

UNIVERSITÀ DEGLI STUDI DI SALERNO



**Dipartimento di Ingegneria dell'Informazione ed
Elettrica e Matematica applicata**

Corso di Laurea in Ingegneria Informatica

ELABORATO DI TESI

INTEGRAZIONE DI UNA RETE NEURALE PER L'ANALISI AUTOMATICA DEI VOLTI IN UNA WEB APPLICATION

Primo Relatore

Prof. Antonio Greco

Secondo Relatore

Prof.ssa Alessia Saggese

Azienda Ospitante

Università degli Studi di Salerno

Candidato

Enrico Maria Di Mauro

Matricola

0612704486

Anno Accademico 2020/2021

Sommario

INTRODUZIONE	1
1. CAPITOLO 1	2
TECNOLOGIE UTILIZZATE	2
1.1. HTML	2
1.2. CSS.....	2
1.3. JAVASCRIPT	3
1.4. BOOTSTRAP	3
1.5. PYTHON.....	3
1.6. DJANGO.....	4
1.7. POSTGRESQL	4
2. CAPITOLO 2	5
PROGETTAZIONE.....	5
2.1. PROGETTAZIONE DEL DATABASE	5
2.1.1. PROGETTAZIONE CONCETTUALE	5
2.1.2. PROGETTAZIONE LOGICA	12
2.1.3. NORMALIZZAZIONE	22
2.1.4. TRIGGER.....	24
2.2. PROGETTAZIONE DEL PORTALE WEB	24
2.2.1. HOMEPAGE.....	25
2.2.2. TEST	26
2.2.3. RESULT.....	27
2.2.4. REGISTRAZIONE	27
3. CAPITOLO 3	28
IMPLEMENTAZIONE	28
3.1. HOMEPAGE.....	30
3.2. TEST	37
3.3. RESULT.....	40
3.4. REGISTRAZIONE	41
4. CAPITOLO 4	43
CONCLUSIONI E PROSPETTIVE FUTURE	43
BIBLIOGRAFIA E SITOGRAFIA	44
RINGRAZIAMENTI	45

INTRODUZIONE

L'attività di tesi riguarda la progettazione e lo sviluppo di un portale web che permette ad un utente loggato, gestito da un amministratore, di scegliere una macrocategoria e, successivamente, di caricare un'immagine, valorizzare le relative categorie, selezionare una tra le reti neurali disponibili ed impostarne la sensibilità selezionando un valore compreso tra 0 ed 1 (più è basso più è sensibile). In seguito, viene eseguita dalla rete neurale la classificazione della medesima immagine.

Un utente non loggato può visualizzare solamente la pagina iniziale e quella di registrazione.

Le macrocategorie messe a disposizione sono:

- Face Counting: permette il conteggio dei volti all'interno dell'immagine
- People Counting: permette il conteggio delle persone all'interno dell'immagine
- Vehicle Counting: permette il conteggio dei veicoli all'interno dell'immagine
- Face Analysis: permette l'analisi di un volto all'interno dell'immagine

Ciascuna macrocategoria possiede delle categorie predefinite che devono essere necessariamente valorizzate per proseguire il test. In particolare, le prime tre posseggono unicamente la categoria Numero che, come dice il nome, permette di indicare il numero di facce, persone o veicoli presenti all'interno dell'immagine. L'ultima, invece, possiede le categorie Età, Genere, Etnia ed Emozione.

Ogni rete neurale è destinata a classificare immagini appartenenti ad una specifica macrocategoria.

Le reti neurali messe a disposizione sono: Count Faces, Count People, Count Vehicle e Face Bio.

Ciascun utente può classificare un numero illimitato di immagini. Per ogni test vengono conservate e mostrate all'utente l'immagine caricata, le categorie da lui valorizzate ed il risultato della classificazione della rete neurale. Inoltre, per la rete neurale utilizzata, il portale calcola le performance medie ottenute attraverso una media mobile esponenziale su tutte le immagini da essa classificate.

La progettazione e l'implementazione del portale web è stata svolta in gruppo perseguendo l'obiettivo di realizzare un codice quanto più modulare possibile. In questo modo, è stato realizzato un portale web facilmente espandibile e che non si limiti alla realtà di interesse. Il mio contributo all'interno del Team di sviluppo è stato a 360 gradi partecipando alla progettazione ed all'implementazione dell'intera attività. La mia attenzione è stata posta particolarmente sulla macrocategoria Face Analysis che permette di sfruttare la rete neurale Face Bio per analizzare un volto e determinarne l'età, il genere, l'etnia e l'emozione.

Nel Capitolo 1 saranno presentate le tecnologie utilizzate durante lo sviluppo del progetto.

Nel Capitolo 2 sarà descritta nel dettaglio la progettazione sia del database utilizzato per la gestione dei dati della realtà di interesse che del portale web.

Nel Capitolo 3 saranno esposte le scelte implementative effettuate per la realizzazione del portale web, fornendo una descrizione degli elementi utilizzati ed immagini rappresentative.

Nel Capitolo 4, infine, saranno fornite le conclusioni.

1. CAPITOLO 1

TECNOLOGIE UTILIZZATE

Le tecnologie utilizzate per la realizzazione del progetto sono: i linguaggi HTML, CSS, JAVASCRIPT e PYTHON, i framework Bootstrap e Django ed il DBMS PostgreSQL.

1.1. HTML

HTML (HyperText Markup Language) è il linguaggio utilizzato per descrivere le pagine web, o documenti ipermediali, che costituiscono i nodi dell'ipertesto situati sui server web. Tale linguaggio viene renderizzato ed interpretato dai browser lato client. Tramite HTML è possibile realizzare documenti con una struttura semplice contenenti testo, immagini e contenuti multimediali, oggetti interattivi e, tramite i link, connessioni ipertestuali ad altri documenti.

I documenti html sono documenti di testo scritti in codice ASCII con estensione .html che possono essere editati da un qualunque text-editor.

Le caratteristiche principali del linguaggio HTML:

- è un linguaggio di marcatura del testo e non è un linguaggio di programmazione.
- è costituito da un insieme di parole chiave dette tag, o markup, con cui identifica porzioni del documento.
- è nato con l'obiettivo di consentire la definizione di documenti ipertestuali distribuiti su una rete di computer.

Con HTML è possibile:

- definire la struttura logica di un documento.
- definire la formattazione tipografica del testo fornendo indicazioni utili per il rendering.
- definire dei collegamenti ipertestuali tra i vari documenti.
- definire l'interfaccia utente di un'applicazione WEB

1.2. CSS

CSS (Cascading Style Sheets) è il linguaggio utilizzato per lo stile delle pagine web. Questi fogli di stile specificano come un browser deve posizionare, formattare e visualizzare i vari elementi che compongono una pagina web. Ogni foglio di stile è una collezione di regole stilistiche che definiscono il look & feel, ossia descrivono le caratteristiche percepite dall'utente di un'interfaccia grafica, sia in termini di apparenza visiva (look) che di modalità di interazione (feel). Uno stile può trovarsi sia all'interno di un file .html sia in un file di testo con estensione .css. Il termine Cascading si riferisce all'ordine di applicazione delle definizioni di stile. Le regole applicate in file esterni hanno priorità minore rispetto a quelle presenti all'interno dei tag delle pagine html. Ogni foglio di stile possiede un insieme di proprietà che caratterizzano la presentazione degli elementi HTML, o meglio suggeriscono come mostrare il contenuto al browser. La definizione dello stile consiste nell'assegnamento di valori a un insieme di proprietà. Utilizzando i fogli di stile è possibile ottenere una netta separazione tra contenuto ed aspetto, ciò porta ai seguenti vantaggi:

- chiarezza semantica: si capisce cosa deve definire il file .html e cosa il file .css.
- chiarezza stilistica: tutte le regole sono riportate in un unico file.

Ogni foglio di stile consiste in una serie di regole chiamate statements che permettono di fare 2 cose:

- identificare gli elementi tramite un selettore
- suggerire al browser come renderizzare gli elementi

L'ordine a cascata per determinare quale stile deve essere applicato ad un determinato elemento, in ordine di importanza, è il seguente:

- Inline Style: all'interno di elementi HTML con l'attributo globale "style".
- Internal Style Sheet: nel tag STYLE all'interno del tag HEAD.
- External Style Sheet: in un file esterno .css.
- User Styles: in un file definito dall'utente all'interno del browser
- Browser Styles: stile di default

1.3. JAVASCRIPT

JavaScript (originariamente chiamato LiveScript) è un linguaggio di scripting, orientato agli oggetti e agli eventi, per client e server web. I programmi scritti in Javascript sono interpretati attraverso lo Javascript Engine. Il Rendering Engine prende il contenuto della pagina HTML e le informazioni di stile CSS e mostra il contenuto formattato dopo averlo fatto interagire con lo Javascript Engine. Uno script può trovarsi sia all'interno di un file .html sia in un file di testo con estensione .js.

Javascript, come tutti i linguaggi di script, è un linguaggio di programmazione con una sintassi semplice, che permette di fare "cose" utili con "pochi" comandi ad-hoc. I linguaggi di script combinano lo scripting con HTML per ottenere pagine interattive. Javascript, ad esempio, viene utilizzato per:

- Validare i dati dei form durante la compilazione.
- Visualizzare messaggi che possono richiedere dati in apposite finestre, chiamate alert box.
- Animare il contenuto della pagina interagendo con l'utente
- Adattare le pagine ai browser
- E molto altro

1.4. BOOTSTRAP

Bootstrap rappresenta un framework per applicazioni web che offre una raccolta di strumenti liberi. La sua documentazione contiene modelli di progettazione basati su HTML, CSS e Javascript utili all'inserimento di componenti innovativi all'interno delle pagine web per realizzare in modo semplice ed efficace interfacce avvincenti. Le sezioni contenute nella documentazione sono: Introduction, Customize, Layout, Content, Forms, Components, Helpers, Utilities, Extend e About.

