

UNIVERSITÀ DEGLI STUDI DI SALERNO



Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica applicata

Corso di Laurea in Ingegneria Informatica

ELABORATO DI TESI

Progettazione e realizzazione di un portale web per il conteggio
automatico dei volti

Primo Relatore:

Prof. Antonio GRECO

Secondo Relatore:

Prof.ssa Alessia SAGGESE

Candidato:

Mattia MARSEGLIA

Matricola:

0612704430

Anno Accademico 2020/2021

*“Le cose migliori si ottengono solo con il
massimo della passione”*

GOETHE

Sommario

<i>Introduzione</i>	<i>3</i>
1. Descrizione ed Analisi della realtà di interesse	5
1.1. Descrizione.....	5
1.2. Analisi dei requisiti	6
1.2.1. Glossario dei termini	6
1.2.2. Strutturazione delle frasi.....	7
1.2.3. Identificazione delle operazioni principali	7
1.3. Classificazione dei requisiti	8
1.3.1. Requisiti funzionali	8
1.3.2. Requisiti non funzionali.....	9
1.4. Casi d'uso	11
2. Progettazione	12
2.1. Progettazione del database	12
2.1.1. Progettazione Concettuale	13
2.1.2. Progettazione Logica	17
2.1.3. Normalizzazione.....	22
2.2. Progettazione delle pagine	24
2.2.1. Homepage	24
2.2.2. Registrazione.....	25
2.2.3. Test	25
2.2.4. Result	26
3. Implementazione.....	27
3.1. Implementazione del back-end.....	27
3.1.1. Implementazione del database	27
3.1.2. Implementazione dell'applicazione.....	30
3.2. Implementazione del Front-end	37
3.2.1. Implementazione pagina di registrazione:	38
3.2.2. Implementazione pagina di homepage	39
3.2.3. Implementazione della pagina per il test e il risultato	40
4. Descrizione delle pagine realizzate	42
4.1. Homepage	42
4.2. Registrazione	50
4.3. Test	51
4.4. Result	53
<i>Conclusione</i>	<i>54</i>
<i>Bibliografia e sitografia</i>	<i>55</i>

Introduzione

L'elaborato di tesi che presento ha inteso realizzare la progettazione e lo sviluppo di un portale web che, classificando immagini, consente di testare l'efficienza di reti neurali, strumento di supporto oramai indispensabile in ambiti tecnologici eterogenei (elettronica, informatica, simulazione ed altre discipline) per risolvere problemi ingegneristici di intelligenza artificiale.

Il percorso che mi ha condotto all'esito della ricerca si è alla fine soffermato su di un particolare aspetto di detta classificazione, che ha riguardato il cd. "*Face Counting*" e, conseguentemente, le reti neurali e le categorie concettuali ad esso associate.

Il cd. "*conteggio delle facce*" negli ultimi anni sta infatti ricevendo molti riscontri positivi soprattutto nell'ambito applicativo della foto-video analisi intelligente, a supporto della Business Intelligence: l'estrazione di dati da flussi foto o video consente, all'interno dei processi adottati nello sviluppo di un'attività produttiva e non solo, di raccogliere dati ed analizzare informazioni strategiche, con l'obiettivo, da una parte, di analizzare la storia passata, al fine di comprendere gli elementi che hanno determinato le performance conseguite, e, dall'altra parte, di stimare le prestazioni future del business, attraverso analisi di serie storiche e/o mediante simulazione di diversi scenari.

Il settore Retail è conseguentemente l'ambito applicativo ove gli algoritmi di foto-video analisi intelligente per la Business Intelligence oggi riescono a trovare la più ampia penetrazione. In esso, infatti, oltre alla ovvia esigenza di impiegare la foto-video sorveglianza intelligente per la protezione di beni e persone, il "*face counting*" è largamente impiegato per estrarre informazioni relative al comportamento dei consumatori. Tra le diverse funzioni di analisi video o foto, vi è quella di contare il numero di volti che ad esempio attraversano un varco mediante l'interpretazione di flussi ottenuti da telecamere montate in posizione verticale rispetto al punto di accesso. In pratica la telecamera perde la sua funzione di sorveglianza visiva per divenire, corredata di un software dedicato, un sensore dedicato al conteggio dei volti che attraversano una linea virtuale collocata sul terreno posta in corrispondenza del varco.

In una pillola futura capiremo veramente quanto l'informazione relativa al numero di persone che transitano attraverso i varchi di un negozio o piuttosto il numero di persone presenti all'interno di un'area o, ancora, il loro tempo di permanenza in un determinato posto, possano attraverso uno studio statistico, condizionare dati della realtà, come ad esempio quelli relativi alle vendite, fornendo informazioni sulla tipologia di clienti nei diversi periodi del giorno e della settimana e di conseguenza consentendo di poter realizzare offerte personalizzate. Un'utilità di dati, allora, che diventa addirittura preziosa se la si ricolleggi ad esempio in termini di ottimizzazione delle risorse di personale con una allocazione legata al numero di utenti previsti, di riduzione dei costi di esercizio (climatizzazione) e di miglioramento dei servizi (riduzione o eliminazione delle code alle casse, più addetti alle informazioni, ecc).

Va tuttavia evidenziato in questa sede che il presente lavoro ha perseguito un ulteriore obiettivo, ovvero quello di porre l'attenzione alla cura del portale stesso, del quale è stata eseguita scrupolosamente l'analisi delle funzionalità, arricchite per meglio adattarle allo scenario tecnico

contemporaneo, anziché considerarlo, come per lo più è stato fatto sinora, come lavoro di contorno rispetto alla rete neurale stessa.

In letteratura, infatti, l'attenzione principalmente rivolta alla possibilità di testare delle reti neurali, come ad esempio quella per capire come funzionano gli algoritmi di conteggio dei volti e le principali problematiche che essi devono affrontare porta, solo come corollario, la realizzazione di una tecnologia di test, che verrà in tal caso esclusivamente finalizzata all'uso specifico previsto, e solo con le funzionalità minime per detto uso. Tuttavia, la criticità di detta procedura riposa nella circostanza che la relativa verifica degli esiti è minimale e circoscritta a quella specifica rete neurale realizzata, limitando l'eventuale confronto con altre verifiche, che per essere testate richiederebbero lo sviluppo di un'ulteriore tipologia di test.

Ed invece, con la realizzazione di un portale, come quello in questione, che sia stato già predisposto per adattarsi a diverse reti neurali, offrendo diverse tipologie di classificazioni e dunque di input da valorizzare, si potrà ottenere il risultato di una più completa ed ottimizzata memorizzazione dei dati d'interesse, oltre che un più semplice ed immediato confronto anche tra classi differenti di reti neurali.

Struttura della tesi

Capitolo 1: presenta una descrizione ed una analisi articolata della realtà di interesse;

Capitolo 2: riporta tutte le progettazioni volte alla realizzazione del portale;

Capitolo 3: espone l'implementazione del portale in ogni sua parte;

Capitolo 4: descrive dettagliatamente le pagine realizzate anche attraverso immagini;

Nell'augurarmi di essere riuscito ad interessarvi con l'introduzione al lavoro effettuato e, soprattutto, ad incuriosirvi,

vi auguro Buona lettura!

1. Descrizione ed Analisi della realtà di interesse

L'analisi dei requisiti del sistema software in questione è stata condotta in maniera estremamente accurata al fine di facilitare la fase di progettazione e realizzazione dell'applicativo. Questa fase preliminare di analisi, infatti, ha fornito una visione più ampia e dettagliata delle specifiche e degli obiettivi da perseguire.

1.1. Descrizione

Si vuole progettare e sviluppare un portale web che permetta ad un utente di caricare un'immagine e di classificarla in base alle sue esigenze. La classificazione viene effettuata mediante una serie di reti neurali. Le immagini sono caratterizzate da una macrocategoria (*Face Counting*, *People Counting*, *Vehicle Counting* e *Face Analysis*) e da un insieme di categorie. Per le macrocategorie *Face Counting*, *People Counting* e *Vehicle Counting* le categorie corrispondenti sono rispettivamente numero di volti, di persone e di veicoli mentre per la macrocategoria *Face Analysis* le categorie sono l'età, il genere, l'etnia e l'emozione del volto ritratto nell'immagine. Le macrocategorie e le categorie sono predefinite.

L'utente semplice del portale (gestito da uno o più amministratori), effettuando la propria registrazione potrà, una volta acceduto al proprio profilo, caricare un numero illimitato di immagini, delle quali poi, la rete neurale ne realizzerà la classificazione. La tipologia di classificazione viene quindi filtrata attraverso le macrocategorie disponibili all'interno del portale.

In particolare, una volta selezionata la macrocategoria e caricata l'immagine, sarà necessario valorizzare la/e categoria/e associate, inserendone i dati richiesti e selezionare una delle reti neurali disponibili per quella macrocategoria. Solo alla fine di tale procedura verrà eseguita la relativa classificazione da parte della rete neurale, che il portale mostrerà all'utente insieme all'immagine caricata, alla categoria attesa ed alle informazioni associate alla rete precedentemente selezionata, quali le prestazioni e il *threshold* scelto. Le prestazioni sono performance medie, calcolate dal portale, ed ottenute da una specifica rete neurale su tutte le immagini classificate.

Ad ogni tentativo l'immagine caricata viene salvata e il sistema conserva le categorie reali e i risultati delle classificazioni

1.2. Analisi dei requisiti

1.2.1. Glossario dei termini

Il glossario che segue semplifica la comprensione dei termini rilevanti per la realtà di interesse, in particolare per tutti quelli considerati è stata inserita:

- una breve descrizione
- i possibili sinonimi
- i collegamenti agli altri termini rilevati

	Termine	Descrizione	Sinonimi	Collegamenti
1	Immagine	Fotografia virtuale scelta dall'utente per sottoporla a classificazione.	-	Macrocategoria, classificazione
2	Macrocategoria	Elemento costitutivo di una macrocategoria.	-	Categoria, Immagine, Rete neurale
3	Categoria	Il cliente che porta l'auto presso l'officina. Può essere il proprietario dell'auto o solo il conducente.	-	Macrocategoria, classificazione
4	classificazione	Operazione di specificazione delle caratteristiche di un'immagine effettuata da una rete neurale.	Tentativo	Rete neurale, immagine, categoria
5	Rete neurale	Modello computazionale che classifica immagini di una determinata macrocategoria	-	Macrocategoria, classificazione

1.2.2. Strutturazione delle frasi

La seguente strutturazione mira a chiarire, in maniera univoca, tutto ciò che è associato ad ogni termine rilevato nel glossario precedentemente costruito.

1) Frasi di carattere generale

Si vuole progettare e sviluppare un portale web che permetta ad un utente di caricare un'immagine e di classificarla in base alle sue esigenze.

2) Frasi relative a immagine

Per l'immagine rappresentiamo l'URL, la valorizzazione delle categorie associate effettuata dall'utente.

3) Frasi relative a macrocategoria

Per la macrocategoria vogliamo rappresentare il nome, la descrizione, le categorie e le reti neurali associate.

4) Frasi relative a categoria

Relativamente alla categoria, rappresentiamo il nome e la macrocategoria di appartenenza.

5) Frasi relative a classificazione

Per la classificazione è di interesse rappresentare la rete neurale da cui è stata effettuata, il valore espresso dalla rete neurale, il *threshold* selezionato dall'utente, le categorie e l'immagine a cui fa riferimento.

6) Frasi relative a rete neurale

Per la rete neurale rappresentiamo il nome, la data di ultimo riprestino, la descrizione, le prestazioni per le categorie a cui fa riferimento e la macrocategoria a cui è associata.

1.2.3. Identificazione delle operazioni principali

1) Operazione 1:

Aggiornamento delle prestazioni di una rete neurale (in media 20 volte al giorno)

2) Operazione 2:

Stampa i valori delle prestazioni di una rete neurale (in media 40 volte al giorno)

3) Operazione 3:

Caricamento di una nuova immagine e valorizzazione delle categorie annesse (in media 30 volte al giorno)

4) Operazione 4:

Selezione di una rete neurale (in media 30 volte al giorno)

5) Operazione 5:

Stampa per ciascuna macrocategoria le categorie e le reti neurali associate (in media 75 volte al giorno)

6) Operazione 6:

Stampa il risultato della classificazione effettuata da una rete neurale su una specifica immagine (in media 10 volte al giorno)

7) Operazione 7:

Registrazione di un nuovo utente (in media 2 volte al giorno)

1.3. Classificazione dei requisiti

In questa ultima parte di analisi, forti della chiarezza espressiva ottenuta nelle sezioni precedenti, sarà condotta una classificazione ben precisa tra requisiti funzionali e non da rispettare nei capitoli successivi fino alla realizzazione completa del portale.

1.3.1. Requisiti funzionali

Scelta della macrocategoria per cui eseguire la classificazione

All'utente registrato ed autenticato il portale deve offrire la possibilità, nella homepage, di selezionare la macrocategoria per cui eseguire la classificazione; nella stessa pagina, infine, la stessa operazione non dovrà essere concessa a chi non ha eseguito l'accesso.

Inserimento dei dati precedenti la classificazione

Il portale deve consentire all'utente autenticato che ha selezionato una macrocategoria quanto segue: caricare un'immagine, valorizzare le categorie associate alla macrocategoria scelta precedentemente e selezionare una delle reti neurali disponibili per quella tipologia di classificazione.

Modifica dei valori inseriti prima della classificazione

Prima della classificazione il portale deve offrire la possibilità, all'utente che sta completando il test, di modificare qualunque valore inserito fino a quel momento.

Gestione delle prestazioni

Il portale deve gestire le prestazioni associate ad ogni rete neurale in modo da mostrare all'utente un valore sempre aggiornato correttamente; questo aggiornamento, in particolare, deve avvenire in una modalità automatica.

Registrazione a valle della classificazione

Il portale web deve garantire un salvataggio automatico sia dell'immagine e dei dati inseriti dall'utente in fase di test, che dei valori forniti dalla rete neurale a valle della classificazione. In questo modo si potrà risalire a qualunque informazione rilevante che riguarda le classificazioni operate da ogni rete neurale.

