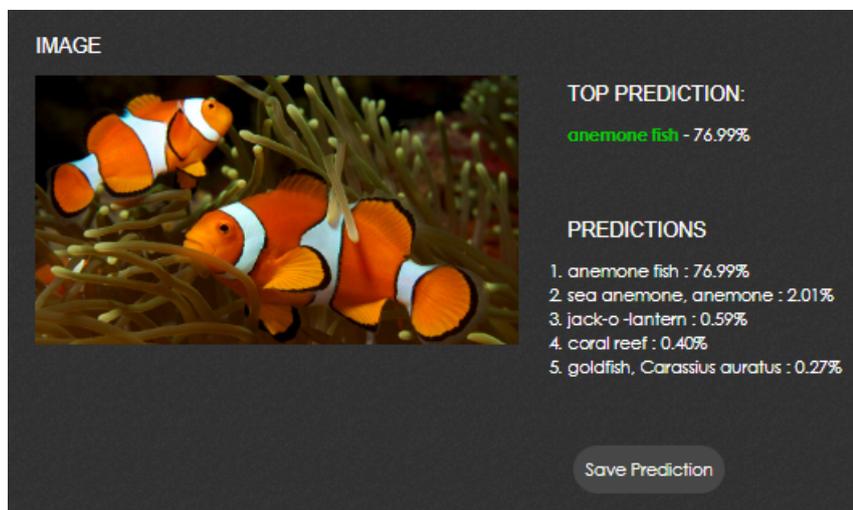


Figura 3.3: Predizione immagine scelta

3.2.2 Pagina delle predizioni

L'utente può visualizzare le predizioni che ha salvato spostandosi dal menu alla voce Predictions dove le immagini sono classificate in tre gruppi:

- VERIFIED: per le predizioni la cui probabilità è superiore o pari a 70%;
- PENDING: per le predizioni la cui probabilità è superiore o pari a 40%;
- UNCERTAIN: per le predizioni la cui probabilità è minore a 40%.

Utilizzando il pulsante *Show stored images* l'utente visualizza tutte le predizioni che ha salvato, mentre utilizzando il pulsante *Clear all* è possibile eliminare tutte le predizioni.

Come mostrato in figura 3.4, le predizioni sono divise in tre colonne ben distinte e per ogni predizione l'utente ha la possibilità di visualizzare l'immagine, la classe di appartenenza e la probabilità.