1.5. PYTHON

Python è un linguaggio di programmazione ad alto livello che supporta diversi paradigmi di programmazione, come quello object-oriented (con supporto all'ereditarietà multipla), quello imperativo e quello funzionale, ed offre una forte tipizzazione dinamica. Tale linguaggio è uno dei più ricchi e comodi da utilizzare grazie alla sua libreria built-in estremamente ampia, alla gestione automatica della memoria ed a robusti costrutti per la gestione delle eccezioni. La sua sintassi è pulita e snella così come i suoi costrutti, chiari e non ambigui. I blocchi logici vengono costruiti semplicemente allineando le righe allo stesso modo, incrementando la leggibilità e l'uniformità del

codice anche se vi lavorano diversi autori. Python è un linguaggio pseudo-compilato: un interprete si occupa di analizzare il codice sorgente (semplici file testuali con estensione .py) e, se sintatticamente corretto, di eseguirlo. In Python, non esiste una fase di compilazione separata (come avviene in C, per esempio) che generi un file eseguibile partendo dal sorgente. Queste caratteristiche rendono Python un linguaggio portabile. È possibile, infatti, eseguire il codice su gran parte delle piattaforme attualmente utilizzate a patto di avere la corretta versione dell'interprete.

1.6. DJANGO

Django rappresenta un framework per applicazioni web gratuito e open source, scritto in Python. Garantisce la creazione semplice e veloce di applicazioni web basate su database. Django fornisce un'area di amministrazione dei contenuti che può gestire molti siti con un'unica installazione. L'applicazione per l'amministrazione permette di creare, aggiornare ed eliminare contenuti, rappresentati da oggetti, tenendo traccia di tutte le operazioni effettuate e fornisce un'interfaccia per la gestione di utenti e gruppi di utenti (inclusa la gestione dei permessi). Django segue il paradigma MVC (Model-View-Controller). Il Model ha il compito di astrarre la rappresentazione dei dati, la View quello di renderizzare i dati ed il Controller quello di gestire gli eventi che processano le interazioni con l'utente. Tale approccio prevede una separazione degli elementi MVC in tre differenti file denominati rispettivamente models.py, view.py ed urls.py. Altre caratteristiche di Django sono:

- riusabilità dei componenti.
- possibilità di installare funzionalità attraverso plugin
- sistema di template basato su tag ed ereditarietà
- sistema per la creazione e la validazione di form HTML

1.7. POSTGRESQL

PostgreSQL è un DBMS Object-Relational (ORDBMS), spesso abbreviato con il nominativo "Postgres", che all'atto dell'installazione si pone come un server sempre attivo.

Per gestire il DBMS si utilizza PgAdmin, ossia un client grafico.

PostgreSQL è uno dei DBMS più conformi allo standard SQL. Quest'ultimo è il linguaggio standard dei DBMS e si divide in 3 sezioni, ognuna si occupa di determinate funzioni:

- DDL (Data Definition Language) si occupa della definizione dei dati
- DML (Data Manipulation Language) si occupa della manipolazione dei dati
- DCL (Data Control Language) si occupa del controllo dei dati

Il principale punto di forza di Postgres è la sua programmabilità ed inoltre esso permette di realizzare applicazioni utilizzando i dati prelevati dal database che sono conservati come una serie di tabelle con chiavi esterne.

PostgreSQL è in grado di risolvere problemi direttamente nella base dati e permette agli utenti di definire nuovi tipi basati sui normali tipi di dato SQL. Permette, inoltre, l'ereditarietà dei tipi, che costituisce una caratteristica principale della programmazione orientata ad oggetti.

2.CAPITOLO 2

PROGETTAZIONE

2.1. PROGETTAZIONE DEL DATABASE

La progettazione del database utilizzato per la gestione del portale web è avvenuta in quattro fasi.

2.1.1. PROGETTAZIONE CONCETTUALE

Lo scopo della progettazione concettuale è quello di rappresentare le specifiche informali della realtà di interesse in termini di una descrizione formale e completa. Il prodotto di questa fase viene chiamato schema concettuale e fa riferimento ad un modello concettuale dei dati.

I modelli concettuali ci consentono di descrivere l'organizzazione dei dati ad un alto livello di astrazione, senza tenere conto degli aspetti implementativi.

Il modello utilizzato in questo progetto è il modello E-R, o Entity-Relationship, che fornisce una serie di strutture grafiche, dette costrutti, atte a descrivere la realtà di interesse in una maniera facile da comprendere e indipendente dalle tecniche usate dagli elaboratori per memorizzare dati.

I costrutti vengono utilizzati per costruire schemi che descrivono i dati di interesse, i principali sono entità, associazioni ed attributi. Altri costrutti del modello sono: cardinalità delle associazioni e degli attributi, identificatori e generalizzazioni.

Le entità rappresentano classi di oggetti (fatti, cose, persone, ecc.) che hanno proprietà comuni ed esistenza "autonoma" ai fini dell'applicazione di interesse. Un'occorrenza di un'entità è un oggetto della classe che l'entità rappresenta pertanto non è qualcosa che individua l'oggetto, ma l'oggetto stesso. In uno schema, ogni entità ha un nome che la identifica univocamente.

Le associazioni rappresentano un legame logico tra due (o più) entità. In uno schema E-R, ogni associazione ha un nome che la identifica univocamente. Un'associazione mette in relazione un'occorrenza di un'entità con una di un'altra entità. Un aspetto molto importante delle associazioni è il seguente: tra le occorrenze di una associazione del modello E-R non ci possono essere ennuple ripetute, ciò vuol dire che un'istanza di un'entità non può far riferimento più volte ad un'istanza di un'altra entità. In tal caso l'associazione deve diventare un'altra entità (reificazione).

Gli attributi descrivono le proprietà elementari di un'entità o associazione. Ad ogni attributo è associato un dominio, che contiene i valori ammissibili. Nel caso in cui possano esistere diversi valori di un certo attributo per un'occorrenza di una certa entità si parla di attributi multivalore. Gli attributi di una medesima entità o associazione che presentano un legame nel loro significato o uso possono essere raggruppati aumentando in questo modo l'espressività, in questo caso si parla di attributi composti.

La cardinalità delle associazioni descrive il numero minimo e massimo di occorrenze di associazione a cui un'occorrenza di entità può partecipare. Osservando le cardinalità massime è possibile classificare le associazioni binarie in base al tipo di corrispondenza che viene stabilita tra le occorrenze delle entità coinvolte:

- Le associazioni aventi cardinalità massima pari a 1 per entrambe le entità coinvolte sono denominate associazioni uno ad uno.
- Le associazioni aventi un'entità con cardinalità massima pari a 1 e l'altra con cardinalità massima pari a N sono denominate associazioni uno a molti.
- Le associazioni aventi cardinalità massima pari a N per entrambe le entità coinvolte vengono denominate associazioni molti a molti.

La cardinalità degli attributi descrive il numero minimo e massimo di valori dell'attributo associati ad ogni occorrenza di entità o relazione. I casi possono essere:

- (1, 1): ad ogni occorrenza è assegnato obbligatoriamente 1 valore dell'attributo
- (1, N): ad ogni occorrenza è assegnato 1 o più valori dell'attributo
- (0, 1): ad ogni occorrenza è assegnato al massimo 1 valore dell'attributo
- (0, N): ad ogni occorrenza è assegnato da 0 a N valori dell'attributo

Quando si ha N come cardinalità massima, stiamo parlando di un attributo multivalore.

Gli identificatori sono specificati per ciascuna entità dello schema e descrivono gli attributi atti ad identificare univocamente le occorrenze di tali entità. Essi possono essere interni o esterni all'entità e possono coinvolgere uno o più attributi.

Le generalizzazioni rappresentano legami logici tra un'entità, detta entità genitore, ed una o più entità, dette entità figlie. Questo concetto non è applicabile alle associazioni.

Tra le entità coinvolte in una generalizzazione valgono le seguenti proprietà generali:

- Una occorrenza di una entità figlia è anche una occorrenza dell'entità padre.
- Ereditarietà: ogni proprietà dell'entità padre è anche una proprietà dell'entità figlia.

Le generalizzazioni possono essere classificate sulla base di due proprietà tra loro ortogonali:

- Una generalizzazione è totale se ogni occorrenza dell'entità genitore è una occorrenza di almeno una delle entità figlie, altrimenti è parziale
- Una generalizzazione è esclusiva se ogni occorrenza dell'entità genitore può essere al massimo un'occorrenza di una delle entità figlie, ma mai a più di una, altrimenti è sovrapposta.

2.1.1.1. PRODUZIONE DELLO SCHEMA CONCETTUALE

Lo sviluppo di uno schema concettuale a partire dalle sue specifiche può essere considerato a tutti gli effetti un processo di ingegnerizzazione e, come tale, risultano ad esso applicabili le strategie di progetto.

In seguito ad un'attenta analisi delle caratteristiche da implementare per la realtà di interesse, è stata utilizzata una strategia mista per la realizzazione di tale schema.

La strategia mista è una delle più utilizzate nell'ambito della realizzazione di schemi concettuali, in quanto è in grado di fornire estrema flessibilità grazie alla possibilità di suddividere un problema complesso in sotto-problemi più semplici e quella di procedere per raffinamenti successivi.