Gestione degli utenti

In ogni momento il portale deve consentire all'amministratore la rimozione di un determinato utente o la modifica dei suoi dati e dei suoi permessi.

Gestione dei dati

In caso di errore dovrà essere possibile, per l'amministratore, rimuovere o modificare qualunque valore memorizzato.

Gestione delle descrizioni

Il portale deve consentire all'amministratore di modificare direttamente, per quanto possibile, la maggior parte delle descrizioni presenti nel sito nella sezione apposita, sia delle categorie che delle macrocategorie e delle reti neurali.

Il portale deve poter essere auto esplicativo

Il nuovo utente del portale deve poter comprendere, interagendo con lo stesso, quali sono le finalità ed avere la possibilità, in caso di necessità di ulteriori chiarimenti, di contattare un servizio informativo.

1.3.2. Requisiti non funzionali

Manutenibilità

Il portale deve essere facilmente modificabile e garantire che le funzioni implementate siano facilmente accessibili da aree ben distinte. L'obiettivo di tale requisito deve essere proprio la possibilità, una volta identificata una correzione o una modifica da apportare, di individuare

facilmente la porzione di codice interessata ed effettuare i cambiamenti solo su di essa. Tale manutenibilità non deve riguardare solo il codice ma anche le descrizioni contenute nel portale stesso che, quando possibile, si suppongono essere modificabili senza necessariamente modificare il codice.

Portabilità

Il portale deve essere fruibile da quanti più dispositivi possibili e per questo motivo è opportuno realizzare un design responsive dello stesso.

Semplicità di gestione

Il portale deve essere gestibile in modo semplice e completo dagli amministratori che devono poter inserire, modificare e cancellare facilmente le informazioni di interesse nelle apposite sezioni.

Completezza

Il portale deve comprendere tutti i requisiti emersi nell'analisi, senza mancanze o impossibilità.

1.4. Casi d'uso

Dopo l'accurata analisi del sistema, svolta precedentemente, si evince che i possibili fruitori del sito sono due: l'utente, che mira a testare le varie reti neurali, e l'amministratore, che gestisce i contenuti del portale e gli utenti, le loro richieste e tutto ciò che è associato ad essi.

E' possibile quindi identificare i seguenti casi d'uso possibili:

Consultazione del portale senza autenticazione:

in questo caso ad interfacciarsi al portale vi è un utente non autenticato. La particolarità di tale caso d'uso, sebbene banale, sta nel fatto che all'utente non è concessa alcuna possibilità di svolgere funzioni, se non quella di raccogliere informazioni in merito al portale web o di autenticarsi.

Esecuzione del processo finalizzato alla classificazione di una immagine ad opera di una rete neurale:

il caso d'uso considerato riguarda utenti autenticati, i quali, selezionando una macrocategoria, eseguono le operazioni finalizzate a classificare un'immagine tramite una delle reti neurali. Questo caso regge la logica del sito, dunque, è sicuramente il più importante e significativo.

Modifica dei contenuti informativi memorizzati del portale:

il caso d'uso considerato è svolto dagli amministratori, i quali, attraverso l'apposita area, possono modificare i contenuti informativi dipendenti dalla realtà applicativa e dunque dalle possibilità momentanee del portale.

Inserimento o cancellazione di un nuovo elemento (rete neurale, categoria, macrocategoria):

il caso d'uso considerato è condotto dagli amministratori, i quali, a seconda delle possibilità offerte al portale, possono inserire o cancellare gli elementi finalizzati a memorizzare in maniera completa e adeguata le classificazioni (rete neurale, categoria, macrocategoria)

Cancellazione di un utente o modifica dei suoi dati:

il caso d'uso considerato identifica la gestione, da parte dell'amministratore, degli utenti registrati; consente di eliminare un utente o modificarne i dati.

2. Progettazione

Per poter realizzare un portale web è necessario preliminarmente predisporre una progettazione. Il capitolo di seguito esposto mira a chiarirne le fasi. In particolare, pertanto, la prima parte del capitolo sarà dedicata alla progettazione del database, mentre la seconda alla progettazione della logica e in parte anche dell'estetica delle pagine HTML. Entrambe le fasi sono infatti fondamentali, in quanto se da una parte fanno chiarezza sui successivi passaggi implementativi, dall'altra definiscono la logica del portale stesso.

2.1. Progettazione del database

Nel presente lavoro di tesi la progettazione del database ha assunto un ruolo centrale; infatti, la sua utilità non si è esaurita con la realizzazione dello stesso, ma ha influenzato la maggior parte delle sezioni di implementazione successive, andando a condizionare scelte e possibilità offerte dal portale agli amministratori ed agli utenti ad esso associati.

Passando in rassegna le fasi di tale tipo di progettazione, come di consueto, preliminarmente sono state previste la raccolta e l'analisi dei requisiti, poi si è passati alla progettazione concettuale e logica ed infine si è proceduto alla normalizzazione del modello realizzato.

2.1.1. Progettazione Concettuale

Schema Concettuale

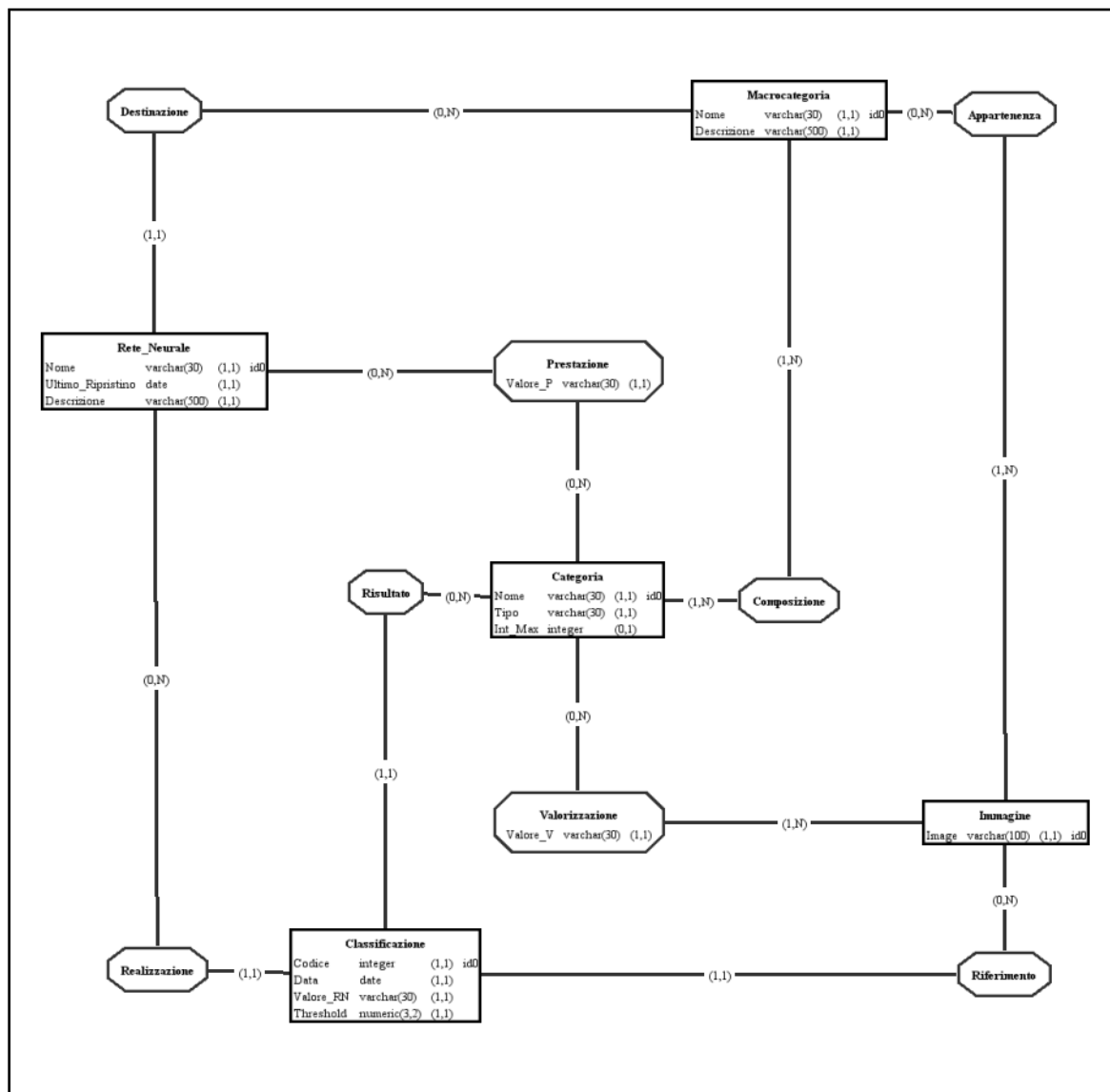


Figura 1. Schema E-R

Note sullo schema E-R

Si è utilizzata una strategia mista realizzando, in una prima fase, lo schema scheletro poi espanso e raffinato ulteriormente. Questa metodologia di progettazione è stata scelta per la grande flessibilità e facilità di applicazione che possiede.

Design Pattern

Pattern Relazione Ternaria *Classificazione*

Si è stato ritenuto opportuno reificare la relazione ternaria *Classificazione* per permettere ad una rete neurale di classificare Immagini diverse per categorie differenziate.

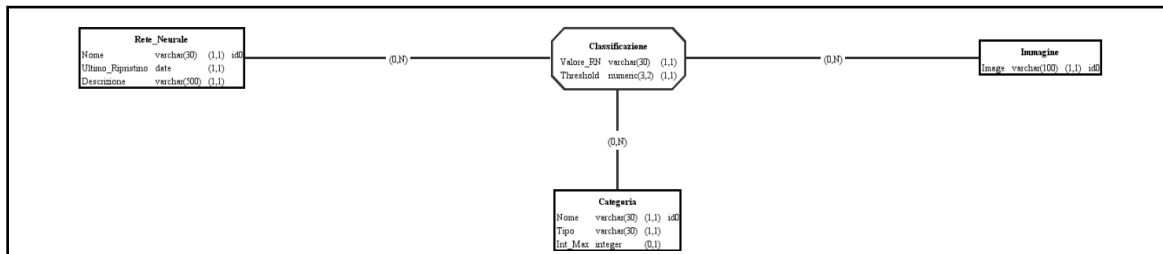


Figura 1. Schema precedente all'applicazione del Pattern Relazione Ternaria *Classificazione*

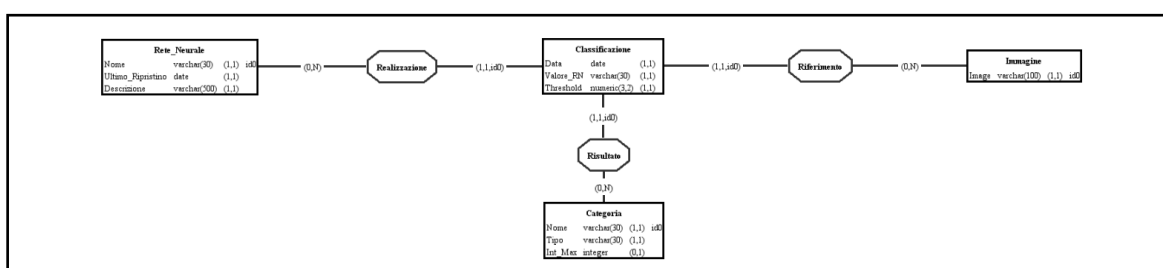


Figura 2 Schema successiva all'applicazione del Pattern Relazione Ternaria *Classificazione*

La sola reificazione, in realtà, ha risolto parzialmente il problema, infatti l'obiettivo finale era permettere ad una rete neurale di classificare per una stessa categoria la stessa immagine più volte. Questo è stato reso possibile solo aggiungendo un codice come identificativo dell'entità *Classificazione*, come mostrato nello schema concettuale.

Dizionario dei Dati

Entità	Descrizione	Attributi	Identificatore
Macrocategoria	Insieme delle categorie che caratterizzano un'immagine	Nome, Descrizione	Nome
Rete_Neurale	Modello computazionale utilizzato per classificare le immagini	Nome, Ultimo_Ripristino, Descrizione	Nome
Immagine	Immagine che l'utente desidera far classificare dalla rete neurale	Image	Image

Classificazione	Risultato dell'analisi della rete neurale	Codice, Data, Valore_RN, Threshold	Codice
Categoria	Caratteristica dell'immagine da classificare	Nome, Tipo, Int_max	Nome

Tabella 1. Dizionario dei dati – Entità

Relazioni	Descrizione	Entità Coinvolte	Attributi
Destinazione	Associa la rete neurale alla macrocategoria a cui è destinata	Macrocategoria, Rete_Neurale	
Composizione	Associa la macrocategoria alle categorie di cui è composta	Macrocategoria, Categoria	
Appartenenza	Associa l'immagine alla macrocategoria a cui appartiene	Macrocategoria, Immagine	
Realizzazione	Associa la classificazione alla rete neurale da cui è stata realizzata	Rete_Neurale, Classificazione	
Riferimento	Associa la classificazione all'immagine a cui fa riferimento	Immagine, Classificazione	
Valorizzazione	Associa la categoria all'immagine valorizzata	Immagine, Categoria	Valore_V
Risultato	Associa alla categoria il risultato della classificazione	Classificazione, Categoria	
Prestazione	Associa alla rete neurale il valore delle proprie prestazioni relativo alla categoria	Rete_Neurale, Categoria	Valore_P

Tabella 2. Dizionario dei dati - Relazioni

Regole Aziendali

Regole di Vincolo
(RV1) Il risultato della classificazione della rete neurale deve essere riferito alle categorie che compongono la macrocategoria per cui la rete neurale è stata scelta
(RV2) La valorizzazione delle categorie per una specifica immagine deve essere riferita alle categorie che compongono la macrocategoria per l'immagine scelta
(RV3) All'inserimento o all'aggiornamento di una rete neurale, l'attributo Valore_P deve essere NULL
(RV4) Dopo la prima classificazione effettuata da una rete neurale, l'attributo Valore_P deve essere maggiore o uguale di 0
(RV5) Una macrocategoria deve essere composta da almeno una categoria
(RV6) Una categoria deve appartenere ad almeno una macrocategoria
(RV7) Un'immagine deve appartenere ad almeno una macrocategoria
(RV8) Un'immagine deve avere almeno una categoria valorizzata
(RV9) Una rete neurale deve essere destinata ad una macrocategoria