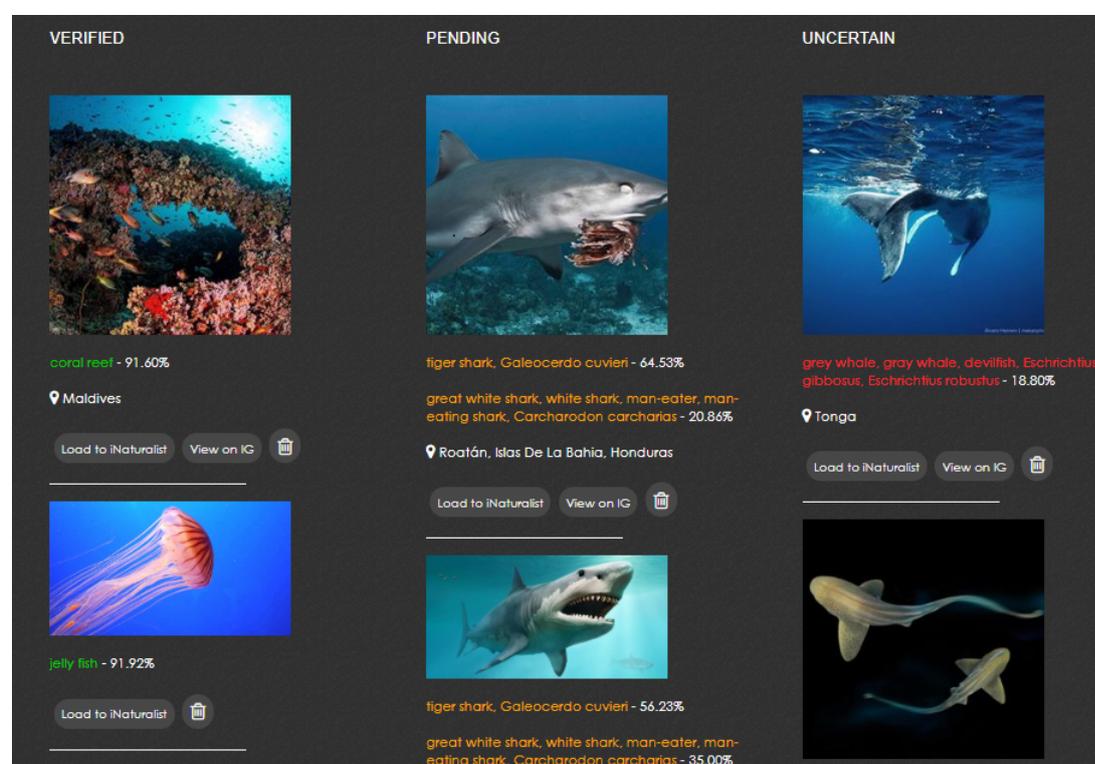


Figura 3.4: Pagina delle predizioni

Ogni predizione dispone inoltre di altri 2 pulsanti:

- *Load to iNaturalist*: per collegarsi all'omonimo sito;
- L'icona del *cestino*: per eliminare il singolo elemento.

Per le immagini che provengono da Instagram è possibile visualizzare la localizzazione, se presente e il pulsante *View on IG* che consente di visualizzare l'immagine sulla pagina Instagram da cui proviene.

3.2.3 Pagina della mappa

Spostandosi dal menu alla voce MAP l'utente può visualizzare una mappa di Google nella quale sono presenti tutti i marker delle immagini caricate sulla piattaforma provenienti da Instagram. Le immagini caricate che possiedono la localizzazione vengono infatti mappate automaticamente al momento del salvataggio.

Come mostra la figura 3.5 cliccando sui marker appare una piccola finestra con le informazioni dell'immagine mappata in quel luogo.

I vari marker hanno colori diversi in base alla categoria di appartenenza dell'immagine.