I passi da seguire quando si utilizza tale strategia sono 5:

- Suddividere le specifiche iniziali in componenti via via sempre più piccole fino ad ottenere descrizioni di frammenti elementari della realtà di interesse.
- Realizzare, tramite i concetti elementari, uno schema scheletro.
- Decomporre lo schema scheletro in modo da avere aree separate sulle quali concentrarsi, anche separatamente dividendole all'interno di un gruppo.
- Raffinare, espandere ed integrare ogni area fino ad ottenere per quest'ultima la struttura completa.
- Ricomporre, seguendo lo schema scheletro, ogni struttura ottenuta per formare lo schema concettuale finale.

Prima di procedere con la progettazione concettuale, però, è stata presa in considerazione la possibilità futura di possedere ulteriori macrocategorie, categorie e reti neurali, per tale motivo è stata scelta una strada che potesse portare alla modularità.

Per la realizzazione dello schema scheletro sono state individuate 4 entità elementari, ossia: Macrocategoria, Categoria, Rete Neurale ed Immagine collegate tra loro mediante opportune

associazioni. Le prime tre sono identificate rispettivamente da Nome, l'ultima, invece, è identificata da Image, ossia l'URL.

Le associazioni sono state nominate in modo da garantire una comprensione quanto più immediata possibile a chi legge lo schema per la prima volta.

Per collegare l'insieme di categorie contenute all'interno di una macrocategoria è stata utilizzata l'associazione Composizione. Inoltre, le categorie sono collegate all'immagine attraverso l'associazione Classificazione.

Per creare una corrispondenza tra l'immagine caricata dall'utente e la macrocategoria scelta da quest'ultimo è stata utilizzata l'associazione Appartenenza.

Per collegare l'immagine e le categorie che appartengono alla stessa macrocategoria è stata utilizzata l'associazione Valorizzazione, che possiede un attributo chiamato Valore_V che indica per ogni categoria il valore immesso dall'utente per quell'immagine. È stato supposto che l'utente inserisca sempre la risposta corretta, per tale motivo la prima valorizzazione per la stessa categoria associata alla medesima immagine sarà l'unica ad essere salvata all'interno del database.

Una rete neurale può far riferimento ad una ed una sola delle macrocategorie a disposizione, per indicare ciò è collegata ad esse attraverso l'associazione Destinazione. Inoltre, la rete neurale è collegata alle categorie attraverso Prestazione, che possiede un attributo chiamato Valore_P che indica per ogni categoria il valore calcolato dalla rete neurale effettuando una media mobile esponenziale sugli esiti di tutte le classificazioni eseguite, ed all'immagine attraverso Classificazione, che possiede l'attributo chiamato Valore_RN che indica per ogni categoria il valore immesso dalla rete neurale per quell'immagine e Threshold che consente di tenere traccia della sensibilità utilizzata dalla rete neurale per effettuare tale classificazione.

Abbiamo ottenuto, così, una visione astratta di ciò che deve essere il nostro schema concettuale.

Dopo una attenta analisi atta a valutare se fossero necessarie modifiche per raffinare ed espandere lo schema scheletro appena ottenuto, è stata rilevata una relazione ternaria tra le entità Immagine, Categoria e Rete Neurale attraverso l'associazione Classificazione e sono stati inseriti nuovi attributi utili a descrivere la realtà di interesse.

Una relazione ternaria è considerata tale se tutte le cardinalità dei vari collegamenti alle entità sono (0, N) o (1, N).

Per poter permettere più classificazioni di una stessa categoria per la medesima immagine utilizzando la stessa rete neurale, è stato utilizzato uno dei design pattern relativi alla progettazione concettuale dei dati, ossia la Reificazione di Relazione Ternaria.

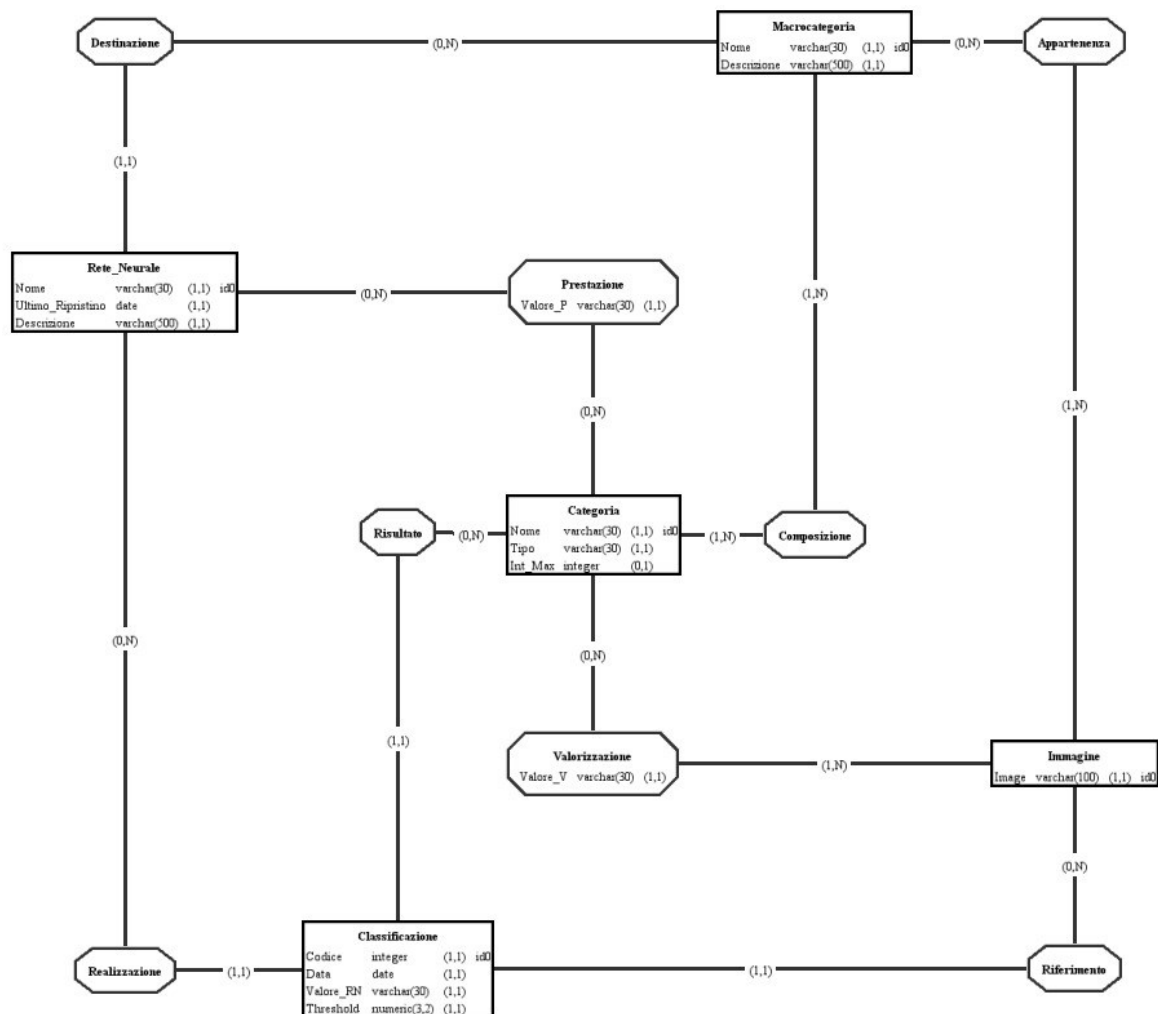
I design pattern sono soluzioni a problemi comuni dell'ingegneria.

Per effettuare la reificazione dell'associazione Classificazione è bastato trasformare tale associazione in un'entità col medesimo nome e gli stessi attributi, e collegarla ad Immagine, Categoria e Rete Neurale, che svolgeranno contemporaneamente il ruolo di identificatori esterni, mediante 3 nuove associazioni: Riferimento, Risultato e Realizzazione. Le nuove associazioni posseggono necessariamente una cardinalità pari ad (1, 1) verso Classificazione e quelle originarie verso le altre entità. Il ruolo svolto dall'identificatore esterno, però, non consente ugualmente di sottostare alle precedenti specifiche, per tale motivo è stata deciso di sostituirlo con un identificativo interno individuato dall'attributo Codice. Oltre a queste operazioni è stato aggiunto un nuovo attributo chiamato Data che si occupa di tracciare temporalmente la classificazione.

Nelle entità Macrocategoria e Rete Neurale è stato aggiunto l'attributo Descrizione che, come dice il nome, consente di possedere una descrizione della macrocategoria e della rete neurale. All'interno di Rete Neurale, inoltre, è stato aggiunto Ultimo Ripristino che consente di tener traccia dell'ultima

data in cui è stata resettata, per volontà dell'amministratore o per un malfunzionamento, in modo da ripristinare anche le prestazioni ad essa relativa.

All'interno dell'entità Categoria sono stati aggiunti gli attributi Tipo ed Int_Max. Il primo consente di specificare la tipologia dell'input che si vuole utilizzare per far inserire all'utente il valore per tale categoria, il secondo consente di specificare, qualora fosse necessario, un limite superiore per l'input considerato.



Schema concettuale

2.1.1.2. DIZIONARIO DEI DATI

Per tener traccia di tutte le entità ed associazioni presenti all'interno dello schema concettuale, con i relativi attributi ed identificatori, è stato realizzato un Dizionario dei Dati, ossia una struttura tabellare contenente tutti i dati prima citati:

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, Descrizione, Ultimo_Ripristino	Nome
Immagine	Immagine che l'utente desidera far classificare dalla rete neurale	Image	Image
Classificazione	Risultato dell'analisi della rete neurale	Codice, Data, Threshold, Valore_RN	Codice
Categoria	Caratteristica dell'immagine da classificare	Nome, Tipo, Int_Max	Nome

Dizionario dei Dati delle entità

Associazione	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

Dizionario dei Dati delle associazioni

2.1.1.3.REGOLE AZIENDALI

Per evidenziare determinate regole che non sono visualizzabili attraverso lo schema E-R si utilizzano le Regole Aziendali che si dividono in Regole di Vincolo e Regole di Derivazione.