Tabella 3. Regole di vincolo

Regole di derivazione
(RD1) L'attributo Valore_P della <i>Prestazione</i> , quando i valori sono interi, a partire dalla prima classificazione è derivabile attraverso la formula:
$\frac{1}{N} \sum_{n=0}^{N-1} Valore_V_n - Valore_RN_n $
(RD2) L'attributo Valore_P della <i>Prestazione</i> , quando i valori sono stringhe, a partire dalla prima classificazione, è derivabile attraverso la formula:
$\frac{1}{N} \sum_{n=0}^{N-1} (Valore_V_n == Valore_RN_n)$
(RD3) L'attributo Valore_P della <i>Prestazione</i> , quando i valori sono interi, a partire dalla seconda classificazione effettuata da quest'ultima e prima di un ripristino, è derivabile anche attraverso il suo valore precedente utilizzando la formula:
$Valore_P = \frac{Valore_P * (N - 1) + Valore_V - Valore_RN }{N}$
(RD4) L'attributo Valore_P della <i>Prestazione</i> , quando i valori sono stringhe, a partire dalla seconda classificazione effettuata da quest'ultima e prima di un ripristino, è derivabile anche attraverso il suo valore precedente utilizzando la formula:
$Valore_P = \frac{Valore_P * (N - 1) + (Valore_V == Valore_RN)}{N}$

Tabella 4. Regole di derivazione

2.1.2. Progettazione Logica

Ristrutturazione Schema Concettuale

Analisi delle Prestazioni

Tavola dei volumi

Concetto	Tipo	Volume
Macrocategoria	E	4
Rete_Neurale	E	4
Immagine	E	400
Classificazione	E	1200
Categoria	E	7
Destinazione	R	4
Composizione	R	7
Appartenenza	R	450
Realizzazione	R	1200
Riferimento	R	1200
Valorizzazione	R	900
Risultato	R	1200
Prestazione	R	8

Tabella 5. Tavola dei volumi

*Nota: Si è ipotizzato che ogni rete neurale abbia effettuato a regime 30 classificazioni.

Si è ipotizzato che a regime una classificazione venga fatta in media su 2 categorie

Tavola delle operazioni

Operazione	Tipo	Frequenza
Operazione 1: Stampa prestazioni (stampa i valori delle prestazioni di una rete neurale)	I	20 al giorno
Operazione 2: Aggiorna prestazione (aggiorna il valore della prestazione di una rete neurale per una categoria in seguito ad una classificazione)	I	40 al giorno

Tabella 6. Tavola delle operazioni

*Nota: L'operazione 1 viene eseguita in seguito ad ogni classificazione più una volta per valutare a fine giornata il miglioramento delle prestazioni di una rete neurale

Analisi delle ridondanze

- Ridondanza: Valore_P (*Prestazione*)

Il valore delle prestazioni di una rete neurale è ottenibile attraverso le formule indicate nelle regole di derivazione

TIPO: Attributo derivabile da altre entità

Analisi della ridondanza: Valore_P

- Operazione 1: Stampa prestazioni**

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIP O
RETE_NEURALE	E	1	L
PRESTAZIONE	R	2	L

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIP O
RETE_NEURALE	E	1	L
PRESTAZIONE	R	2	L
CATEGORIA	E	2	L
RISULTATO	R	300	L
CLASSIFICAZIONE	E	300	L
RIFERIMENTO	R	300	L
IMMAGINE	E	300	L
VALORIZZAZIONE	R	300	L

- Operazione 2: Aggiornamento prestazione**

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIP O
RETE_NEURALE	E	1	L
PRESTAZIONE	R	1	L
CATEGORIA	E	1	L
RISULTATO	R	150	L
VALORIZZAZIONE	R	1	L
IMMAGINE	E	1	L
RIFERIMENTO	R	1	L
CLASSIFICAZIONE	E	1	L
PRESTAZIONE	R	1	S

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIP O
-	-	-	-

Valutazione della ridondanza

Dopo aver analizzato le operazioni che coinvolgono la ridondanza, con il carico considerato, si osserva che:

- In presenza di ridondanza il costo delle operazioni è di circa 6420 accessi giornalieri
- L'occupazione di memoria è di circa 32 byte
- In assenza di ridondanza il costo delle operazioni è di 30100 accessi giornalieri

Per questo motivo si è deciso di mantenere la ridondanza in quanto riduce il numero di accessi in maniera sostanziale e l'occupazione di memoria è trascurabile.

Scelta degli identificatori principali

Rete_Neurale

Per l'entità *Rete_Neurale*, cui si era considerato come identificatore Nome, è stato aggiunto l'identificatore principale Codice in modo da essere facilmente gestibile e non ambiguo.

Immagine

Per l'entità *Immagine*, cui si era considerato come identificatore Image, è stato aggiunto l'identificatore principale Codice in modo da risultare di più semplice gestione.

Schema ristrutturato finale

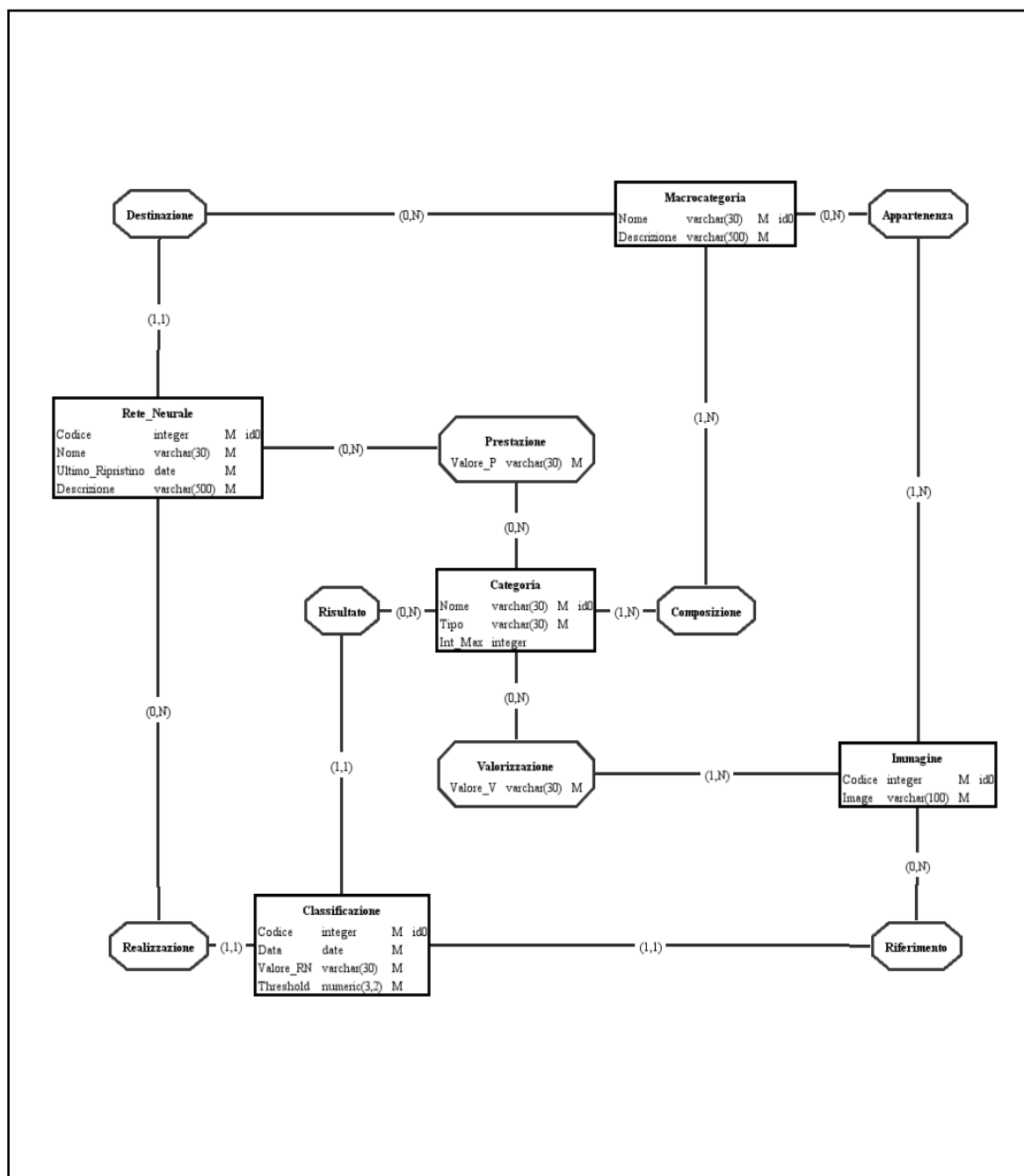


Figura 3. Schema ER Ristrutturato

Schema logico

IMMAGINE (Codice, Image)

APPARTENENZA (Macrocategoria, Immagine)

MACROCATEGORIA (Nome, Descrizione)

RETE_NEURALE (Codice, Nome, Ultimo_Rispristino, Macrocategoria)

CLASSIFICAZIONE (Codice, Data, Rete_Neurale, Immagine, Categoria, Valore_RN, Threshold)

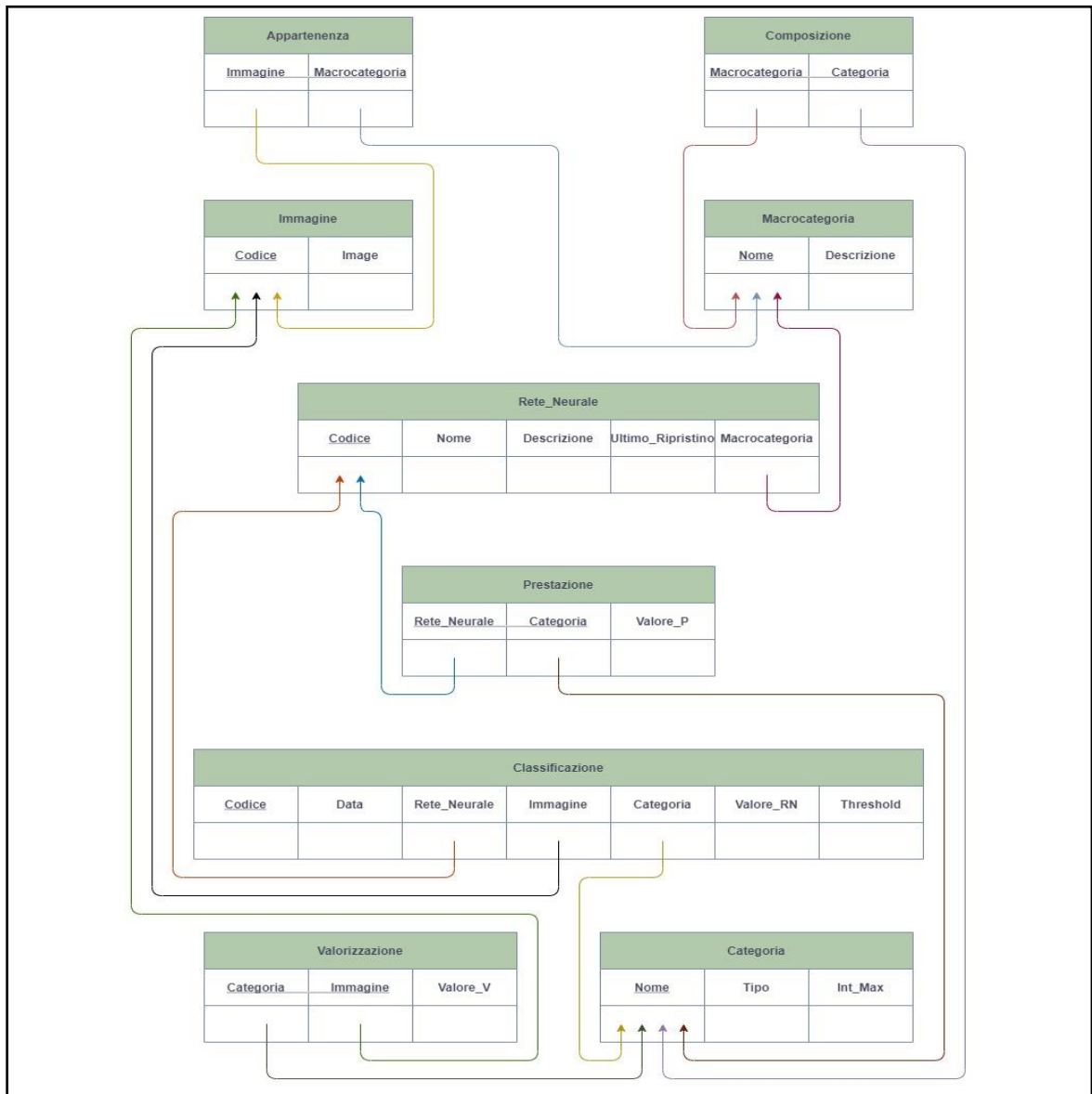
CATEGORIA (Nome, Tipo, Int_max)

VALORIZZAZIONE (Categoria, Immagine, Valore_V)

COMPOSIZIONE (Macrocategoria, Categoria)

PRESTAZIONE (Rete_Neurale, Categoria, Valore_P)

Rappresentazione Grafica



2.1.3. Normalizzazione

Dipendenze Funzionali

Si riportano di seguito, per ciascuna relazione, le dipendenze funzionali non banali:

IMMAGINE (Codice, Image)

Codice → Image

Image → Codice

APPARTENENZA (Macrocategoria, Immagine)

MACROCATEGORIA (Nome, Descrizione)

Nome → Descrizione

RETE_NEURALE (Codice, Nome, Ultimo_Rispristino, Macrocategoria)

Codice → Nome, Ultimo_Rispristino, Macrocategoria

Nome → Codice, Ultimo_Rispristino, Macrocategoria

CLASSIFICAZIONE (Codice, Data, Rete_Neurale, Immagine, Categoria, Valore_RN, Threshold)

Codice → Data, Rete_Neurale, Immagine, Categoria, Valore_RN, Threshold

CATEGORIA (Nome, Tipo, Int_max)

Nome → Tipo, Int_max

VALORIZZAZIONE (Categoria, Immagine, Valore_V)

Categoria, Immagine → Valore_V

COMPOSIZIONE (Macrocategoria, Categoria)

PRESTAZIONE (Rete_Neurale, Categoria, Valore_P)

Rete_Neurale, Categoria → Valore_P

Le relazioni ottenute sicuramente rispettano i prerequisiti della **prima forma normale**, che altro non sono che requisiti del modello relazionale, curati durante la progettazione logica. Invece dall'analisi delle dipendenze funzionali è possibile affermare che le relazioni rispettano i requisiti della **seconda forma normale**, in quanto si trovano in prima forma normale e tutti gli attributi non chiave dipendono dall'intera chiave e non solo da parte di essa. Sempre dalla stessa analisi, si può affermare che tutte le relazioni rispettano i requisiti della **terza forma normale** in quanto si trovano in seconda forma normale e ogni attributo non primo (attributo non appartenente a nessuna chiave candidata della relazione) dipende in modo non transitivo da ogni chiave della relazione. Infine, possiamo anche dire che tutte le relazioni rispettano i requisiti della **forma normale di Boyce-Codd** in quanto, come detto sopra le relazioni rispettano la prima forma normale e inoltre ogni determinante delle dipendenze funzionali individuate è una chiave candidata.