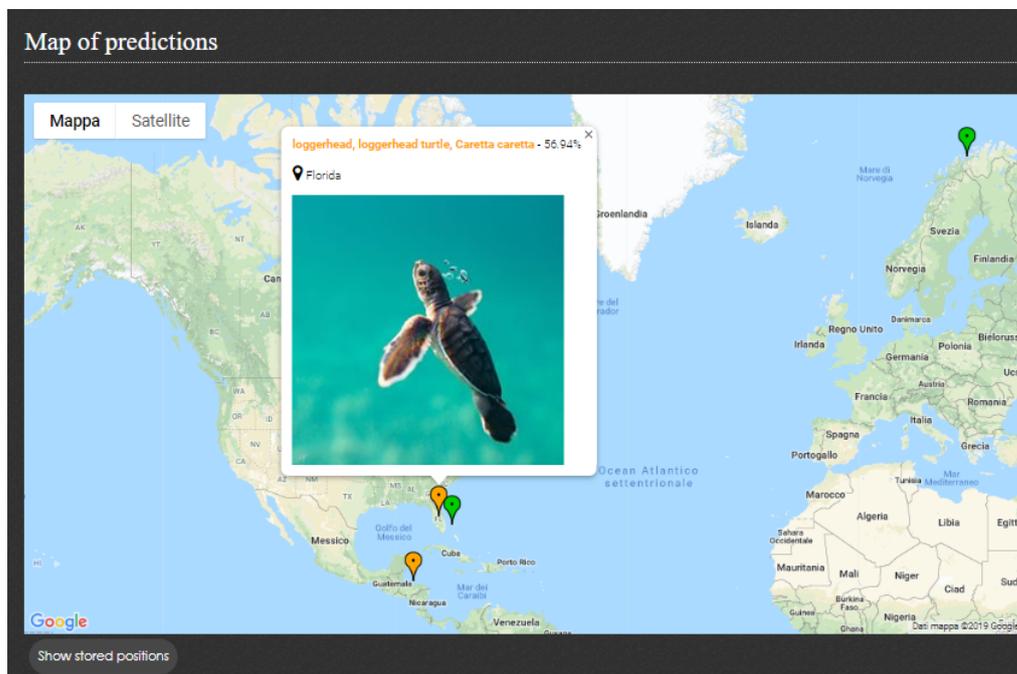


Figura 3.5: Mappa delle predizioni

Capitolo 4

Implementazione

In questo capitolo viene discussa l'implementazione della piattaforma realizzata.

Nel paragrafo 4.1 vengono introdotte le tecnologie utilizzate nello sviluppo del sistema, mentre nel paragrafo 4.2 vengono discusse le scelte implementative.

4.1 Tecnologie utilizzate

4.1.1 Javascript e Node.js

Javascript [11] è uno dei principali linguaggi di programmazione per lo sviluppo di applicazioni web, nato come linguaggio di scripting client-side ma utilizzato attualmente anche per gli aspetti di programmazione server-side.

Un linguaggio di scripting è un linguaggio di programmazione che è utilizzato per manipolare, personalizzare e automatizzare le funzioni di un sistema esistente. [12]

Lo standard JavaScript è ECMAScript, un linguaggio di programmazione orientato agli oggetti.

Tramite apposite librerie è possibile estendere Javascript. Per realizzare la piattaforma è stata utilizzata la libreria JQuery, utile per la manipolazione di documenti HTML e la gestione degli eventi.

Node.js [13] è uno dei framework Javascript più utilizzati. Permette di realizzare applicazioni web in linguaggio Javascript, tipicamente utilizzato nel client-side, anche per la scrittura di applicazioni server-side.

Express [14] è un framework per applicazioni web Node.js che fornisce una serie di funzioni avanzate per le applicazioni web.

Javascript e Node.js sono stati utilizzati per realizzare la piattaforma rispettivamente per la parte client-side e quella server-side.

4.1.2 HTML, CSS e Bootstrap

HTML, acronimo di HyperText Markup Language, è un linguaggio di markup che permette di indicare come disporre gli elementi all'interno della pagina e stabilire collegamenti tra le pagine. Queste indicazioni vengono date attraverso appositi marcatori, detti tag. Attraverso i tag vengono indicati, quindi, quali elementi dovranno apparire sullo schermo e come essi debbano essere disposti.

CSS, detto anche “foglio di stile”, comprende una serie di regole che permettono di definire l'aspetto che devono assumere gli elementi sulla pagina. Può quindi manipolare dimensioni, colori, animazioni e ogni altra caratteristica visuale.

Bootstrap [15] è un insieme di strumenti open source che mette a disposizione modelli grafici utili a modificare l'aspetto di una pagina web.

HTML, CSS e Bootstrap sono stati utilizzati per la realizzazione dell'interfaccia della piattaforma.

4.1.3 Instagram API

Le API, Application Programming Interface, tradotto come interfaccia di programmazione di un'applicazione, sono strumenti di programmazione messi a disposizione degli sviluppatori per facilitare la realizzazione di applicazioni di vario genere.

Instagram [16] mette a disposizione diverse API basate sui servizi che offre. Purtroppo negli ultimi tempi una gran parte delle API Instagram è stata deprecata e rimossa e la maggior parte degli endpoint è stata chiusa a causa della sicurezza delle informazioni dell'utente.

Nella realizzazione della piattaforma sono state utilizzate tecniche per aggirare questi limiti e ottenere così i post relativi ad hashtag e profili pubblici.