Le regole di vincolo individuate per la realtà di interesse, ossia dei vincoli da rispettare, sono:

- Il risultato della classificazione della rete neurale deve essere riferito alle categorie che compongono la macrocategoria per cui la rete neurale è stata scelta
- La valorizzazione delle categorie per una specifica immagine deve essere riferita alle categorie che compongono la macrocategoria per l'immagine scelta
- All'inserimento o al ripristino di una rete neurale, l'attributo Valore_P deve essere NULL
- Dopo la prima classificazione effettuata da una rete neurale, l'attributo Valore_P deve essere maggiore o uguale di 0

Le regole di derivazione individuate per la realtà di interesse, ossia formule matematiche che permettono di derivare un particolare dato, sono:

- L'attributo Valore_P della Prestazione, quando i valori sono interi, a partire dalla prima classificazione è derivabile attraverso la formula:

$$\frac{1}{N} \sum_{n=0}^{N-1} |Valore_V_n - Valore_RN_n|$$

- L'attributo Valore_P della Prestazione, quando i valori sono stringhe, a partire dalla prima classificazione, è derivabile attraverso la formula:

$$\frac{1}{N} \sum_{n=0}^{N-1} (Valore_V_n == Valore_RN_n)$$

- L'attributo Valore_P della Prestazione, 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:

$$Valore_P = \frac{Valore_P * (N - 1) + |Valore_V - Valore_RN|}{N}$$

- L'attributo Valore_P della Prestazione, 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:

$$Valore_P = \frac{Valore_P * (N - 1) + (Valore_V == Valore_RN)}{N}$$

Dopo aver associato allo schema concettuale una buona documentazione di supporto contenente tutti gli elementi ed i vincoli che non possono essere rappresentati graficamente, bisogna verificare se tale schema è di buona qualità o meno analizzando le seguenti proprietà generali:

- Correttezza: tutti i costrutti devono essere usati correttamente sia dal punto di vista sintattico che semantico:
 - Sintattico: uso non ammesso di costrutti
 - Semantico: l'uso dei costrutti che non rispetta la loro definizione

- Completezza: tutti i dati di interesse devono essere rappresentati e tutte le operazioni richieste devono essere eseguibili a partire dai concetti descritti nello schema.
- Leggibilità: tutti i requisiti devono essere rappresentati in maniera naturale e facilmente comprensibile. Lo schema deve risultare autoesplicativo, mediante una scelta opportuna dei nomi ed una disposizione adeguata del diagramma.
- Minimalità: tutte le specifiche devono essere rappresentate una sola volta nello schema. In presenza di ridondanze si ottiene uno schema non minimale. Tali ridondanze potrebbero essere desiderate e, quindi, nate da opportune scelte progettuali, in tal caso devono opportunamente essere documentate.

Per la realtà di interesse lo schema concettuale ottenuto rispetta tali proprietà generali e, quindi, è definibile di buona qualità.

2.1.2. PROGETTAZIONE LOGICA

L'obiettivo della progettazione logica è quello di costruire uno schema logico in grado di descrivere, in maniera corretta ed efficiente, tutte le informazioni contenute nello schema E-R prodotto nella fase di progettazione concettuale.

Prima di procedere con la realizzazione dello schema logico, lo schema E-R deve essere sottoposto ad un'analisi del carico applicativo e ad una fase di ristrutturazione per soddisfare due esigenze:

- Semplificare la traduzione: non tutti i costrutti del modello E-R sono supportati dal modello relazionale.
- Ottimizzare il progetto: modifica dello schema concettuale in modo da renderlo più efficiente nell'esecuzione delle operazioni previste.

Attraverso una fase di traduzione dello schema ristrutturato ottenuto si produce lo schema logico.

2.1.2.1. CARICO APPLICATIVO

Il Carico Applicativo si riferisce ad una vera e propria analisi sullo Schema E-R in modo da regolare le prestazioni dei sistemi Software attraverso gli studi di massima di 2 parametri:

- Costo di Operazione: viene valutato in termini di numero di occorrenze di entità ed associazioni che mediamente devono essere visitate per rispondere ad un'operazione sulla base dati.
- Occupazione di Memoria: viene valutato in termini dello spazio di memoria, espresso in byte, necessario per memorizzare i dati descritti dallo schema.

Per ottenere questo carico si producono 2 tavole:

- Tavola dei Volumi: vengono riportati tutti i concetti dello schema cioè entità (E) ed associazioni (R) con il volume previsto a regime.
- Tavola delle Operazioni: viene riportata per ogni operazione la frequenza prevista a regime ed un simbolo che indica se l'operazione è interattiva (I) o batch (B). Un'operazione è interattiva se prevede un'interazione con l'utente e viene svolta in un tempo breve. Un'operazione è batch se ha un tempo di risposta lungo e non prevede la presenza dell'utente.

I volumi per la realtà di interesse sono:

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

È stato ipotizzato che il regime venga raggiunto dopo 30 giorni, che ogni rete neurale abbia effettuato a regime 300 classificazioni e che ad ogni inserimento di un'immagine da parte di un utente vengano classificate in media 2 categorie. Inoltre, il volume delle immagini è stato posto a 400, anziché 600 ($1200/2$), in quanto è possibile che una stessa immagine venga inserita più volte dall'utente senza produrre duplicati nel database. Il volume della relazione appartenenza è stato posto a 450, anziché 400 (numero di immagini), poiché una stessa immagine può essere inserita in riferimento a differenti macrocategorie. Infine, sono state ipotizzate a regime 900 valorizzazioni, anziché 1200 (numero di classificazioni), in quanto corrispondono al numero di istanze di appartenenza moltiplicate per il numero di categorie classificate in media.

Le operazioni principali per la realtà di interesse sono:

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

L'operazione 1 viene eseguita dopo ogni test per permettere all'utente di visualizzare il risultato e l'operazione 2 in seguito ad ogni classificazione. Siccome ogni test è costituito in media da 2 classificazioni, la frequenza dell'operazione 2 è pari al doppio di quella dell'operazione 1.

2.1.2.2. RISTRUTTURAZIONE

La fase di Ristrutturazione si occupa di dirigere lo schema E-R verso il Modello Relazionale e si può suddividere in una serie di passi da effettuare in sequenza:

- Analisi delle Ridondanze: viene deciso se eliminare o no eventuali ridondanze.
- Eliminazione delle Generalizzazioni: tutte le generalizzazioni vengono analizzate e sostituite con altri costrutti.
- Partizionamento/Accorpamento di entità ed associazioni: viene deciso se partizionare concetti in più parti o viceversa.
- Eliminazione Attributi Composti/Multivalore: tutti gli attributi non semplici vengono analizzati e sostituiti con altri costrutti.
- Scelta degli identificatori primari: viene scelto un identificatore principale per tutte le entità.

Analisi delle ridondanze

Una ridondanza in uno schema concettuale corrisponde alla presenza di un dato che può essere derivato (cioè ottenuto attraverso una serie di operazioni) da altri dati. In particolare, in uno schema E-R si possono presentare varie forme di ridondanza:

- Attributi derivabili, occorrenza per occorrenza, da altri attributi della stessa entità o associazione.
- Attributi derivabili da attributi di altre entità o associazioni.
- Attributi derivabili da operazioni di conteggio di occorrenze.
- Associazioni derivabili dalla composizione di altre associazioni in presenza di cicli.

La presenza di un dato derivato presenta un vantaggio e alcuni svantaggi:

- Il vantaggio è una riduzione degli accessi necessari per calcolare il dato derivato.
- Gli svantaggi sono una maggiore occupazione di memoria, spesso con un costo trascurabile, e la necessità di effettuare operazioni aggiuntive per mantenere il dato derivato aggiornato.

La decisione di mantenere o eliminare una ridondanza va quindi presa confrontando, nei casi di presenza e assenza della stessa, il costo di esecuzione delle operazioni che coinvolgono il dato ridondante e la relativa occupazione di memoria.

Per poter determinare il costo delle operazioni che coinvolgono una ridondanza è necessario sfruttare le tavole dei volumi e delle operazioni calcolate in precedenza per poter costruire le Tavole degli Accessi.

Queste tabelle sono composte da 4 colonne che rappresentano rispettivamente il nome dell'entità/associazione, il tipo di costrutto, il numero di accessi ed il tipo di accesso. La distinzione del tipo di accesso, in lettura (L) o scrittura (S), va fatta perché, generalmente, le operazioni di scrittura sono più onerose di quelle in lettura. Esse, infatti, devono essere eseguite in modo esclusivo e possono richiedere l'aggiornamento di indici, ossia strutture ausiliarie per l'accesso efficiente ai dati. Per tale motivo nell'ambito della valutazione delle ridondanze, un accesso in scrittura ha un costo doppio rispetto ad uno in lettura.

Nella realtà di interesse analizzata è stata identificata un'unica ridondanza causata dall'attributo Valore_P presente nell'associazione Prestazione. Il valore di tale attributo, infatti, è ottenibile attraverso le formule indicate nelle regole di derivazione, pertanto si tratta di un attributo derivabile da attributi di altre entità o associazioni.