La progettazione sopra descritta ci permette di avere un database normalizzato alla forma di Boyce Codd. Quest'ultimo, oltre ad essere esaustivo nella descrizione dell'intera realtà di interesse, risulta anche essere ottimizzato, non conserva infatti dati ridondanti e risulta essere privo di possibili anomalie. Si è deciso poi, al termine di questa progettazione, di memorizzare effettivamente il database su PostgreSQL (DBMS), in modo da poterlo realizzare nella sua interezza senza particolari limiti nei costrutti utilizzabili.

2.2. Progettazione delle pagine

Preliminarmente alla realizzazione delle singole pagine da inserire nel portale è stata condotta un'analisi relativa sia al loro numero che alla tipologia. Si è deciso così di realizzare quanto segue:

- una Homepage, considerata come pagina iniziale, capace di catturare l'attenzione del nuovo utente
- una pagina Registrazione estremamente semplice ed intuitiva, per consentire l'effettiva registrazione;
- una pagina Test utile per preparare la classificazione della rete neurale selezionata;
- una pagina Result che permettesse all'utente di visualizzare tutto il necessario a valle della classificazione.

Infine, è stato deciso di mantenere per tutte le pagine sia un motivo unico che una struttura di base simile. Ogni pagina, infatti, è stata progettata con un header che consentisse di muoversi nel portale attraverso i link opportuni, da un body, contenuto effettivo di ogni pagina e da un footer.

L'estetica delle pagine è stata abbozzata su Marvel, applicativo che consente di creare prototipi di applicazioni mobili e progetti web, per avere, nella fase iniziale, un'idea chiara del design di ciò che si è successivamente realizzato.

2.2.1. Homepage

La Homepage è la prima pagina visualizzata dall'utente durante il suo primo accesso al portale web; nella fase di progettazione è emersa la necessità di inserirvi tutte le informazioni necessarie alla comprensione delle potenzialità offerte dallo stesso. Questa pagina è stata progettata in lunghezza e divisa in sezioni, ognuna con contenuti diversi, come segue:

1. la prima è totalmente informativa rispetto al portale e consente di comprendere rapidamente i passaggi da seguire per far classificare un'immagine ad una rete neurale, vero obiettivo del portale;
2. la seconda permetta di conoscere, quanto meno come informazione di base, le reti neurali e il loro possibile utilizzo; mostra poi quelle al momento disponibili sul portale, riportandone una breve descrizione associata;
3. la terza, con all'interno un portfolio, permette agli utenti di capire che genere di immagini caricare al variare della macrocategoria scelta; in essa sono anche riportati dati esplicativi riguardanti le possibili macrocategorie selezionabili;
4. la quarta sezione contiene informazioni sui singoli componenti del team che ha partecipato allo sviluppo del portale;
5. la quinta ed ultima sezione, detta "contattaci", permette sia di inviare una mail ad un indirizzo associato al portale stesso, che a identificare la zona presso cui è stato sviluppato, attraverso una geolocalizzazione con Maps.

Se questa è stata la progettazione dei contenuti effettivi della pagina, da un punto di vista logico si è previsto, già in quest'ambiente, di far eseguire all'utente sia l'accesso che la scelta della

macrocategoria per avviare il processo di classificazione. Questo meccanismo ha avuto l'obiettivo di consentire agli utenti già registrati di eseguire direttamente tutti i passi preliminari alla classificazione, senza doversi spostare tra le pagine web del portale. Si è stabilito infine, quando necessario, di offrire la possibilità di registrazione spostandosi semplicemente nella pagina opportuna.

2.2.2. Registrazione

La pagina Registrazione, proprio in virtù del suo scopo, è stata progettata per essere semplice, intuitiva e priva di parti forvianti, riprendendo il motivo presente nella homepage e in tutte le altre pagine. I dati che si è deciso acquisire nei diversi campi sono stati: il nome e il cognome, l'e-mail, l'username e la password con il conferma password per verificare la correttezza della stessa. Per poter ottenere solo registrazioni corrette si è deciso di porre particolare attenzione sul controllo di tali dati: il nome e cognome possono contenere solo lettere, l'e-mail deve rispettare un formato prestabilito, l'username, rappresentando la primary key nella relazione creata per memorizzare gli utenti, deve essere univoco e la password deve contenere almeno una maiuscola e 6 caratteri.

È importante precisare che questa pagina prevede una comunicazione con una specifica relazione non presente nella progettazione del database, poiché da questo dissociata, ma fondamentale per gestire i vari utenti registrati nel portale. Al termine di tutta l'operazione si è previsto che l'utente venga reindirizzato alla homepage, in modalità già loggata, evitando di fargli eseguire un nuovo accesso.

2.2.3. Test

La progettazione della pagina Test è risultata essere centrale nel funzionamento del portale, si è stabilito infatti che questa gestisca sia l'acquisizione dei dati inseriti dall'utente, contornati da tutti i controlli del caso, sia l'indirizzamento verso la pagina Result, transizione nella quale avviene l'effettiva classificazione della rete neurale. La gestione di questi meccanismi necessariamente ha coinvolto il database sottostante principalmente per operazioni di memorizzazione. I dati salvati in quest'ultimo provengono sia dall'utente, attraverso l'immagine caricata e le valorizzazioni delle categorie richieste, sia dalla rete neurale stessa, relativamente alla classificazione da questa effettuata.

Nella progettazione del design si è deciso invece di organizzare la pagina in modo da essere speculare: da un lato vi è la parte relativa all'immagine da caricare, dall'altro vi è una sezione riguardante i dati da inserire o selezionare. Qui, in linea con gli obiettivi prefissati nel lavoro di tesi, è emerso dalla progettazione la grande flessibilità che la pagina deve possedere nella sezione degli input da inserire per le varie categorie. Dopo la compilazione di questo form l'utente, in modalità indiretta, invierà i dati da lui inseriti al database procedendo con il test e, allo stesso tempo, manderà in esecuzione la classificazione ad opera della rete neurale. Al termine si è previsto che l'utente venga reindirizzato alla pagina Result dove visualizzerà i risultati forniti della rete neurale. Infine, come

ulteriore possibilità, si è stabilito nella medesima progettazione che l'utente potrà tornare alla home senza comandare alcuna classificazione.

2.2.4. Result

Result è una pagina integralmente informativa, preleva i dati inseriti dal database dopo la classificazione ad opera della rete neurale e li organizza in tabelle tali da mostrarli con chiarezza all'utente. Durante la progettazione si è deciso di permettere la visualizzazione, anche contemporanea, sia delle risposte della rete neurale, sia dei dati inseriti nella pagina Test, considerati pienamente corretti, che delle informazioni sulla rete precedentemente selezionata.

Il layout, per continuità, è stato considerato simile alla pagina di test con la possibilità di gestire un numero indefinito di dati da far visualizzare all'utente.

Al termine della visualizzazione si è deciso poi di permettere all'utente sia l'inserimento di una nuova immagine per la stessa macrocategoria, sia di tornare alla Homepage.

3. Implementazione

La fase di implementazione è risultata essere una concretizzazione dalle scelte e delle valutazioni prese durante la progettazione del portale web. È importante analizzare come ogni sezione del capitolo precedente ha supportato lo sviluppo di un tassello nel seguente, questo è stato infatti diviso in una parte detta back-end e nel front-end.

In particolare, la precedente progettazione del database ha visto la sua realizzazione in una componente fondamentale del lato “sommerso” del portale (back-end), mentre quella relativa la progettazione delle pagine è stata fondamentale per entrambi i lati della web application.

Entrambi le implementazioni sono state realizzate attraverso diversi applicativi software, largamente diffusi e all'avanguardia.

3.1. Implementazione del back-end

Il back-end rappresenta ciò che è situato alla base di un sito/portale, risulta estremamente importante in quanto ne permette il corretto funzionamento.

È composto da: un server, un'applicazione web e un database. Nel caso del presente lavoro di tesi il database sarà realizzato tramite il DBMS PostgreSQL, l'applicazione tramite il framework Django che permetterà anche di avviare un server per il testing sulla macchina locale (localhost).

3.1.1. Implementazione del database

L'implementazione della base di dati è risultata essere una traduzione, tramite il linguaggio SQL, dello schema logico e dunque delle relazioni che lo compongono, prodotto finale della progettazione della stessa. Per lo schema succitato, sono stati individuati diversi vincoli, inseriti anche nella documentazione, divisi in intrarelazionali e interrelazionali. I primi si dividono a loro volta come: di tupla (relativi alla singola tupla della relazione), di dominio (relativi al dominio di un attributo) e di chiave (not null e unique). Mentre i vincoli interrelazionali sono invece quelli che coinvolgono più relazioni, appartengono a questa categoria i vincoli di integrità.

L'implementazione si suddivide in:

- creazione
- trigger
- popolamento

Creazione

Tale implementazione, si è concentrata nella creazione delle “table”, corrispondenti a tutte le relazioni individuate precedentemente uniti alla traduzione dei vari vincoli ad esse associate. In tale traduzione, questi sono stati gestiti tramite l'individuazione delle chiavi primarie, degli attributi che ammettessero valori nulli e quelli invece obbligatori, e degli eventuali valori di default associati già in fase di creazione. Infine, sono stati tradotti anche i vincoli di dominio ove presenti.

Sullo schema logico in fase di definizione sono stati individuati anche tutti i vincoli di integrità referenziale, per i quali sono state definite delle politiche associate. Determinate politiche si definiscono di reazione, rappresentano la risposta alle violazioni generate da modifiche sulla tabella esterna (tabella referenziata). Sono state selezionate tra le alternative, per operazioni di modifica e cancellazione, quelle che maggiormente risultassero adeguate alla specifica situazione in esame.

Le possibili politiche di reazione sono:

- *cascade*: il nuovo valore dell'attributo della tabella principale viene riportato su tutte le corrispondenti righe della tabella secondaria;
- *set null*: alla chiave esterna della tabella secondaria viene assegnato il valore nullo al posto del valore modificato nella tabella principale;
- *set default*: alla chiave esterna della tabella secondaria viene assegnato il corrispondente valore di default al posto del valore modificato nella tabella principale;
- *no action*: nessuna azione viene intrapresa e viene generato un errore.

Si elencano di seguito per tutte le relazioni che le possiedono le politiche associate ad ogni loro attributo foreign key:

- **Appartenenza:**
 - per l'attributo macrocategoria le modifiche e cancellazioni risultano cascade.
 - per l'attributo immagine le modifiche e cancellazioni risultano cascade.
- **Rete_neurale:**
 - per l'attributo macrocategoria le modifiche risultano cascade, le cancellazioni risultano no action.
- **Classificazione:**
 - per l'attributo rete neurale le modifiche risultano cascade, le cancellazioni risultano no action.
 - per l'attributo categoria le modifiche risultano cascade, le cancellazioni risultano no action.
 - per l'attributo macrocategoria le modifiche risultano cascade, le cancellazioni risultano no action.
- **Valorizzazione:**
 - per l'attributo categoria le modifiche risultano cascade, le cancellazioni risultano no action.
 - per l'attributo immagine le modifiche e cancellazioni risultano cascade.

-
- **Composizione:**
 - per l'attributo categoria le modifiche risultano cascade, le cancellazioni risultano no action.
 - per l'attributo macrocategoria le modifiche e cancellazioni risultano cascade.
 - **Prestazione:**
 - per l'attributo categoria le modifiche risultano cascade, le cancellazioni risultano no action.
 - per l'attributo rete neurale le modifiche e cancellazioni risultano cascade.

Trigger

Per rendere totale il funzionamento del database sono stati implementati i trigger, necessari a gestire alcune delle regole aziendali e di derivazioni individuate, rendendo attiva la base di dati.

tali regole attive, infatti, seguono il paradigma ECA:

- **Evento:** regola di reazione a un determinato evento (ad esempio un'istruzione SQL come INSERT, UPDATE, ecc.). Quando l'evento ha luogo il trigger è chiamato attivato;
- **Condizione:** condizione da valutare, se essa è vera il trigger passa nello stato di valutato;
- **Action:** azione da eseguire (istruzione o procedura in SQL) se la condizione è verificata, successivamente il trigger sarà nello stato di eseguito.