4.1.4 TensorFlow.js

TensorFlow.js [17] è una libreria Javascript per la formazione l'implementazione di modelli di apprendimento automatico. E' possibile utilizzare questa libreria per definire, addestrare ed eseguire modelli di apprendimento automatico interamente nel browser, utilizzando Javascript. Includendo il seguente script nell'intestazione del file html il programma è pronto per essere eseguito.

```
1 <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@0.12.0"> </script>
```

Per realizzare la piattaforma è stato utilizzato TensorFlow.js per eseguire un modello pre-addestrato Keras MobileNet.

4.1.5 Keras MobileNet

Keras [18] è una libreria open source scritta in Python per l'apprendimento automatico. E' utilizzata per costruire e addestrare modelli.

Keras dispone di diverse applicazioni. Le applicazioni Keras sono modelli di apprendimento approfondito messi a disposizione insieme a pesi preaddestrati. Questi modelli possono essere utilizzati per la previsione e l'estrazione di caratteristiche.

Per realizzare la piattaforma è stato utilizzato un modello Keras addestrato per la classificazione delle immagini chiamato MobileNet. La dimensione di input predefinita per questo modello è 224x224. Esso dispone di pesi preaddestrati su ImageNet. ImageNet è un database di immagini organizzato secondo la gerarchia WordNet, un database lessicale. Ogni nodo della gerarchia è rappresentata da centinaia di migliaia di immagini.

E' stato scelto MobileNet perchè è leggero nella sua architettura. Questa architettura utilizza reti neurali, ossia modelli matematici composti da "neuroni", che si ispirano a una rete neurale biologica. Un neurone riceve in ingresso segnali da vari altri neuroni tramite connessioni fino ad arrivare ai neuroni di output.

Quando si addestra una rete neurale su un set di immagini, esse vengono passate attraverso la rete di connessioni applicando diversi filtri su ciascun livello dell'immagine. Una volta arrivate ai neuroni di output si può trovare con precisione la classe a cui appartiene l'immagine.

I filtri sui primi strati della rete imparano a riconoscere i colori e alcune linee orizzontali e verticali. Quelli successivi imparano lentamente a riconoscere forme banali usando le linee e colori appresi negli strati precedenti. Quindi i livelli ancora successivi imparano a conoscere parti di oggetti come gambe, occhi, naso, ecc. Infine negli ultimi strati vengono riconosciuti oggetti interi come cani, automobili, ecc. [19] La figura 4.1 mostra neuroni di output che la rete neurale utilizza per classificare le immagini.

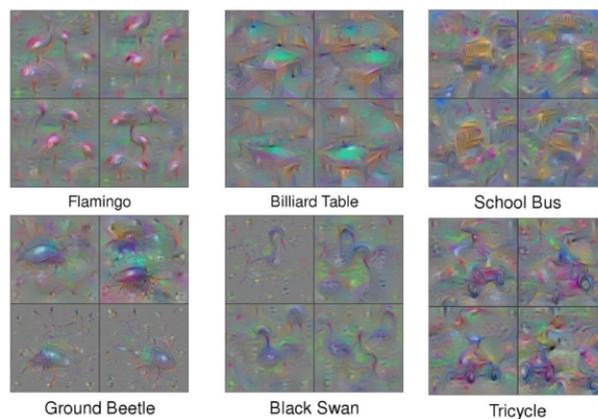


Figura 4.1: Neuroni di output [20]

4.2 Scelte implementative

4.2.1 Caricamento del modello

Il caricamento del modello MobileNet nel browser avviene tramite l'apposito pulsante *Load Model*, situato nella pagina principale della piattaforma. E' stata utilizzata una funzione asincrona che carica un modello di keras json usando *tf.loadModel()*. Il comando *await* permette il caricamento “dietro le quinte”, senza disturbare l'interfaccia utente.