- Per l'operazione 1 le tavole degli accessi sono:

Con Ridondanza:

Concetto	Costrutto	Accessi	Tipo
Rete_Neurale	E	1	L
Prestazione	R	2	L

Senza Ridondanza:

Concetto	Costrutto	Accessi	Tipo
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

- Per l'operazione 2, le tavole degli accessi sono:

Con Ridondanza:

Concetto	Costrutto	Accessi	Tipo
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	Tipo
-	-	-	-

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

- In presenza di ridondanza il costo delle operazioni è di circa 6420 accessi giornalieri.
- L'occupazione di memoria è di circa 32 byte, poiché sono necessari 4 byte per ogni istanza di prestazione.
- In assenza di ridondanza il costo delle operazioni è di 30100 accessi giornalieri.

Pertanto, si è deciso di mantenere la ridondanza in quanto riduce notevolmente il numero di accessi e l'occupazione di memoria è trascurabile.

Eliminazione delle generalizzazioni

Siccome i sistemi tradizionali per la gestione delle basi di dati non consentono di rappresentare in modo diretto una generalizzazione, è necessario eliminarle trasformandole in entità ed associazioni, ossia costrutti che possono essere implementati.

Per fare ciò esistono tre possibili alternative:

- Accorpamento delle figlie nel genitore: le entità figlie vengono eliminate ed i loro attributi vengono inseriti nell'entità padre che riceve un ulteriore attributo per identificare il tipo di una sua occorrenza. Infatti, nel caso di generalizzazione parziale, un'occorrenza dell'entità padre può appartenere a nessuna o ad una delle entità figlia, mentre nel caso di generalizzazione totale appartiene sicuramente ad una delle figlie. Questa alternativa è conveniente quando le operazioni non fanno distinzione tra le occorrenze e gli attributi del padre e delle figlie. Nonostante l'uso aggiuntivo di memoria, infatti, viene ridotto il numero di accessi.
- Accorpamento del genitore nelle figlie: l'entità genitore viene eliminata ed i suoi attributi, il suo identificatore e le associazioni a cui partecipava vengono aggiunti alle entità figlie. Questa soluzione è applicabile solamente se la generalizzazione è totale, altrimenti le occorrenze del padre che non sono occorrenze delle figlie non verrebbero rappresentate. Questa alternativa è conveniente quando sono presenti operazioni che fanno riferimento unicamente alle figlie. Si ottiene, infatti, un risparmio di memoria e non sono presenti attributi caratterizzati da valori nulli.
- Sostituzione della generalizzazione con associazioni: la generalizzazione si trasforma in due associazioni di tipo uno ad uno che legano l'entità genitore con le entità figlie. Non ci sono trasferimenti di attributi o associazioni e le entità figlie sono identificate esternamente dall'entità padre. Vanno aggiunti, inoltre, dei vincoli: ogni occorrenza dell'entità padre non può partecipare contemporaneamente a più di una delle nuove associazioni e, se la generalizzazione è totale, deve partecipare necessariamente ad una di esse. Questa alternativa è conveniente quando la generalizzazione non è totale e sono presenti operazioni che si riferiscono ad occorrenze solo delle figlie o solo del genitore. In questo caso non sono presenti valori nulli, ma si verifica un incremento degli accessi per ottenere la consistenza delle occorrenze rispetto ai vincoli introdotti.

Nello schema concettuale non sono presenti generalizzazioni. Per tale motivo questa fase è stata saltata.

Partizionamento ed Accorpamento di concetti

Per garantire una maggior efficienza delle operazioni è possibile partizionare o accorpare entità o associazioni secondo il seguente principio: gli accessi si riducono separando attributi di uno stesso concetto che vengono acceduti da operazioni diverse e raggruppando attributi di concetti diversi che vengono acceduti dalle medesime operazioni.

Un'entità o associazione può essere partizionata applicando una delle seguenti decomposizioni:

- La decomposizione verticale prevede la suddivisione del concetto sulla base dei suoi attributi.
- La decomposizione orizzontale prevede la suddivisione del concetto sulla base delle occorrenze delle entità. Tale decomposizione corrisponde ad una generalizzazione a livello logico.

L'accorpamento di entità, operazione inversa del partizionamento, è utile quando esistono operazioni frequenti su più entità. In questo caso, queste ultime possono essere accorpate in un'unica entità che contiene tutti gli attributi. Di solito, gli accorpamenti si effettuano su entità legate da associazioni di tipo uno ad uno, raramente su entità legate da associazioni uno a molti e mai su entità legate da associazioni molti a molti. Questo perché gli ultimi due casi generano ridondanze.

Spesso è conveniente partizionare o accorpare associazioni. È utile decomporre un'associazione tra due entità in due o più associazioni tra le medesime entità per separare occorrenze dell'associazione originale accedute sempre separatamente. Inoltre, è utile accorpare due o più associazioni tra le medesime entità quando le relative occorrenze sono sempre accedute contemporaneamente.

Nell'ambito dello schema concettuale in questione si è ritenuto opportuno non effettuare alcun tipo di partizionamento o accorpamento in quanto esso è costituito da entità ed associazioni con pochi attributi e le operazioni accedono ad esse in modo efficiente.

Eliminazione degli attributi composti e multivalore

Siccome i DBMS non supportano gli array, è necessario eliminare gli attributi composti dallo schema concettuale. Questo avviene inserendo singolarmente nell'entità contenente l'attributo composto tutti gli attributi che lo costituiscono lasciando inalterate le cardinalità. Inoltre, è opportuno aggiungere un vincolo aggiuntivo per indicare che sono ammesse solo due possibilità nella valorizzazione degli attributi che originariamente costituivano l'attributo composto. Essi, infatti, devono essere contemporaneamente valorizzati o nulli.

Anche gli attributi multivalore devono essere eliminati dallo schema. Per fare ciò si aggiunge un'entità con lo stesso nome dell'attributo multivalore e identificata da un attributo aggiuntivo che ne descrive il concetto. La nuova entità viene collegata a quella originaria attraverso un'associazione che presenta dal lato dell'entità preesistente la stessa cardinalità posseduta originariamente dall'attributo multivalore. Verso il lato della nuova entità, invece, la cardinalità può assumere diversi valori in base alla realtà di interesse.

Con quest'ultimo procedimento è possibile eliminare anche gli attributi composti multivalore.

Nell'ambito del progetto, lo schema concettuale non contiene attributi composti o multivalore.

Per tale motivo questa fase è stata saltata.

Scelta degli identificatori principali

Nel modello relazionale le chiavi vengono utilizzate per stabilire legami tra dati in relazioni diverse. Inoltre, in genere i DBMS richiedono una chiave primaria sulla quale vengono costruite automaticamente delle strutture ausiliarie, dette indici, per reperire i dati in modo efficiente.

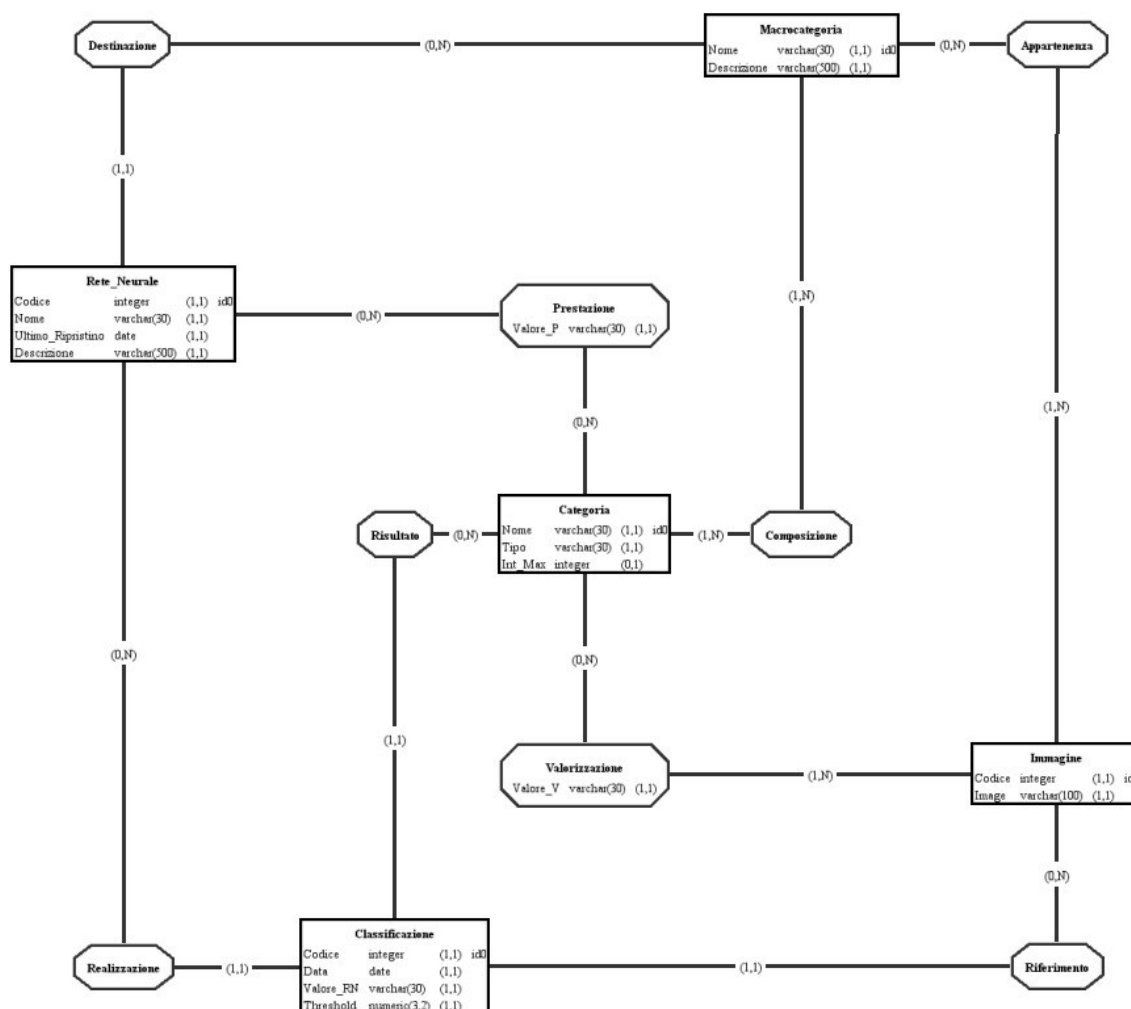
In questa fase, per ogni entità dello schema E-R si sceglie l'identificatore principale, che rappresenta la chiave primaria che verrà usata dal DBMS.