I trigger realizzati sono stati:

- Quelli relativi al soddisfacimento delle regole di vincolo per le cardinalità minime (ins_macrocategoria, ins_categoria e ins_immagine), tutti aventi come evento scatenante l'inserimento su una tabella target differente (rispettivamente macrocategoria, categoria e immagine), in modalità after, con un livello di granularità di tupla (row-level), infine è stata aggiunta la clausola "deferrable initially deferred", per attivare la modalità differita, in modo che la loro valutazione avvenisse dopo il commit work. Qualora il controllo non avesse esito positivo, verrebbe lanciata un'eccezione, con relativo messaggio di errore, che necessariamente causerebbe il rollback di tutte le operazioni eseguite all'interno della transazione. Questi trigger risolvono le seguenti regole di vincolo: ins_macrocategoria RV5, ins_categoria RV6 e ins_immagine RV7 e RV8.
- Quelli relativi al corretto aggiornamento delle prestazioni della rete neurale, per una specifica categoria, a seguito di una nuova classificazione. Sicuramente tale aggiornamento avrebbe potuto essere effettuato tramite una funzione inserita all'interno del programma applicativo. È stato preferito però implementare tale funzionalità tramite un trigger denominato modify_performance. Questo viene attivato dopo un inserimento nella relazione classificazione, con granularità di tupla. All'interno della funzione associata a tale trigger, vengono selezionati, associati alla specifica immagine oggetto della classificazione e per la specifica categoria in esame, sia il valore inserito dall'utente che quello individuato

dalla rete neurale. Vengono prelevati inoltre l'attuale valore della prestazione della rete neurale in questione, e il numero di classificazioni la cui data sia successiva a quella di ultimo ripristino (Saranno considerate solo queste classificazioni nell'aggiornamento delle). Soltanto qualora la data di svolgimento della classificazione fosse successiva alla data di ultimo ripristino della rete neurale si procederebbe, tramite le formule individuate nelle regole di derivazione e tradotte in linguaggio plpgsql, all'aggiornamento del valore corrente delle prestazioni. Questo risolve l'aggiornamento automatico durante il funzionamento della rete neurale, in realtà è stato creato uno ulteriore per gestire il reset delle prestazioni nel caso di aggiornamento della rete neurale stessa. Questo viene rilevato dal cambiamento dell'attributo ultimo ripristino, in questo caso il trigger realizzato è denominato `reset_performance` e viene attivato proprio in seguito ad un'operazione di modifica sull'attributo `ultimo_ripristino` della relazione rete neurale, ha granularità di tupla. La funzione ad esso associata prevede che in questo caso i valori delle prestazioni di quella rete neurale siano resettati, ovvero siano posti a NULL.

Popolamento

tale fase è stata implementata in modo da popolare in maniera rapida e corretta la base di dati. Infatti rispetta i vincoli aziendali individuati e le soluzioni rappresentate dai trigger per la gestione delle cardinalità minime, attraverso una corretta gestione delle transizioni.

Il popolamento è stato implementato in modo tale da prevedere l'inserimento:

- della macrocategoria, delle relative categorie, e delle relative composizioni per associarle. La scelta di inglobare tali operazioni all'interno di una transazione ha permesso il loro inserimento, che altrimenti sarebbe stato negato da uno dei trigger per le cardinalità minime.
- delle varie reti neurali e delle varie prestazioni.
- Si è proceduto infine a testare la correttezza sia dei trigger che della base dati in generale simulando il secondo caso d'uso attraverso l'inserimento di immagini, delle valorizzazioni e delle classificazioni necessarie per la stessa, utilizzando, dei valori casuali. Si è potuto constatare che l'implementazione realizzata riusciva a soddisfare le operazioni principali individuate e che le soluzioni adottate per la gestione di cardinalità minime, aggiornamento delle prestazioni e ripristino di queste ultime fossero effettivamente solide e funzionanti.

Questa correttezza ha permesso di integrare correttamente questa implementazione con le successive.

3.1.2. Implementazione dell'applicazione

Per l'implementazione di dell'app come detto precedentemente è stato utilizzato il framework Django. Tale framework rappresenta un importante supporto allo sviluppo Web, cerca infatti di

velocizzarlo ed ottimizzarlo tramite un approccio totalmente modulare. Le proprietà che possiede un applicativo realizzato tramite questo framework sono: la robustezza, la chiarezza, la scalabilità, la flessibilità e la sicurezza, evitando errori generalmente possibili come SQL injection e gestendo in automaticamente l'aspetto dell'autenticazione.

Django è basato sul paradigma “Model-Template-View”. Nello specifico l'implementazione della presente applicazione si è strutturata proprio in maniera conforme a tale paradigma. L'implementazione è stata dunque suddivisa come segue:

- dei modelli
- dei template
- delle viste

Modelli

La prima fase di Implementazione ha riguardato la creazione dei modelli. Questi nascono dalla programmazione ad oggetti, la quale vuole rappresentare la realtà di interesse proprio attraverso un insieme di “oggetti”, ciascuno caratterizzato da proprietà ed azioni.

Django fa utilizzo di tali modelli, che permettono di salvare dei contenuti all'interno di un database, rappresentano sostanzialmente la fonte di informazione sui dati, poiché contengono i campi e i comportamenti essenziali di ciò che si dovrà memorizzare. La creazione di un modello corrisponde alla creazione di una tabella nel database, dunque, conseguentemente all'implementazione dello stesso, l'operazione che fatta è stata la traduzione dello script di creazione nei vari modelli in questione. Tale scelta è stata animata dal desiderio di uniformarsi quanto più possibile al paradigma su cui si fonda Django, in modo da poter sfruttare l'enorme potenziale di tutte le sue componenti.

Ciascuno di tali modelli è stato definito come sottoclasse di `django.db.models.Model`, e sono stati specificati una serie di attributi corrispondenti a quelli delle diverse relazioni a cui si faceva riferimento, ognuno di questi è stato definito come un'istanza della classe `Field` appropriata al tipo dell'attributo in questione. All'interno di tali istanze sono state valorizzate una serie di opzioni sia quelle caratteristiche per lo specifico campo (come, ad esempio, la `max-length` nel caso dei `CharField`) sia quelli comuni a tutti i campi, relativi alla possibilità di avere valori nulli o meno, chiavi primarie, o valori di default.

Sono stati implementati anche i vincoli di chiave esterna corredandoli con le politiche di reazione relative all'evento di cancellazione e impostando *cascade* di default sulla modifica. L'unica nota di differenziazione rispetto allo script di creazione è stata rappresentata dal fatto che per quelle relazioni che prevedevano delle chiavi primarie formate da due o più attributi si è proceduto all'utilizzo di un campo di chiave primaria automatico, nello specifico ciò è stato previsto per le relazioni “appartenenza”, “valorizzazione”, “Composizione”, “Prestazione”. In aggiunta agli attributi sopra descritti nella definizione dei modelli sono stati forniti anche i “metadati” tramite la classe interna `Meta`, i quali permettono di specificare informazioni aggiuntive del modello, infatti:

- tramite la clausola `db_table`, si è in questo modo chiarito il nome della tabella oltre che eventuali attributi che insieme dovessero soddisfare il vincolo di `unique`, tramite l'espressione `unique_together`.

-
- il metodo `__str()` capace di restituire una rappresentazione di qualsiasi oggetto come una stringa.

Tra i vari modelli sono stati creati anche “Genere”, “Etnia” ed “Emozione”, per soddisfare delle specifiche esigenze implementative, infatti in fase di progettazione è stato deciso che per tutte quelle categorie i cui valori associabili non fossero di tipo numerico, all’utente, in fase di compilazione dei campi, fosse offerta la possibilità di scegliere tra una serie di opzione propostegli che fossero predefinite. La necessità di tale scelta risiede nell’esigenza non solo di essere certi che il valore scelto dall’utente sia valido per il campo di riferimento e per poter essere inserito nella base di dati, ma anche di assicurarsi che sia espresso in maniera tale da poter essere comparato con il risultato fornito dalla rete neurale.

Dunque, affinché le opzioni da mostrare all’utente fossero facilmente modificabili ed espandibili, senza dover apportare delle modifiche ai template, e soprattutto con la finalità di uno snellimento dei codici, si è proceduto alla creazione di un modello per ciascuna categoria non numerica, tale da contenere le alternative possibili da prelevare al momento di visualizzazione del form. Per tali modelli è stato sufficiente prevedere un unico attributo opzione.

Infine, in questa fase implementativa non si è realizzato alcun modello relativo agli utenti registrati poiché già presente in Django.

View e Template

Continuando i passi del paradigma MTV su cui si basa il framework Django, è stato fatto uso delle views per definire la logica associata ad ogni pagina visualizzata. Queste, durante la fase implementativa, hanno rappresentato la connessione tra le informazioni contenute all’interno dei modelli e le visualizzazioni da mettere a disposizione dell’utente all’interno dei template. Le c.d. “viste” sono delle funzioni Python che:

- vengono chiamate da una richiesta Web, attraverso il file `urls.py` che gestisce il reindirizzamento alla vista corretta
- eseguono determinate azioni
- reindirizzano verso una effettiva pagina html presente nei “template” di Django.

Data l’interconnessione forte che le viste hanno con i relativi template, si procederà ad una descrizione parallela, al fine di apprezzare le funzionalità raggiunte. Risulta però fondamentale una prima premessa relativamente ad uno strumento messo a disposizione da Django e fortemente utilizzato per soddisfare i requisiti funzionali del portale web, ovvero i form, i quali accreditano maggiormente la scelta di creazione dei modelli nella fase precedente.

Implementazione della Homepage

La vista Homepage prevede che alla richiesta effettuata da parte dell’utente del path base a cui segue “/Homepage”, venga effettuata il render al template `login.html`. Questa vista sfrutta il meccanismo di autenticazione messo a disposizione da Django, attraverso l’inserimento di una `LoginView` nel

file `urls.py`, evitando di implementare un form che consentisse l'accesso all'utente, all'interno del template `login.html` si potrà richiamare, tramite i comandi Django, un form specificando i campi di `username` e `password`. Tale meccanismo prevede oltre che una corretta visualizzazione del login anche un meccanismo di controllo delle credenziali inserite ed in caso di correttezza all'automatizzazione del meccanismo di autenticazione dell'utente. Nuovamente attraverso i comandi Django, `login.html` estende `homepage.html` attraverso il meccanismo dei "block", dunque l'utente percepirà di essere indirizzato alla Homepage, costruita da una sezione per l'autenticazione dell'utente. Inoltre, sfruttando il campo `extra_content` nella `LoginView`, si è potuto passare alla pagina visualizzata un dizionario aggiuntivo addizionale ai contenuti di default, formato da una serie di campi e di valori, prelevati in maniera dinamica dal database, in modo da rendere le visualizzazioni di diversi valori e sezioni presenti nella homepage costantemente aggiornate ai valori conservati nella base di dati (come richiesto dai requisiti funzionali). In particolare viene prelevato:

- il numero di immagini appartenenti a ciascuna delle macrocategorie in maniera per aggiornare correttamente la sezione riguardante i numeri delle immagini sottoposte al test delle reti neurali.
- i nomi delle reti neurali e le relative descrizioni in maniera tale da popolare la sezione della homepage corrispondente.
- i nomi e le relative descrizioni delle macrocategorie, utilizzate in diversi punti della homepage.

Tali valori, tramite il metodo `append()`, una volta prelevati tramite delle query dal database, sono stati inseriti all'interno tre liste (`home_img`, `home_rn` e `home_macro`), che all'interno dell'Homepage sono state richiamate tramite l'utilizzo di indici. Tramite Django, si è inoltre condizionata visualizzazione della Homepage a seconda del fatto che l'utente si è autenticato o meno (attraverso `user.is_authenticated`), si rimanda alla descrizione della pagina per cogliere tale condizionamento. Infatti, Django utilizza sessioni e middleware per agganciare il sistema di autenticazione agli oggetti di richiesta. Questi forniscono un attributo `request.user` su ogni richiesta che rappresenta l'utente corrente. Se l'utente corrente non ha effettuato l'accesso, questo attributo verrà impostato su un'istanza di `AnonymousUser`, altrimenti sarà un'istanza di `User`.

Implementazione della Registrazione

La vista Registrazione prevede che alla richiesta effettuata da parte dell'utente del path base a cui segue `"/Registrazione"`, venga effettuato il render al template `Registrazione.html`.

Dopo il reindirizzamento l'utente deve poter completare i campi richiesti e successivamente sottoscriverli, in questo caso la necessità di utilizzare un form si è tramutato nella creazione in `forms.py` di una classe `AccountForm` che estendesse la classe `UserCreationForm`. L'estensione applicata è stata necessaria in quanto `UserCreationForm` non prevedeva tutti i campi definiti in fase di progettazione (soltanto `username` e `password`), in particolare sono stati aggiunto i campi:

- nome
- cognome

-
- e-mail

è invece già implementato in tale form il meccanismo di validazione della password, tramite conferma password. Specificati dunque i campi all'interno della vista di registrazione, viene conservata un'istanza di tale form all'interno della variabile "form". successivamente, qualora il metodo con cui viene effettuata la richiesta della pagina sia il metodo POST (Dunque dopo aver compilato i campi), la variabile form sarà popolata con le risposte inserite da parte dell'utente, ottenute tramite *AccountForm(request.POST)*. la validazione dei campi inseriti è realizzata dal metodo *is_valid*, dopo il quale si può procedere al salvataggio dell'utente in Users, relazione automaticamente configurata da Django, nel database. Infine si procede all'autenticazione diretta dell'utente, tramite le funzioni *authenticate()* e *login()*, prima di reindirizzarlo alla homepage. Qualora invece il form non fosse stato sottomesso, dunque la richiesta non fosse stata una POST oppure il form non fosse valido, l'utente visualizza la pagina di registrazione con il form ancora da compilare. Tale valore, in entrambi i casi, viene passato al template di registrazione, in cui si procede alla stampa dei suoi contenuti tramite la variabile *field*, contenente i campi appartenenti al form.

Implementazione del Test-Result

La vista Test gestisce, a seconda di sé la richiesta viene effettuata tramite il metodo GET o il metodo POST due template: *test.html* e *result.html*. Bisogna chiarire che prima della definizione di tale vista, è stato posto il decoratore "@login_required" con l'intento di limitare l'accesso a tali pagine web ai soli utenti autenticati, e a reindirizzare alla homepage eventuali altre richieste.