Il seguente codice mostra l'implementazione del caricamento del modello.

```
1 let model;
2 async function loadModel() {
3   // clear the model variable
4   model = undefined;
5   // load the model (where you have stored your model files)
6   model = await tf.loadModel('http://localhost:8081/mobilenet
7     /model.json');
```

4.2.2 Caricamento delle immagini

Dal PC

Per caricare le immagini dal pc è stato utilizzato l'oggetto *FileReader*. Come mostra il codice seguente, è stata creata un'istanza di questo oggetto, quindi è stato assegnato un gestore all'evento *load* ed invocato il metodo *readAsDataURL()* passandogli l'oggetto file da leggere.

```
1 let reader = new FileReader();
2 reader.onload = function(){
3     let dataURL = reader.result;
4     $("#selected-image").attr("src",dataURL);0
5 }
6 let file = $("#image-selector").prop('files')[0];
7 reader.readAsDataURL(file);
8 document.getElementById("image-selector").value = "";
```

Da Instagram

Il caricamento delle immagini da instagram avviene per mezzo di un Modal che si apre cliccando il pulsante *Search on Instagram*.

In base al carattere inserito prima del nome è possibile cercare un utente o un tag, utilizzando rispettivamente i caratteri @ o #.

E' quindi necessario utilizzare due link url diversi:

```
1 switch (type) {
2     case 'user':
3         searchUrl = 'https://www.instagram.com/' + search;
4         break;
5     case 'tag':
6         searchUrl = 'https://www.instagram.com/explore/tags/' +
7             search + '/?__a=1';
8         break;
9 }
```

Viene utilizzato il metodo jQuery `$.get()` per richiedere i dati dal server, specificando l'url giusto. Successivamente, viene letto il file json e prelevate le informazioni necessarie.

Come viene mostrato nel seguente codice, le immagini caricate nel Modal sono pulsanti. Questo per fare in modo che l'utente possa scegliere l'immagine che preferisce tra quelle caricate.

```
1 if(type == 'user') {
2   var datas = JSON.parse(data.split("window._sharedData = "
   ) [1].split("</script>") [0]).entry_data.ProfilePage
   [0].graphql;
3   if(datas.user.edge_owner_to_timeline_media.edges == ""){
4     $('<posts>').html('The user has a private profile.<br>
   ');
5   } else {
6     var $this = datas.user.edge_owner_to_timeline_media.
       edges[i].node;
7     var a = $this.thumbnail_resources[1].src;
8     var b = $this.shortcode;
9     $('<posts>').append('<button onclick=\"sel(\'' + a + '
   \', \'' + b + '\')\"><img src=\"'+a+'\"></button>');
10  }
11 }
```

4.2.3 Elaborazione dell'immagine

L'elaborazione dell'immagine viene effettuata per rendere le dimensioni dell'immagine accessibili a MobileNet. La dimensione dell'immagine di input deve essere 224x224.

L'immagine viene quindi caricata usando `tf.fromPixels()`, ridimensionata usando `resizeNearestNeighbor()` e i suoi valori convertiti in float usando `toFloat()`. Sono stati così ridimensionati i valori nel tensore dell'immagine utilizzando un valore scalare di 127,5, che è il valore centrale dell'intervallo di pixel dell'immagine [0, 255].

Ad ogni valore di pixel viene sottratto il valore di offset e viene per questo valore di offset in scala tra [-1, 1].

Quindi vengono espanse le dimensioni usando *expandDims()*.

```
1 let tensor = tf.fromPixels(image)
2   .resizeNearestNeighbor([224, 224])
3   .toFloat();
4 let offset = tf.scalar(127.5);
5 return tensor.sub(offset)
6   .div(offset)
7   .expandDims();
```

4.2.4 Predizione

Dopo aver elaborato l'immagine è possibile fare la predizione di essa utilizzando il pulsante *Predict*. La predizione viene fatta attraverso *model.predict(tensor)* a cui vengono passati i dati. I risultati delle predizioni vengono salvati su un array chiamato *results*. L'array viene poi ordinato in base alla probabilità più alta usando *sort()* e vengono prese soltanto le prime 5 probabilità usando *slice()*.

```
1 let tensor = preprocessImage(image, modelName);
2 let predictions = await model.predict(tensor).data();
3 let results = Array.from(predictions)
4   .map(function (p, i) {
5     return {
6       probability: p*100,
7       className: IMAGENET_CLASSES[i]
8     };
9   }).sort(function (a, b) {
10    return b.probability - a.probability;
11  }).slice(0, 5);
```

4.2.5 Web storage

Per il salvataggio dei dati è stato utilizzato il web storage locale in modo che i dati persistano anche dopo la chiusura della finestra del browser.