I criteri di decisione per effettuare la scelta sono i seguenti:

- gli attributi con valori nulli non possono costituire identificatori principali;
- un identificatore composto da uno o da pochi attributi è da preferire a identificatori costituiti da molti attributi. Questo garantisce dimensioni ridotte degli indici, un risparmio di memoria nella realizzazione dei legami logici tra le relazioni e una facilitazione nelle operazioni di join;
- un identificatore interno con pochi attributi è da preferire ad un identificatore esterno che coinvolge diverse entità. Questo criterio esiste per le stesse motivazioni di quello precedente;
- un identificatore che viene utilizzato da molte operazioni per accedere alle occorrenze di un'entità è da preferire rispetto agli altri;

Per la realtà di interesse le uniche entità che hanno subito una variazione di identificativo principale sono Rete Neurale ed Immagine.

Per le entità Rete Neurale ed Immagine, inizialmente identificate rispettivamente da Nome ed Image, si è optato per l'aggiunta di un identificatore Codice da utilizzare come principale in modo da rendere l'entità più facile da identificare, evitando ambiguità.



Schema ristrutturato

2.1.2.3. TRADUZIONE

La fase di traduzione si occupa, come dice il nome di tradurre i modelli di dati diversi in modo da ottenere, a partire da uno schema E-R ristrutturato, uno schema logico compatibile con il modello relazionale in grado di rappresentare le medesime informazioni.

Per tradurre correttamente uno schema E-R ristrutturato in uno schema logico relazionale devono essere utilizzati dei pattern ben definiti:

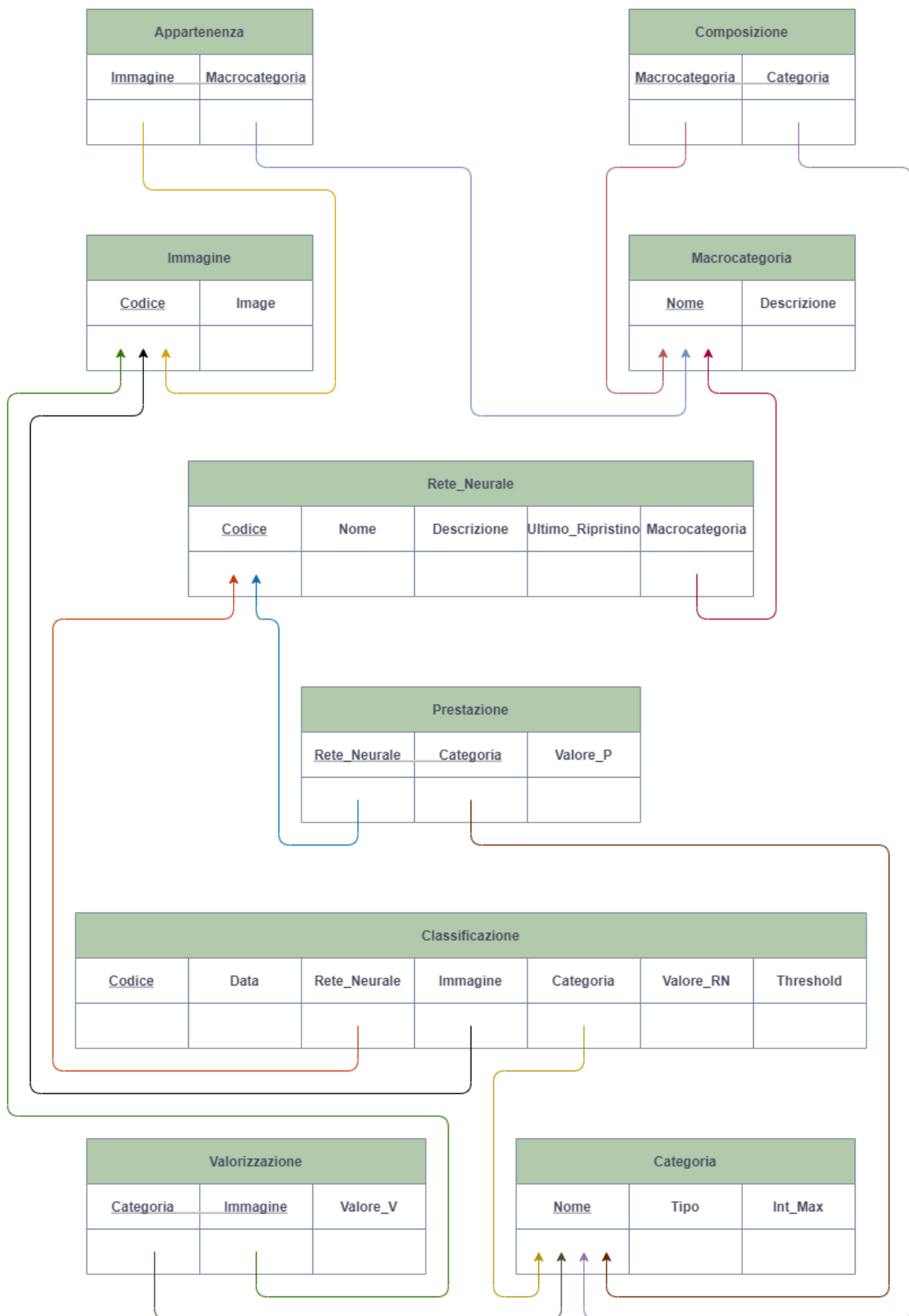
- Pattern per entità forti: un'entità forte è un'entità che possiede l'identificatore principale interno e viene tradotta in una relazione con lo stesso nome e gli stessi attributi. L'identificatore principale assume il ruolo di chiave primaria, mentre altri identificatori quello di chiave alternativa,
- Pattern per associazioni molti a molti: per ogni entità viene prodotta una relazione con lo stesso nome avente per attributi i medesimi attributi dell'entità e per chiave il suo identificatore. Per l'associazione viene prodotta una relazione con lo stesso nome avente per attributi gli attributi dell'associazione e gli attributi identificatori delle entità coinvolte, tali identificatori formano la chiave della relazione.
- Pattern per associazioni uno a molti: per la traduzione di questo tipo di associazione possono essere applicati 2 differenti metodi:
 - Tradurre sia le entità forti che l'associazione in relazioni ed aggiungere in quella riferita all'associazione gli attributi chiave delle altre, rendendo quello della relazione riferita all'entità dove la cardinalità massima è 1 la chiave primaria della relazione,
 - Tradurre solo le entità forti in tabelle, tutti gli attributi che fanno parte dell'associazione vengono inglobati nella relazione riferita all'entità dove la cardinalità massima è 1. L'attributo chiave della relazione con cardinalità massima N diventa attributo dell'altra
- Pattern per associazione uno ad uno: per la traduzione di questo tipo di associazione possono essere applicati 3 differenti metodi:
 - Tradurre sia le entità forti che l'associazione in relazioni ed aggiungere in quella riferita all'associazione gli attributi chiave delle altre, rendendo uno dei due la chiave primaria della relazione e unique l'altro.
 - Tradurre solo le entità forti in tabelle, tutti gli attributi che fanno parte dell'associazione vengono inglobati in una delle due relazioni. L'attributo chiave della relazione che non ha inglobato l'associazione diventa attributo dell'altra. Vengono introdotti, inoltre, vincoli aggiuntivi.
 - Stesso procedimento del secondo metodo, ma vengono utilizzati dei vincoli di integrità referenziale incrociati.

Tranne che per il primo pattern descritto, è necessario inserire gli opportuni vincoli di integrità referenziale, ossia dei vincoli tra relazioni che affermano che se una tupla in una relazione A fa riferimento ad un'altra relazione B, essa deve far riferimento ad una tupla esistente in B.

Infine, è necessario inserire, dove opportuno, i vincoli per soddisfare le cardinalità minime pari ad 1. Nell'ambito della traduzione dello schema ristrutturato, per tradurre le associazioni uno a molti ed uno ad uno è stato utilizzato il secondo metodo del rispettivo pattern.

Relazione	Chiave Alternativa
IMMAGINE (<u>Codice</u> , Image)	Image
APPARTENENZA (<u>Macrocategoria</u> , <u>Immagine</u>)	
MACROCATEGORIA (<u>Nome</u> , Descrizione)	
RETE_NEURALE (<u>Codice</u> , Nome, Descrizione, Ultimo_Ripristino, Macrocategoria)	Nome
CLASSIFICAZIONE (<u>Codice</u> , Data, Rete_Neurale, Immagine, Categoria, Valore_RN, Threshold)	
CATEGORIA (<u>Nome</u> , Tipo, Int_Max)	
VALORIZZAZIONE (<u>Categoria</u> , <u>Immagine</u> , Valore_V)	
COMPOSIZIONE (<u>Macrocategoria</u> , <u>Categoria</u>)	
PRESTAZIONE (<u>Rete_Neurale</u> , <u>Categoria</u> , Valore_P)	

Schema logico



Rappresentazione grafica dello schema logico

2.1.3. NORMALIZZAZIONE

Per verificare la qualità di uno schema logico si utilizza la Teoria della Normalizzazione. Quest'ultima nasce come una procedura che permette di trasformare schemi di relazioni non normalizzate in schemi che soddisfano una o più forme normali. Nell'ambito della progettazione, invece, rappresenta solamente una tecnica di verifica.