Tale vista serve sostanzialmente è il vero e proprio core della web application, infatti qui l'utente deve poter valorizzare tutti i campi necessari per visualizzarne poi i risultati.

In fase di progettazione si era scelto che l'utente selezionasse attraverso la homepage la macrocategoria su cui voler effettuare il test. Dunque, è stato in prima battuta necessario gestire che il valore della macrocategoria cliccata da parte dell'utente fosse passato alla vista tramite il metodo GET, inserendo il nome della macrocategoria all'interno dell'URL.

Per di consentire all'utente di caricare l'immagine, effettuare la valorizzazione delle categorie ad essa annesse e selezionare la rete neurale a cui far effettuare la classificazione è stato necessario prevedere l'utilizzo dei form. In fase progettuale infatti, si era deciso che per le diverse tipologie di categorie le tipologie di input all'interno dei form dovessero essere diverse, e poiché la strutturazione dei modelli prevedeva che si dovesse fare riferimento a più modelli contemporaneamente, sono state create tre diverse tipologie di form, tali da gestire le tre diverse tipologie di input previsti per le macrocategorie (la select, il radio e l'input di tipo Number), oltre che un form per l'inserimento dell'immagine e uno per la scelta della rete neurale:

- *ImageForm*: estende la classe *forms.ModelForm* e prevede un input di tipo file, limitato al caricamento delle sole immagini.
- *ClassificazioneForm*: destinato alla scelta delle rete neurale e del valore di threshold da associare. La scelta della rete neurale viene effettuata tramite un controllo di tipo select, all'interno del quale come opzioni vengono mostrate le reti neurali prelevate dal database,

corrispondenti alla macrocategoria selezionata dall'utente. Il valore di threshold viene selezionato tramite un input di tipo range, con valori variabili tra 0 e 1.

- ValorizzazioneNumberForm, ValorizzazioneSelectForm, ValorizzazioneRadioForm: sono i tre form utilizzati per gestire le valorizzazioni delle categorie. Così come è intuibile dal nome, generano un modello di form contenente rispettivamente un input di tipo number, una select o un radio, prelevando dal database, a seconda della categoria passata in fase di creazione del form il valore massimo selezionabile, o le opzioni tra cui effettuare la scelta per gli ultimi due form.

La vista è strutturata per due diverse richieste. Se la richiesta è di tipo GET ed è stato selezionato un valore per la macrocategoria, allora si preleva il nome della macrocategoria selezionata dall'URL, si prelevano dal database tutte le categorie associate a quella macrocategoria e gli attributo tipo corrispondenti, che saranno Select, Number o Radio. Attraverso il codice sottostante, per ciascuna categoria verrà creata un'istanza di form del tipo corrispondente, e tutti quanti saranno contenuti all'interno del set form_set. Verranno inoltre create istanze di form per il caricamento dell'immagine e per le selezioni riguardanti alle reti neurali. Tale dizionario, contenente i vari form, sarà passato al template test.html, dove, tramite un accesso con indici corretti, saranno stampati e rappresenteranno i campi del form che l'utente deve compilare.

```
if (request.method == 'GET') and (request.GET.get('macro') != None):
    macro = request.GET.get('macro').replace('_', ' ')
    comp_set = Composizione.objects.all().filter(macrocategoria=macro)
    form_set = []
    form1 = ImmagineForm()
    for i in comp_set:
        tmp = Categoria.objects.get(nome=i.categoria)
        a = 'Valorizzazione'+tmp.tipo+'Form'
        form_set.append((i.categoria, globals()[a](i.categoria)))
    form3 = ClassificazioneForm(macro)
    return render(request, 'ImageClassifier/Test.html', {'form1': form1, 'macro': macro, 'form_set': form_set, 'form3': form3})
```

Qualora il metodo fosse invece il metodo POST, significherebbe che il form è stato compilato e inviato. Dunque, verrebbero prelevate le istanze dell'immagine caricata, della rete neurale e del threshold scelti e tramite un meccanismo simile al codice sopra riportato verrebbero recuperati anche i valori espressi dall'utente in relazione alle diverse categorie.

Si riporta di seguito tale codice dove valore_v corrisponde al valore/i inserita dall'utente:

```
form_set = []
form1 = ImmagineForm(request.POST, request.FILES)
form1_valid = form1.is_valid()

form_scroll_valid = True
for count, i in enumerate(comp_set):
    tmp = Categoria.objects.get(nome=i.categoria)
    a = 'Valorizzazione'+tmp.tipo+'Form'
    tmp_form = globals()[a](i.categoria, {'valore_v': str(
        request.POST.getlist('valore_v')[count])})
    if tmp_form.is_valid() == False:
        form_scroll_valid = False
    else:
        form_set.append(
            (i.categoria, str(request.POST.getlist('valore_v')[count])))
form_value = []
form3 = ClassificazioneForm(macro, request.POST)
form3_valid = form3.is_valid()
img = Immagine.objects.all().filter(image=form1.instance)
```

Al termine di tale procedura, si procede al salvataggio nella medesima transazione:

- Nel caso non sia già presente nel database dell'immagine caricata dall'utente
- Nel caso non sia già presente nel database dell'istanza di Appartenza che lega l'immagine caricata alla macrocategoria in questione
- In base alla rete neurale selezionata del valore da questa fornito per ciascuna categoria, in classificazione
- Nel caso non siano già presenti dei valori inseriti dall'utente per ciascuna categoria associati all'immagine caricata, in valorizzazione

In seguito al salvataggio, tali informazioni, con in aggiunta il numero di classificazioni effettuate dalla rete neurale, vengono passati come dizionario al template Result, dove saranno mostrati all'utente in modo tale da rispettare le specifiche iniziali.

3.2. Implementazione del Front-end

L'implementazione del front-end ha previsto l'utilizzo del framework Bootstrap, il quale fornisce dei modelli di progettazione basati su pagine HTML corredati da CSS e JavaScript messi a disposizione da quest'ultimo. La finalità, quindi, è stata di realizzare delle pagine web che fossero responsive, dunque adattabili a diverse visualizzazioni. Il contributo offerto ha sostanzialmente riguardato l'utilizzo di alcuni template grafici a cui ispirarsi e una serie di componenti con funzionalità proprie riusabili, da poter essere integrati nella struttura progettata.

Per la realizzazione di interfacce che fossero effettivamente conformi agli obiettivi prefissati è stata talvolta necessaria l'integrazione di funzioni JavaScript, finalizzate alla messa appunto di alcuni controlli, quali per esempio la validazione dell'input in fase di registrazione.

Caratteristiche comuni

Ciascuna delle pagine facenti parte del portale, è stata strutturata in modo da prevedere un header e un footer, di colore nero, e un corpo centrale. Fatta eccezione la homepage, le altre pagine presentano un header strutturato in modo da contenere il logo del portale e una breadcrumb, ovvero dei link di navigazione che ripercorrono il percorso disegnato dal sito e consentono all'utente di capire dove si trova e di poter tornare alla homepage.

All'interno del footer presentano invece informazioni quali dei contatti per reperire i membri del team sviluppatore del portale. La homepage si discosta dalle altre pagine in quanto prevede in aggiunta una barra di navigazione per consultare agevolmente le varie sezioni della pagina, e dai pulsanti posti nella breadcrumb, per la gestione dei meccanismi di login e di logout condizionati da se l'utente è autenticato o meno.

È importante precisare per ciascuna pagina sia che l'header è stato realizzato in modo tale da essere sticky, semplificando la navigazione nel sito, sia che tramite una serie di ridimensionamenti dei diversi componenti, specificati all'interno dei file CSS collegati e realizzati tramite l'applicazione delle media queries, ci si è assicurati che fossero capaci di adattarsi a qualsiasi tipologia di visualizzazione: computer, tablet, cellulari di varie dimensioni.

Nella definizione grafica delle varie pagine, dal punto di vista dei colori, delle forme e delle risposte all'interazione dell'utente si è soddisfatto l'obiettivo proposto in fase progettuale, ovvero quello di realizzare un'omogeneità dei diversi componenti al fine di rendere la navigazione dell'utente più intuitiva.

3.2.1. Implementazione pagina di registrazione:

Il body della pagina di registrazione prevede un'immagine di sfondo, leggermente opacizzata, realizzata in modo occupare l'intera visualizzazione. È stato realizzato tramite l'utilizzo di una struttura organizzata in colonne, in modo tale che quella centrale contenesse la sezione formata dai vari campi da compilare.

La colonna centrale possiede un contenitore con un bordo arrotondato di colore giallo, con uno sfondo parzialmente trasparente di colore nero, questo è organizzato in modo da prevedere una sezione titolo, una sezione formata dai vari campi di input e una sezione contenente i tasti "Registrati" e "Reset".

Per la validazione dell'input sono state realizzate delle funzioni javascript, attivate su eventi target differenti, che implementassero i controlli progettati.

- checkmail: la quale controlla che l'e-mail rispetti i requisiti richiesti, ovvero che contenga sia la '@' che il nome di dominio, nell'ordine corretto. Tale funzione viene attivata nel momento in cui si toglie il focus dalla casella corrispondente
- controlla: tale funzione controlla che non sia presente la '@' all'interno dei campi nome e cognome, viene attivata al momento in cui si toglie il focus dal campo corrispondente e in caso di fallimento del controllo prevede che venga generato un>alert per notificare l'evento all'utente.
- nonnumber: tale funzione controlla che non siano presenti numeri o caratteri speciali all'interno dei campi nome e cognome, viene attivata al momento di digitazione dei caratteri e in caso di fallimento del controllo prevede che venga generato un alert per notificare l'evento all'utente.
- checkpass: tale funzione prevede il controllo per la definizione della robustezza della password inserita dall'utente, viene attivata al momento di digitazione dei caratteri e prevede l'incremento di percentuale e il cambiamento del colore della progressbar sottostante.
- pass: tale funzione controlla che la password inserita soddisfi i prerequisiti posti, viene attivata nel momento in cui si toglie il focus dalla casella corrispondente.
- formcheck_reg: tale funzione permette l'invio del form soltanto qualora tutti i campi siano stati compilati e, tramite le verifiche effettuate dalle altre funzioni, che l'utente abbia rispettato i prerequisiti.

3.2.2. Implementazione pagina di homepage

La homepage è stata strutturata e organizzata in diverse sezioni.

- Home: è la sezione che viene visualizzata al momento di caricamento della pagina. Prevede una riga, ripartita in più colonne, all'interno delle quali sono contenute delle icon-box, ovvero sostanzialmente dei box cliccabili tramite i quali l'utente autenticato può effettuare la scelta della macrocategoria su cui effettuare il test.
- Info: è la sezione all'interno della quale l'utente può reperire informazioni sul portale. È stata organizzata tramite l'impiego della struttura offerta dal container, all'interno del quale, per separare le immagini dai componenti testuali è stata usata una ripartizione tramite la componente di layout grid messa a disposizione da Bootstrap.
- Reti Neurali: è la sezione all'interno della quale l'utente può consultare i nomi e le descrizioni delle reti neurali disponibili. Anche queste, contenute all'interno di un container, sono state organizzate in questa disposizione, ripartendo una riga su più colonne. Ciascuno di tali box è un icon-box.
- Portfolio: contiene immagini di esempio fornite all'utente come guida, sulle quali è possibile effettuare una selezione. Ciascuna di tali immagini è organizzata all'interno di colonne di dimensioni differenti. Oltre questo vi è una parte contenente dei numeri associati alle macrocategorie uniti ad una piccola descrizione realizzate sulla base di componenti Bootstrap (Carousel ecc.).
- Team: all'interno di tale sezione si possono conoscere i membri del team, ovvero i loro nomi e eventuali contatti utili. Le diverse card sono organizzate su più colonne, allineate all'interno della medesima riga.
- Contatti: è la sezione contenente i contatti utili per l'utente e la sede del team. La visualizzazione della mappa è resa possibile tramite l'utilizzo di un contenuto di tipo frame. Vi è in questa sezione la possibilità di inviare un'e-mail in automatico all'indirizzo di posta specificato, cliccando sul bottone.

3.2.3. Implementazione della pagina per il test e il risultato

Tali due visualizzazioni hanno una struttura estremamente somigliante, anche nella sezione centrale. Orizzontalmente sono state strutturate su righe successive. La prima sezione prevede il nome della macrocategoria scelta, la sezione subito sottostante prevede gli effettivi contenuti per la compilazione dei campi o la visualizzazione dei risultati, quest'ultima prevede nel caso del test i due tasti per caricare l'immagine e per sottomettere i campi inseriti, mentre nel caso di visualizzazione dei risultati è previsto il tasto per il caricamento di una nuova immagine per la classificazione, sulla medesima macrocategoria. L'area centrale risulta essere formata da una card (Bootstrap), organizzata in modo tale che la sezione di sinistra preveda la visualizzazione dell'immagine da caricare o già caricata, mentre la sezione di destra preveda delle strutture per l'inserimento dei valori o la visualizzazione dei risultati.

Per entrambe le visualizzazioni la sezione di destra prevede dei box cliccabili, al cui evento di click vengono rese visibili oppure nascoste le sezioni associate. Nella pagina di test vi sono due di questi box, uno destinato all'inserimento dei valori per le categorie corrispondenti e uno destinato all'inserimento dei valori riguardanti la rete neurale. Le sezioni associate sono state realizzate tramite il componente collapse messo a disposizione da Bootstrap.

Nella pagina in cui vengono visualizzati i risultati vi sono invece tre dei box precedentemente descritti, a cui sono associate delle sezioni dotate del medesimo meccanismo, una associata alla visualizzazione delle risposte dell'utente, una della rete neurale e una per la visualizzazione di informazioni inerenti alla rete neurale, quali l'indice di performance medio per ciascuna categoria, il threshold selezionato in fase di caricamento e il numero di classificazioni fino a quel momento effettuate. Tutte e tre queste sezioni sono al loro interno organizzate in maniera tabellare, prevedendo un header della tabella che renda auto-esplicativa la consultazione dei dati.

Le sezioni presenti in entrambe le visualizzazioni sono al loro interno organizzate in maniera tabellare, tali da contenere ciascun controllo all'interno di una riga. In caso di aumento dei campi è stata predisposta una scroll-bar che ne consenta la visualizzazione, evitando modifiche nell'assetto della grafica.