Per salvare la predizione di un'immagine occorre cliccare il pulsante *Save Prediction* che si trova nella pagina principale.

Gli elementi salvati sono mostrati nella *Pagina delle predizioni*. Una volta cliccato il pulsante, la predizione viene inserita in un array chiamato *itemsArray*. Questo array viene poi aggiunto al localStorage con una chiave.

```
1 itemsArray.push(item);
2 localStorage.setItem('items', JSON.stringify(itemsArray));
```

Per fare in modo che *itemsArray* non venga reimpostato su un array vuoto ogni volta che viene eseguito lo script, viene fatta una dichiarazione condizionale che controlli se *localStorage* esiste già.

```
1 let itemsArray = localStorage.getItem('items') ? JSON.parse(
  localStorage.getItem('items')) : [ ];
```

Inoltre, per cancellare i dati salvati, è possibile cliccare sull'icona del cestino presente sotto ogni predizione salvata. In questo modo verrà eliminato il singolo elemento dall'array. In alternativa è possibile eliminare tutti i dati salvati tramite il pulsante *Clear all*.

4.2.6 Mappa

Per visualizzare la mappa è stata utilizzata l'API JavaScript di Maps, caricata utilizzando il seguente script tag in cui è stata inserita la chiave API.

```
1 <script async defer src="https://maps.googleapis.com/maps/api
  /js?key=[API_KEY]
2 &callback=initMap"> </script>
```

Per aggiungere i vari marker vengono recuperate dal localStorage la latitudine e la longitudine delle immagini di Instagram che dispongono di localizzazione.

Tramite la funzione *addInfoWindow* viene inserito il marker con la relativa finestra di dialogo che mostra le informazioni di quella posizione.

```
1 var mark = new google.maps.Marker({position: pos, map: map,
   icon: pinImage});
2 var contentString = '<div style="color:black">'
3   + '<b style="color:'+pinColor+'">' + data[i].cn1 + '</b>'
4     + ' - ' + data[i].pb1 + '%',
5   + '<br><br> <i class="fa fa-map-marker" style="font-size'
6     + ':20px"></i> ' + data[i].locN
7     + '<br><br>'
8     + '</div>';
9 addInfoWindow(mark, contentString);
```

Capitolo 5

Conclusioni e lavori futuri

Questa tesi ha trattato l'implementazione di una piattaforma per il riconoscimento e la classificazione delle immagini, fornendo anche una funzionalità per caricare le immagini da Instagram e inserire la loro posizione in una mappa.

E' infatti stata implementata una piattaforma software che permette agli utenti di fare predizioni sulle proprie foto e quelle di altri utenti.

Il design della piattaforma è semplice e intuitivo, in modo che sia utilizzabile da tutti. Una volta caricati il modello e l'immagine, infatti, la piattaforma si occupa di fare la predizione in totale autonomia, fornendo anche l'opzione di salvare la predizione appena fatta.

Poiché la piattaforma permette di caricare immagini da Instagram è presente anche una mappa che contiene i marker delle immagini salvate che dispongono di localizzazione.

La piattaforma, infine, permette di collegarsi al sito di iNaturalist. Sotto ogni predizione salvata è infatti presente l'apposito pulsante Load to iNaturalist.