Gli attributi all'interno delle relazioni possono essere classificati come primi se fanno parte della chiave e non primi altrimenti.

Per poter utilizzare tale tecnica è necessario individuare le dipendenze funzionali presenti nello schema logico.

Una dipendenza funzionale è un particolare vincolo di integrità per il modello relazionale che, come ci suggerisce il nome, descrive legami di tipo funzionale tra gli attributi di una relazione, ed è molto utile per scoprire e rimuovere le anomalie in uno schema del modello logico.

Data una relazione r su uno schema $R(X)$ e due sottoinsiemi di attributi non vuoti Y e Z di X , diremo che esiste su r una dipendenza funzionale tra Y e Z ($Y \rightarrow Z$) se e solo se: per ogni coppia di tuple t_1 e t_2 di r aventi gli stessi valori sugli attributi Y , risulta che t_1 e t_2 hanno gli stessi valori anche sugli attributi Z . In termini matematici la dipendenza funzionale si definisce nel modo seguente

$$\begin{aligned} &\forall \text{ istanza lecita } r \text{ di } R: \\ &\forall t_1, t_2 \in r : t_1[Y] = t_2[Y] \Rightarrow t_1[Z] = t_2[Z] \end{aligned}$$

Una dipendenza funzionale viene definita banale se l'insieme di attributi implica un suo sottoinsieme. Si definisce Forma Normale una proprietà di uno schema relazionale che ne garantisce la qualità.

È possibile affermare che uno schema è in una determinata forma normale, se tutte le relazioni che lo costituiscono appartengono a tale forma.

Esistono 4 forme normali:

- Prima Forma Normale: Il dominio degli attributi deve comprendere solo valori atomici, semplici e indivisibili, pertanto non deve contenere attributi composti e multivalore
- Seconda Forma Normale: La relazione è in prima forma normale ed ogni attributo non primo dipende in maniera piena dalla chiave di $R(X)$
- Forma Normale di Boyce & Codd: La relazione è in prima forma normale e per ogni dipendenza funzionale $Y \rightarrow Z$ (non banale) definita su $R(X)$, Y contiene la chiave K di $R(X)$, cioè Y è una superchiave di $R(X)$
- Terza Forma Normale: La relazione è in seconda forma normale e per ogni DF non banale $Y \rightarrow Z$ definita su $R(X)$: Y è chiave di $R(X)$ oppure Z è un attributo primo

Nella realtà di interesse per ciascuna relazione le dipendenze funzionali sono:

Relazione	Dipendenze Funzionali
IMMAGINE (<u>Codice</u> , Image)	Codice \rightarrow Image Image \rightarrow Codice
APPARTENENZA (<u>Macrocategoria</u> , <u>Immagine</u>)	
MACROCATEGORIA (<u>Nome</u> , Descrizione)	Nome \rightarrow Descrizione
RETE_NEURALE (<u>Codice</u> , Nome, Descrizione, Ultimo_Ripristino, Macrocategoria)	Codice \rightarrow Nome, Descrizione, Ultimo_Ripristino, Macrocategoria Nome \rightarrow Codice, Descrizione, Ultimo_Ripristino, Macrocategoria
CLASSIFICAZIONE (<u>Codice</u> , Data, Rete_Neurale, Immagine, Categoria, Valore_RN, Threshold)	Codice \rightarrow Data, Rete_Neurale, Immagine, Categoria, Valore_RN, Threshold Rete_Neurale, Immagine, Categoria \rightarrow Codice, Data, Valore_RN, Threshold
CATEGORIA (<u>Nome</u> , Tipo, Int_Max)	Nome \rightarrow Tipo, Int_Max
VALORIZZAZIONE (<u>Categoria</u> , <u>Immagine</u> , Valore_V)	Categoria, Immagine \rightarrow Valore
COMPOSIZIONE (<u>Macrocategoria</u> , <u>Categoria</u>)	
PRESTAZIONE (<u>Rete_Neurale</u> , <u>Categoria</u> , Valore_P)	Rete_Neurale, Categoria \rightarrow Valore_P

Le relazioni ottenute rispettano i requisiti della prima forma normale, che altro non sono che requisiti del modello relazionale, curati durante la progettazione logica.

Dall'analisi, invece, delle dipendenze funzionali, possiamo affermare che le relazioni rispettano i requisiti della seconda forma normale, in quanto esse si trovano in prima forma normale e tutti gli attributi non chiave dipendono dall'intera chiave.

Dall'analisi delle dipendenze funzionali, è possibile 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 dipende in modo non transitivo da ogni chiave della relazione.

Dall'analisi delle dipendenze funzionali, possiamo affermare 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 ed ogni determinante delle dipendenze funzionali individuate è una chiave candidata.

2.1.4. TRIGGER

Per rendere attivo e reattivo un database è necessario definire delle regole attive, chiamate trigger. Un trigger è una procedura, ossia un insieme di istruzioni, eseguita autonomamente dal sistema, in conseguenza ad un evento che può essere inserimento, cancellazione o aggiornamento di una data relazione. Quando un trigger è attivato, esso provoca l'esecuzione di una funzione, specificata al momento della definizione dello stesso.

Le regole seguono il cosiddetto paradigma Evento-Condizione-Azione:

- Evento: quando accade, il trigger viene attivato.
- Condizione: predicato che identifica se l'azione del trigger deve essere eseguita.
- Azione: sequenza di primitive SQL generiche o una procedura.

Nella realtà di interesse sono stati implementati 2 trigger di inizializzazione e 3 di vincolo.

Il primo trigger di inizializzazione ha l'obiettivo di aggiornare le prestazioni di una rete neurale relativamente alle categorie da lei classificate in seguito ad ogni test. Tale trigger scatta in seguito ad un inserimento all'interno della relazione Classificazione.

Il secondo trigger di inizializzazione ha l'obiettivo di resettare le prestazioni di una rete neurale relativamente alle categorie da lei classificate in seguito ad un ripristino. Tale trigger scatta in seguito ad un aggiornamento dell'attributo Ultimo_Ripristino all'interno della relazione Rete_Neurale.

I trigger di vincolo si occupano di implementare i vincoli dovuti alle cardinalità minime aggiunti durante la fase di traduzione dello schema ristrutturato. In particolare, essi scattano rispettivamente in seguito all'inserimento di una tupla all'interno delle relazioni Macrocategoria, Categoria ed Immagine. Tali trigger, inoltre, sono ritardati, in quanto devono mantenere il database in uno stato consistente, ossia uno stato in cui tutti i vincoli devono essere rispettati.

2.2. PROGETTAZIONE DEL PORTALE WEB

Per la progettazione del portale web è stato utilizzato uno dei template messi a disposizione all'interno della libreria di Bootstrap.

La scelta del template utilizzato è stata effettuata sulla base di esigenze grafiche e stilistiche che potessero catturare l'attenzione di un utente e che fossero facilmente riusabili in modo da garantire una possibile estensione rispetto alle specifiche fornite.

Per tali motivazioni è stata eseguita un'attenta analisi di tutti i template prestando particolare attenzione alle sezioni, animazioni e strutture grafiche al loro interno.

La pagina web contenuta nel template scelto, caratterizzata dai colori giallo e nero, è stata utilizzata come Homepage, ossia la pagina iniziale che un utente può raggiungere in seguito ad una ricerca su un qualsiasi browser.

Sulla base di questa è stata progettata la creazione di tre ulteriori pagine: quella che permette di effettuare un test, quella che permette di visualizzare i risultati e quella di registrazione.

2.2.1. HOMEPAGE

Inizialmente, tale pagina era costituita da molteplici sezioni suddivise in modo da essere raggiungibili attraverso un semplice scroll o una selezione all'interno della barra di navigazione. È stato stabilito il mantenimento di tale struttura decrementando le sezioni per renderle adeguate alle informazioni della realtà di interesse da mostrare agli utenti.

Si è scelto di modificare l'header sostituendo il logo preesistente con quello del portale web (IC.), destinato a ricaricare la pagina, ed aggiungendo sulla destra i pulsanti di accesso (Accedi) e registrazione (Registrati) per un utente non loggato. Al posto di questi ultimi, un utente loggato visualizza un'icona ed il pulsante di logout (Esci).

Per garantire l'accesso è stato deciso di inserire all'interno del footer, accanto ad informazioni generali quali indirizzo della sede aziendale ed e-mail dei membri del Team che hanno lavorato al sito, un form facilmente raggiungibile con un click sul pulsante Accedi. Tale form permette all'utente non loggato l'inserimento di username e password.

Per il pulsante Registrati, invece, è stato definito il reindirizzamento ad una nuova pagina, quella di registrazione, progettata in seguito.