Alla pagina in cui l'utente esegue il caricamento, sono state associate due diverse funzionalità gestite tramite delle funzioni Javascript.

- readURL: Tale funzione consente all'utente di poter visualizzare, nella sezione apposita, l'immagine che ha appena caricato, prima si sottomettere i campi, utilizzando i meccanismi offerti da javascript di navigazione nel DOM.

-
- loading: Tale funzione consente di intrattenere l'utente quando, dopo la sottomissione del form e prima della visualizzazione dei risultati, bisogna attendere che la rete neurale effettui la propria elaborazione. Si riesce in questo modo, tramite i messaggi "Aspetta ancora un po..." e "La rete neurale sta caricando...", a comunicare all'utente di attendere non facendo apparire una situazione che sarebbe assimilabile ad un blocco del sistema.

4. Descrizione delle pagine realizzate

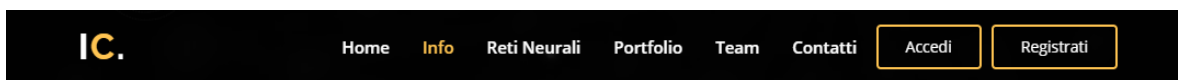
4.1. Homepage

La Homepage è composta da: un Header, un Body diviso in sezioni e un Footer.

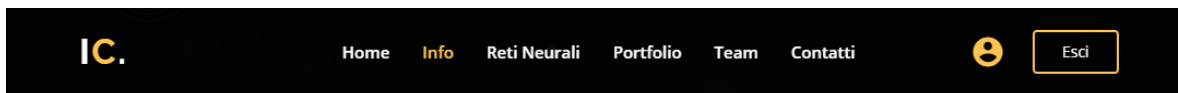
Header

L'Header, nel caso della Homepage, è composto dal logo, da una navbar e da una parte condizionata dal fatto che l'utente sia o meno autenticato; in particolare abbiamo:

- il logo, simbolo del portale, che una volta cliccato ricarica la pagina
- Il menu, che porta a tutte le sezioni definite nelle fasi precedenti (Home, Info, Reti Neurali, Portfolio, Team, Contatti)
- La parte finale, che:
 - nel caso di un utente non autenticato è composta sia dal pulsante Accedi che conduce alla parte presente nel Footer adibita a contenere i campi per il Login, sia dal pulsante Registrati che conduce alla pagina di registrazione.



- nel caso di un utente autenticato, invece, è composta da un logo per l'utente e dal pulsante Esci finalizzato ad eseguire il logout.



Body

Home



La prima sezione dell'Homepage è stata realizzata in maniera tale da avere 4 tasti centrali corrispondenti alle 4 macrocategorie selezionabili dall'utente che ha effettuato l'accesso. Una volta Premuto uno di essi l'utente verrà reindirizzato alla pagina dei test, avviando in tal modo i passaggi per la classificazione da parte delle reti neurali; a seconda della macrocategoria selezionata, la pagina mostrerà opzioni e input differenti.

È importante precisare che un utente non loggato visualizzerà la Home allo stesso modo di un utente loggato, ma i 4 tasti non saranno selezionabili.

Info

INFO

SCOPR(I.C.)I

Un portale web per la classificazione di immagini tramite reti neurali

Puoi caricare un numero illimitato di immagini per testare le performance delle reti neurali disponibili

- ✔ Le icone visualizzate nella Home corrispondono alle macrocategorie disponibili
- ✔ Ad ognuna di esse sono associate delle categorie e delle reti neurali da testare
- ✔ Leggi le istruzioni di seguito riportate per sapere cosa fare



Cosa aspetti?



Autenticazione

Registrati o Accedi per iniziare



Caricamento di un'immagine

Clicca su una delle macrocategorie sopra disponibili, carica l'immagine che preferisci e specifica i campi richiesti



Selezione della rete neurale

Scegli, tra quelle disponibili, la rete neurale con cui effettuare la classificazione



Visualizzazione del risultato



Visualizza il risultato della rete neurale selezionata e il suo indice di performance medio

Questa sezione riporta le informazioni utili all'utente per effettuare il test delle reti neurali. Le informazioni poste a sinistra rispetto all'immagine sono di carattere generale, mentre quelle in basso nella parte destra rappresentano una "To do list" per permettere la classificazione. Ogni immagine scelta mira da essere auto-esplicativa. Il logo all'interno delle parentesi del titolo della sezione è cliccabile e riporta alla Home.

Reti

RETI NEURALI

SCOPRI LE NOSTRE RETI NEURALI

 Face Bio Effettua l'analisi dei volti in una foto	 Count Vehicles Effettua il conteggio dei veicoli in una foto	 Count People Effettua il conteggio delle persone in una foto	 Count Faces Effettua il conteggio dei volti in una foto
--	---	---	--



Che cos'è una rete neurale?

Una rete neurale è un modello matematico composto da neuroni artificiali di ispirazione alle reti neurali biologiche (quella umana o animale) e viene utilizzata per risolvere problemi ingegneristici di Intelligenza Artificiale legati a diversi ambiti tecnologici come l'informatica, l'elettronica o altre discipline. Le reti neurali fanno sì che i computer siano in grado di risolvere i problemi in modo indipendente e che migliorino le loro capacità. Esse prevedono la necessità di un addestramento iniziale effettuato da persone, in base al metodo di intelligenza artificiale utilizzato.

[Scopri Di Più](#)

La sezione contiene una griglia relativa alle reti neurali disponibili. Ogni riquadro riporta il nome identificativo ed una breve descrizione della rete, dati presi dinamicamente dal database sottostante. Passando con il mouse su un riquadro lo si mette in evidenza. Nella parte inferiore è presente una definizione di rete neurale accompagnata da un pulsante che, se cliccato, riporta alla pagina di Wikipedia.

Portfolio

PORTFOLIO

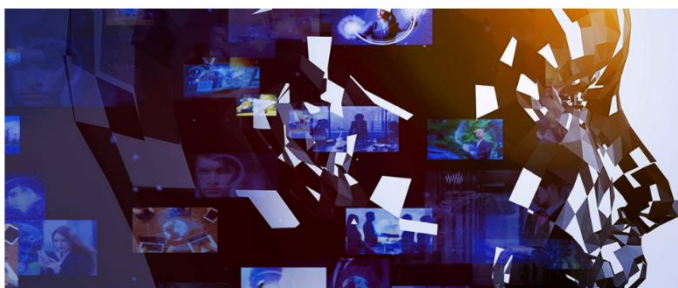
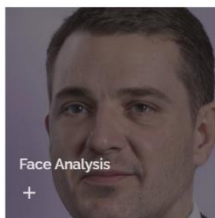
SCOPRI IL NOSTRO PORTFOLIO

TUTTE

VOLTI

PERSONE

VEICOLI



Un po' di numeri...

...grazie al contributo dei nostri utenti le reti neurali sono state testate, per ogni macrocategoria, sul seguente numero di immagini:

5

Face Counting

2

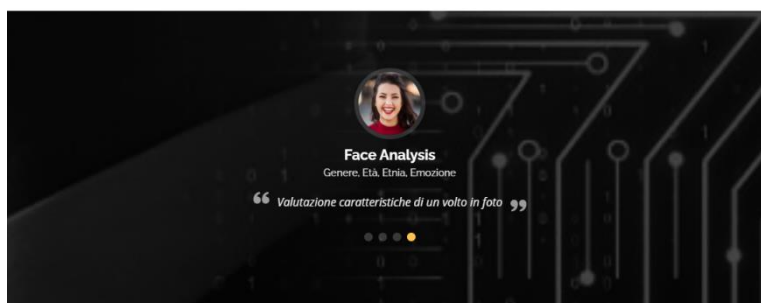
People Counting

0

Vehicle Counting

5

Face Analysis



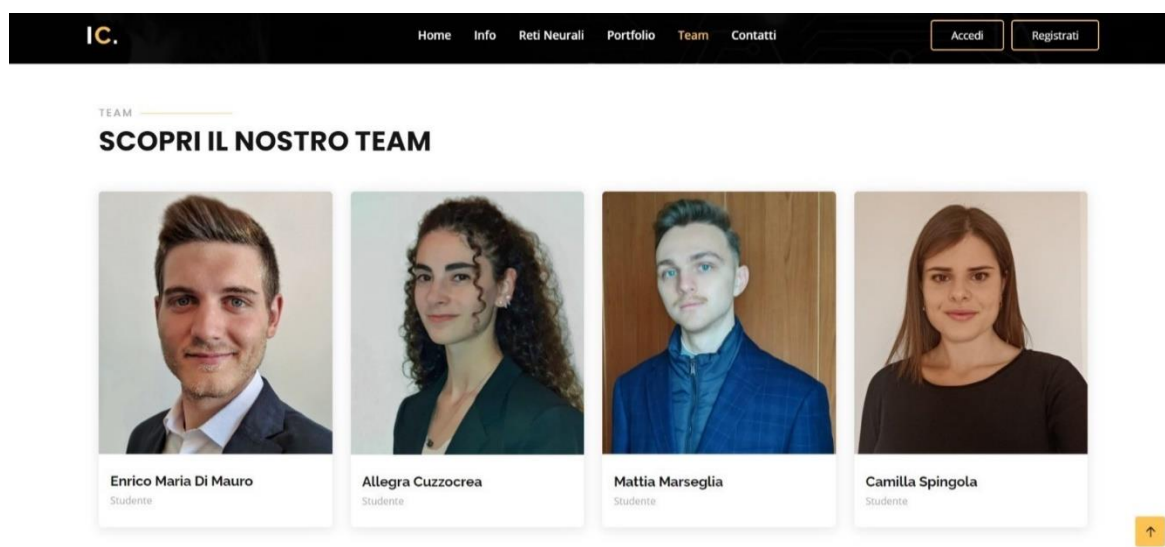
La sezione del portfolio riporta alcune immagini di esempio divise in base a ciò che le reti neurali dovranno riconoscere: facce, persone e veicoli. È possibile visualizzare le immagini per gruppi cliccando gli appositi pulsanti.

Passando con il mouse su un'immagine essa si evidenzia ed appare in basso a sinistra sia un'icona che permette di visualizzarla zoomata, che la macrocategoria per cui risulta adatta maggiormente.

Sotto le immagini è presente un riquadro, aggiornato dinamicamente, in cui viene riportato il numero di immagini per ogni macrocategoria su cui gli utenti hanno testato le reti neurali.

Nell'ultima parte in basso, infine, è presente un riquadro a scorrimento in cui vengono presentate le macrocategorie con un'immagine caratterizzante, le categorie che la compongono e una breve descrizione dinamica.

Team



La sezione Team riporta la foto, il nome e la professione di ogni componente del gruppo che ha realizzato il portale.

Passando con il mouse su uno dei riquadri appare il link che invia alla pagina social del componente in questione.

Contatti

CONTATTI

CONTATTACI



Luogo:

Via Giovanni Paolo II, 132, 84084
Fisciano SA



Email:

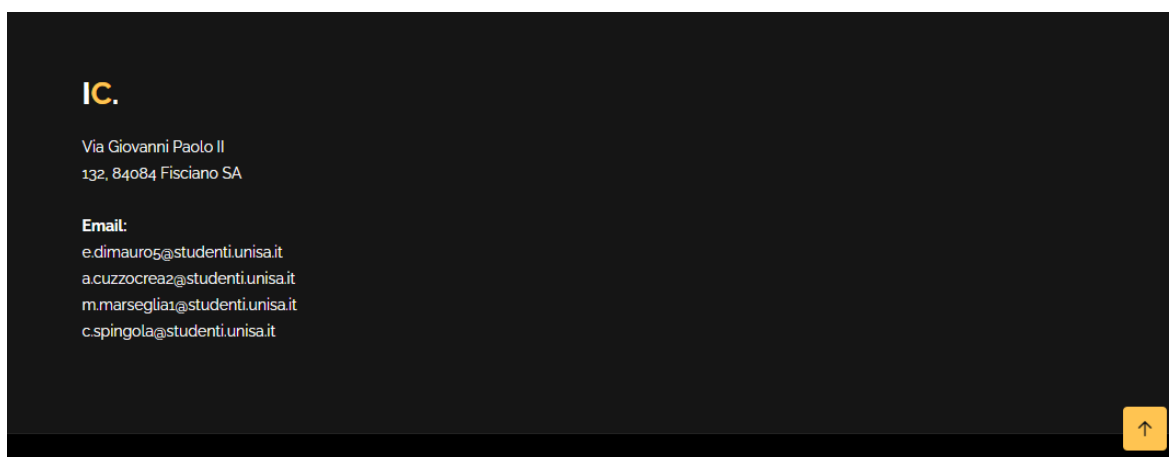
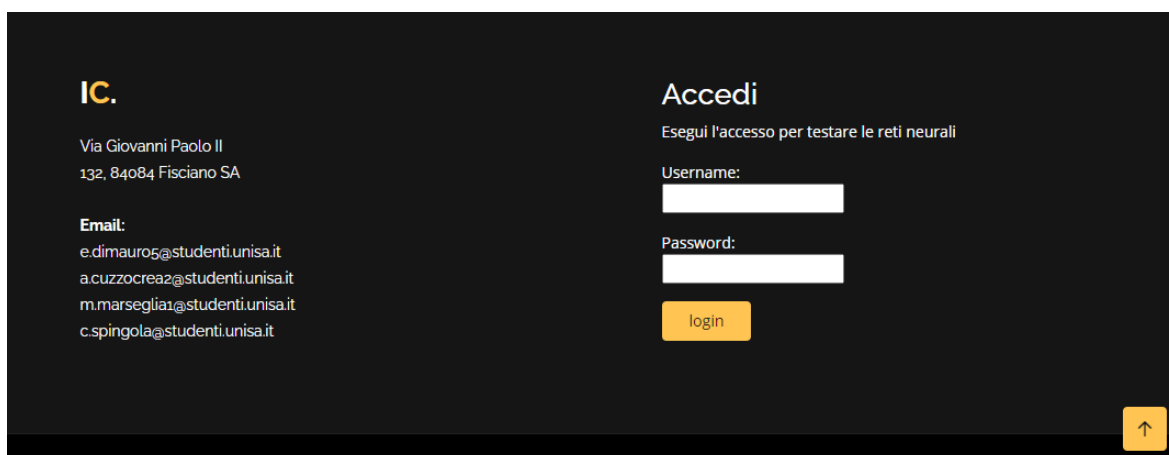
image.classifier@telegmail.com

Contattaci

La sezione Contatti riporta la geolocalizzazione dell'Università degli Studi di Salerno, l'indirizzo della stessa e la mail di riferimento per contattare il gruppo.