I limiti che riguardano questa piattaforma riguardano soprattutto l'estrazione dei dati da Instagram, dato che l'API di questo social media non consente il caricamento di più di 9 post per ogni ricerca. Inoltre non è possibile im-

postare un intervallo di tempo dal quale estrarre i post, ma vengono sempre visualizzati gli ultimi 9 post più recenti per ogni hashtag o utente ricercato. Un altro limite riguarda invece la mappa caricata con l'API Javascript di Maps. Quest'API infatti, richiede di abilitare la fatturazione sul proprio account per assicurarsi che l'utente non sia un robot. Se questa opzione non è abilitata è possibile fare soltanto una richiesta al giorno per quest'API. Nonostante questo, la mappa e i suoi marker sono sempre visualizzabili.

Gli sviluppi futuri riguardano soprattutto la possibilità di caricare le proprie predizioni sul sito di iNaturalist, aggiungendo le proprie osservazioni con predizioni che sono quasi certe.

Inoltre, per quanto riguarda la predizione delle immagini, si potrebbe sostituire il modello Keras MobileNet pre-addestrato utilizzato per questa piattaforma con un nuovo modello più specializzato. Questo significa che si potrebbe addestrare il modello in maniera più approfondita facendo in modo che riconosca, ad esempio, diverse specie della stessa pianta o dello stesso animale. Si potrebbero, così, escludere le predizioni che riguardano oggetti, come un tavolo o una sedia, rendendo il modello più adatto a riconoscere esclusivamente elementi che fanno parte della natura. In questo modo anche il caricamento della predizione su iNaturalist sarebbe più preciso e quindi più utile ad altri utenti e a fini di ricerca.

Bibliografia

[1] Citizen Science.

<http://www.scienzainrete.it/articolo/citizen-science-scienza-di-tutti/valentina-meschia/2016-03-10>

[2] Nature Ecology Evolution.

<https://www.nature.com/articles/s41559-018-0608-1>

[3] National Geographic.

http://www.nationalgeographic.it/natura/animali/2018/07/25/news/il_rovescio_della_medaglia_4057844/?refresh_ce

[4] DUEproject.

<http://dueproject.org/en/wp-content/uploads/2018/01/20170417ScienceHunter.pdf>

[5] DUEproject.

<http://dueproject.org/en/research/>

[6] iNaturalist.

<https://www.inaturalist.org/observations>

[7] iNaturalist.

<https://www.innaturale.com/inaturalist-unapplicazione-riconoscere-gli-animale/>

[8] iNaturalist.

https://www.inaturalist.org/pages/computer_vision_demo

- [9] Image Recognition.
<https://searchenterpriseai.techtarget.com/definition/image-recognition>
- [10] Image Recognition.
<https://www.marutitech.com/working-image-recognition/>
- [11] Javascript.
<https://www.javascript.com/>
- [12] ECMA International.
<https://www.ecma-international.org/>
- [13] Node.js.
<https://nodejs.org/>
- [14] Express.
<https://expressjs.com/>
- [15] Bootstrap.
<https://getbootstrap.com/>
- [16] Instagram.
<https://www.instagram.com/>
- [17] TensorFlow.js
<https://js.tensorflow.org/>
- [18] Keras.
<https://keras.io/>
- [19] MobileNet.
<https://towardsdatascience.com/keras-transfer-learning-for-beginners-6c9b8b7143e>
- [20] Neural Network.
<http://yosinski.com/deepvis>

Ringraziamenti

Ringrazio innanzitutto il Dott. Luca Bedogni, relatore di questa tesi, per la disponibilità, i consigli e il tempo che mi ha dedicato.

Un grande ringraziamento va a tutti i miei amici e, soprattutto, alla mia migliore amica che mi è sempre stata vicina e mi ha sopportata e compresa in tutti questi anni.

Ringrazio anche i miei colleghi di corso, che hanno reso questi 3 anni più piacevoli e leggeri riempiendoli di risate.

Il ringraziamento più grande va alla mia famiglia, che è stata e sarà sempre il mio sostegno più grande.