Premendo sul pulsante Contattaci è possibile inviare una mail all'indirizzo specificato sopra.

Footer



Il Footer presente nella homepage riporta a sinistra il logo accompagnato dall'indirizzo dell'università per la quale si è svolto il seguente lavoro di tesi e le mail dei membri del gruppo; a destra è riportata la sezione adibita al Login, visualizzabile, come mostrato nelle immagini, solo dagli utenti non autenticati.

Cliccando sul logo l'utente viene riportato alla Home.

Cliccando su Accedi, dopo aver inserito Username e Password corretti, l'utente viene riportato alla Homepage come utente autenticato, altrimenti verrà visualizzato un messaggio di errore prima di provare nuovamente l'autenticazione per l'accesso.

4.2. Registrazione

IC. Home / Registrazione

REGISTRAZIONE

Nome

Cognome

E-mail

Username

•

La Password deve contenere :
• almeno 6 caratteri
• almeno una maiuscola

Weak

Conferma Password

IC. Home / Registrazione

REGISTRAZIONE

Nome

Cognome

E-mail

Username

•

La Password deve contenere :
• almeno 6 caratteri
• almeno una maiuscola

Good

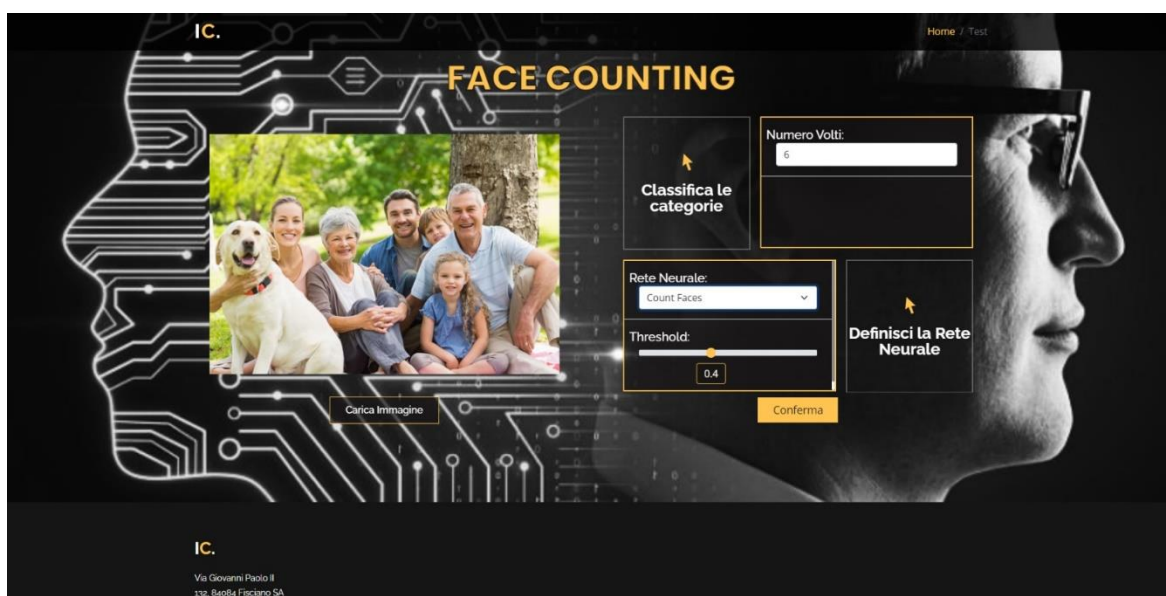
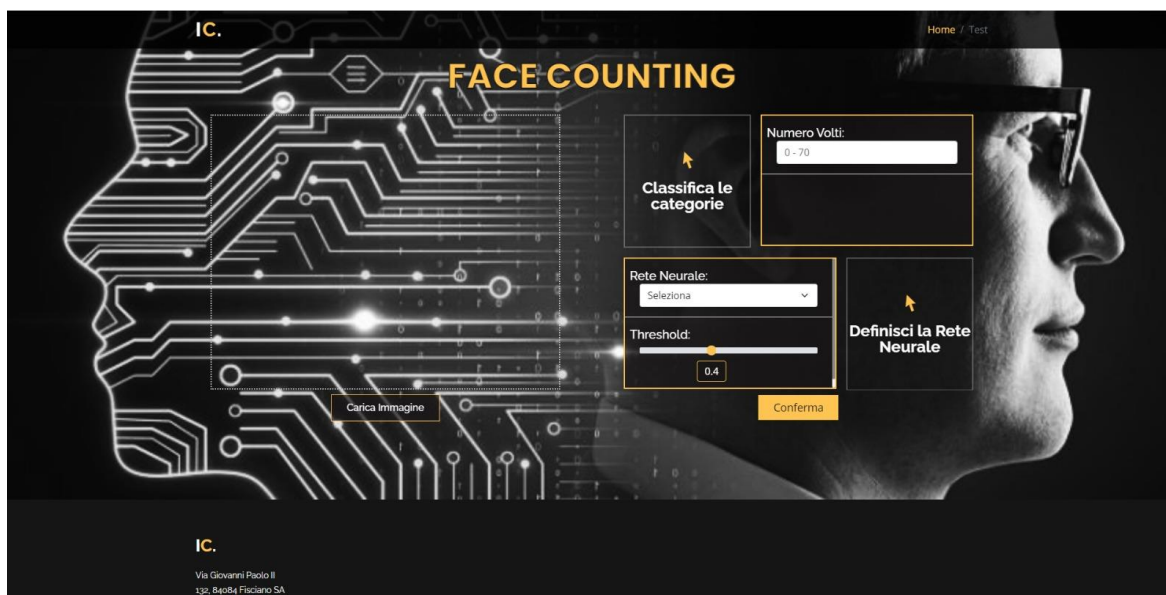
Conferma Password

La pagina Registrazione riporta il form contenente i campi richiesti per potersi registrare al portale. Cliccando sul pulsante Registrati senza aver compilato tutti i campi verrà notificato un messaggio di errore tramite un popover; esso richiederà di compilare il primo campo vuoto partendo dall'alto. In caso di corretto inserimento di tutti i dati si verrà reindirizzati alla pagina principale come già autenticati.

Cliccando sul pulsante reset vengono svuotati tutti i campi.

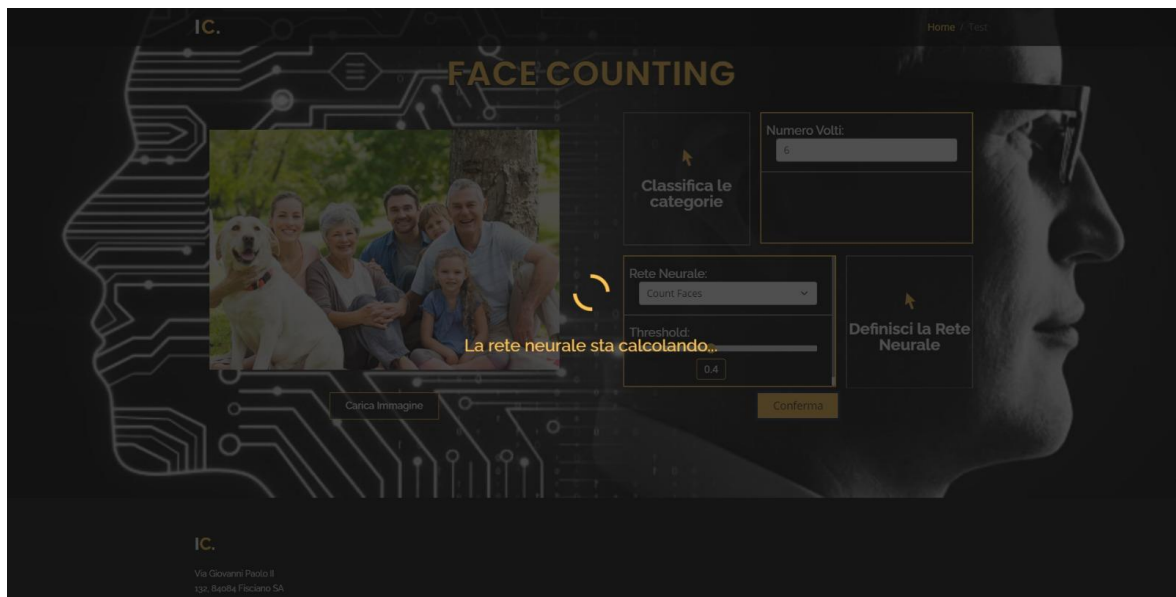
Sono presenti tutti i controlli del caso definiti nella fase di progettazione e realizzati nell'implementazione.

4.3. Test



La pagina Test riporta ciò che l'utente loggato vede dopo aver cliccato su una delle macrocategorie disponibili nella Home. In particolare, l'utente potrà caricare un'immagine, attraverso l'apposito pulsante, valorizzare le relative categorie e scegliere la rete neurale desiderata per effettuare il test. La visualizzazione delle scelte di ogni categoria sarà vincolata dalla macrocategoria selezionata apriori.

Cliccando sul pulsante Conferma i dati verranno memorizzati nel database con le relative classificazioni della rete neurale e l'utente verrà reindirizzato a Result visualizzando l'ultima la seguente schermata durante il caricamento:



Per rendere più interattiva e leggera la pagina sono stati inseriti dei pulsanti cliccabili al loro fianco che aprono e chiudono i contenitori.

Se non si inseriscono tutti i campi non si può proseguire verso la pagina Result, così come da progettazione, con implementazione di tutti i controlli del caso.

4.4. Result



Nella pagina Result l'utente visualizza: l'immagine caricata, il confronto tra le sue risposte e quelle date dalla rete neurale e le informazioni associate alla rete neurale scelta. In particolare, sono visibili: l'indice di prestazione aggiornato associato alla rete neurale scelta, il threshold selezionato precedentemente e il numero di classificazioni effettuate da quella rete neurale.

Cliccando sul pulsante "Carica nuova immagine" si potrà caricare una nuova immagine e visualizzare la pagina nella versione riportata in precedenza.

Cliccando il logo o la breadcrumb in corrispondenza della home nell'header l'utente tornerà alla pagina iniziale.

In continuità con la pagina Test anche quest'ultima, al fine di alleggerire il contenuto della pagina, possiede i pulsanti per chiudere le informazioni al loro fianco.

Conclusione

Il punto di forza della ricerca sopra riportata va assegnato sicuramente alla sua estrema adattabilità e contemporaneità. I risultati del lavoro di tesi sono stati estremamente soddisfacenti, conducendomi sicuramente verso una crescita personale nell'acquisizione di competenze nell'ambito discusso, ma anche da un punto di vista umano. L'attenzione prestata alla cura dei dettagli e all'approfondimento ha sicuramente contribuito infatti a far crescere in me autostima e capacità applicativa, oltre che consapevolezza della bontà della scelta intrapresa tre anni fa.

A questo punto non mi rimane che auspicare che questo sia solo l'inizio di un percorso entusiasmante che mi porti sempre più a conoscere quel mondo virtuale che cammina sempre più parallelamente alla vita di tutti i giorni.

Bibliografia e sitografia

Paolo Atzeni, Stefano Ceri, Piero Fraternali, Stefano Paraboschi, Riccardo Torlone “Basi di Dati”, V Edizione.

Web Languages. URL:

<https://www.w3schools.com/>.

Bootstrap documentation. URL:

<https://getbootstrap.com/docs/4.1/getting-started/introduction/>.

PgAdmin 4 documentation. URL:

<https://www.pgadmin.org/docs/pgadmin4/4.25/index.html>.

Django documentation. URL:

<https://docs.djangoproject.com/en/3.2/>.

Python documentation. URL:

<https://www.tutorialspoint.com/python/index.htm>.

Ringraziamenti

In primis, ho raccolto con piacere lo stimolo offerto dai miei relatori Antonio Greco e Alessia Saggese, ai quali va un ringraziamento speciale per la loro immensa pazienza, gli indispensabili consigli e le conoscenze trasmesse durante tutto il percorso di stesura dell'elaborato e dell'attività di tirocinio, condivisa con i miei stimabili colleghi Camilla, Allegra ed Enrico.

Ringrazio mio padre e mia madre che mi hanno sempre sostenuto, accompagnandomi durante la crescita, e sono riconoscente nei loro confronti dell'infinito amore che mi hanno sempre dimostrato.

Ringrazio mio fratello Simone, che mi ha sopportato e supportato nel condividere gli spazi che sono stati testimoni dell'impegno continuo e costante profuso negli studi intrapresi.

Ringrazio Camilla, collega e compagna di vita di questi anni, per avermi trasmesso la sua immensa forza e determinazione e, soprattutto, per tutta la costruzione di un lavoro condiviso ed entusiasmante che abbiamo portato avanti insieme durante l'intero percorso di studi. Grazie perché hai contribuito a rendermi la persona che sono oggi.

Ringrazio mio cugino Giuseppe, sempre presente nelle mie scelte e nei momenti importanti della mia vita e per la stima che manifesta continuamente nei miei confronti, che contraccambio.

Ringrazio i nonni e gli zii, che sono stati per me una guida ed un esempio di vita, familiare e professionale, per l'affetto che mi hanno sempre mostrato.

Infine, dedico questa tesi a me stesso, ai miei sacrifici e alla mia tenacia, che mi hanno permesso di arrivare fin qui.

Ringrazio tutti i colleghi ed in particolar modo Ferdinando e Vito, con cui ho condiviso i momenti salienti di questo percorso in un periodo così difficile.