La strutturazione delle sezioni della Homepage è stata stabilita nel seguente modo:

- Home: al centro è posto il nome del portale web rispettando i colori del template scelto, seguito da un sottotitolo e da 4 pulsanti raffiguranti le macrocategorie messe a disposizione. Tali pulsanti sono cliccabili solamente da utenti loggati. Essi posseggono animazioni differenti a seconda se l'utente è loggato o meno.
- Info: sono riportate informazioni utili sul funzionamento del portale ed una serie di step da seguire per poter effettuare un test. All'interno del titolo di tale sezione, inoltre, è utilizzato un gioco di parole in modo da inserire le lettere caratterizzanti il logo che, se cliccate, reindirizzano alla sezione Home.
- Reti Neurali: sono poste nella parte superiore tante schede quante sono le reti neurali da poter testare, insieme ad una breve descrizione. Passando con il mouse su una di esse la si mette in evidenza. Nella parte inferiore, invece, è riportata una spiegazione di cosa sono le reti neurali ed un pulsante che riporta alla pagina specifica di Wikipedia per ulteriori informazioni.
- Portfolio: sono riportate nella parte superiore alcune immagini di esempio suddivise in base a ciò che le reti neurali sono in grado di riconoscere: facce, persone e veicoli. È possibile visualizzare le immagini per gruppi cliccando gli appositi pulsanti. Passando con il mouse su un'immagine, quest'ultima si evidenzia ed in basso a sinistra appare un'icona che permette di visualizzarla ingrandita. Nella parte centrale è presente un riquadro in cui viene riportato il numero di immagini per ogni macrocategoria, prelevato dal database, su cui gli utenti hanno testato le reti neurali. Nella parte inferiore, infine, è presente un pannello a scorrimento in cui vengono presentate le macrocategorie con un'immagine che le rappresenta e la relativa descrizione.
- Team: sono riportate le schede dei membri del Team che ha lavorato al portale web mostrando per ognuna di esse il nome, la foto e la professione.
- Contatti: è riportata nella parte superiore una mappa interattiva con la geolocalizzazione della sede dell'azienda, ossia l'Università degli Studi di Salerno. Nella parte inferiore, invece, sono presenti l'indirizzo della sede, qualora la mappa non fosse visibile, l'e-mail di contatto ed un pulsante che consente l'apertura automatica del mail client predefinito presente sul dispositivo con il campo del destinatario già compilato.

Per ogni sezione sono state selezionate immagini ed icone da inserire in base al contenuto mostrato.

2.2.2. TEST

La pagina Test.html è finalizzata alla visualizzazione del form da compilare per effettuare il test. È stato deciso di riutilizzare l'Header della Homepage eliminandone la barra di navigazione e sostituendo i pulsanti con una breadcrumb in modo da permettere di tornare alla Homepage. Anche il Footer della Homepage è stato riutilizzato eliminando il form per l'accesso.

Dopo la scelta dell'immagine di sfondo, è stato deciso di posizionare al centro della regione superiore della pagina il nome della Macrocategoria selezionata caratterizzato dal colore giallo e da un sottile bordo nero.

È stato scelto, inoltre, di includere il form in una card inserita al centro della pagina e dividerlo in due regioni uguali. La prima contiene un riquadro tratteggiato in cui sarà visualizzata l'immagine una volta caricata dall'utente e la seconda contiene i campi da valorizzare. I campi sono costituiti dalle categorie che compongono la macrocategoria scelta, dalla rete neurale da scegliere per effettuare la classificazione e dalla sensibilità della rete, parametro chiamato Threshold.

In particolare, per la macrocategoria Face Analysis, le categorie da valorizzare sono Età, Genere, Etnia ed Emozione.

Si è scelto di progettare la compilazione delle categorie attraverso input differenti. In particolare, per la categoria Età è stato utilizzato un input di tipo number, in modo da permettere all'utente di selezionare il numero desiderato o inserirlo manualmente. Per Genere ne è stato utilizzato uno di tipo radio, in modo da permettere un'unica scelta tra le opzioni disponibili. Per Etnia ed Emozione ne è stato utilizzato uno di tipo select in modo da permettere una scelta da un elenco di opzioni, visualizzato come menu a tendina. Anche la scelta della Rete Neurale avviene mediante un input di tipo select in cui possono essere selezionate solo le reti neurali disponibili per la macrocategoria scelta. Nel caso di Face Analysis, l'unica rete che può essere scelta è Face Bio. Infine, si è ritenuto opportuno implementare l'impostazione del Threshold attraverso un input di tipo range per renderla semplice e piacevole da utilizzare.

I campi sono stati progettati in riquadri che compaiono e scompaiono in funzione di due pulsanti. In particolare, al click sul primo pulsante identificato dall'espressione: "Classifica le categorie", si apre il riquadro contenente i nomi delle categorie ed i relativi input da compilare, mentre al click sul secondo, caratterizzato dall'espressione: "Definisci la rete neurale", compare un riquadro che permette di selezionare la rete neurale ed impostare la sua sensibilità. Al caricamento della pagina i riquadri risultano già aperti.

È stata prevista la presenza di una scroll-bar all'interno del riquadro contenente le categorie da visualizzare. In questo modo è possibile garantire una modularità dei contenuti.

Infine, sono stati progettati due tasti, uno per permettere di caricare l'immagine ed uno per confermare i dati immessi, far partire il test ed indirizzare alla pagina del risultato. Il primo è stato idealizzato di colore scuro, con la scritta bianca ed un bordo giallo, mentre il secondo di colore giallo e con la scritta nera.

2.2.3. RESULT

La pagina Result.html è finalizzata alla visualizzazione del risultato del test. È stata scelta una struttura della pagina con molti elementi in comune con quella della pagina di test.

In questo caso, però, i riquadri sono tre:

- Il primo contiene i valori di ogni categoria immessi dall'utente
- Il secondo contiene i valori forniti dalla rete neurale in seguito alla classificazione effettuata
- Il terzo contiene alcune informazioni sulla rete, ovvero le prestazioni relative ad ogni categoria, il Threshold selezionato dall'utente ed il numero di immagini classificate

Per ognuno di essi è stata prevista la presenza di una scroll-bar al fine di mantenere un assetto modulare.

L'immagine visualizzata nella pagina di test viene visualizzata nuovamente in questa pagina.

Infine, è stato progettato un tasto per permettere il caricamento di una nuova immagine e reindirizzare nuovamente alla pagina di test. Questo è stato idealizzato di colore scuro, con la scritta bianca ed un bordo giallo.

2.2.4. REGISTRAZIONE

La pagina di registrazione è una delle pagine da creare in aggiunta alla Homepage già fornita dal template. Per tale pagina, è stata inizialmente scelta un'immagine di sfondo e sono stati progettati gli stessi header e footer presenti all'interno delle pagine del test e del risultato.

Al centro della pagina sono stati posti il titolo "REGISTRAZIONE" ed il form per l'inserimento dei campi dell'utente in un riquadro caratterizzato da un bordo giallo ed arrotondato agli angoli e da uno sfondo leggermente sfocato rispetto a quello dell'intera pagina.

I campi scelti per la registrazione sono: Nome, Cognome, E-mail, Username, Password e Conferma Password. Si è stabilita, inoltre, la presenza e la posizione dei tasti Registrati e Reset. Il primo consente l'invio dei dati ed è stato posizionato al di sotto dei campi sulla sinistra, mentre il secondo consente di resettare i campi per eliminare i valori inseriti ed è stato posizionato al di sotto dei campi sulla destra.

Infine, il footer contiene sulla sinistra gli stessi elementi del footer della Homepage e sulla destra non presenta la visualizzazione condizionata del form per l'accesso al portale.

3. CAPITOLO 3

IMPLEMENTAZIONE

In questa fase sono state implementate tutte le funzionalità descritte nella fase di progettazione del portale web.

In aggiunta al file html incluso nel template, prelevato dalla libreria di Bootstrap, denominato Homepage.html sono state realizzate ulteriori pagine con la medesima estensione:

- Registrazione.html
- Test.html
- Result.html
- common.html
- login.html

Prima di poter procedere con tale implementazione, però, è stato creato il progetto denominato IC_dj costituito dall'applicazione ImageClassifier.

La creazione del progetto ha prodotto una cartella omonima contenente i seguenti file python che consentono l'esecuzione del sito:

- manage.py: permette di automatizzare le operazioni di Django. È in grado di avviare un web server locale
- __init__.py: file vuoto che indica a Python che la directory deve essere considerata come un pacchetto Python
- settings.py: contiene le impostazioni del progetto Django come la definizione dei path per l'applicazione e la configurazione per l'accesso al DBMS
- urls.py: contiene la configurazione degli URL richiesti in invocazioni verso le componenti del sistema
- asgi.py: rappresenta il punto di ingresso per i server web compatibili con ASGI (Asynchronous Server Gateway Interface)
- wsgi.py: rappresenta il punto di ingresso per i server web compatibili con WSGI (Web Server Gateway Interface)

La creazione dell'applicazione ha prodotto una cartella omonima contenente i seguenti file Python:

- admin.py: utility di Django per i compiti amministrativi
- apps.py
- models.py: contiene i modelli dei dati
- test.py
- views.py: contiene le viste Python per gestire le interazioni con l'utente

Viene, inoltre, generata la cartella "migrations" contenente tutte le modifiche apportate allo schema del database sotto forma di file Python.

Quando un utente richiede una pagina da un sito realizzato con Django, lo urlresolver di quest'ultimo esegue una serie di passi per permetterne la visualizzazione:

- Django determina il modulo Python root URLconf da utilizzare. Di solito, quest'ultimo è il valore assegnato all'impostazione ROOT_URLCONF in settings.py che di default fa riferimento al file urls.py del progetto.
- Django carica il modulo e cerca la variabile urlpatterns. Questa è costituita da una sequenza di istanze della funzione django.urls.path(). La funzione path() prende in ingresso quattro argomenti, non tutti obbligatori, e restituisce un oggetto che può essere incluso negli urlpatterns. I parametri di ingresso sono:

- route: stringa che contiene un pattern URL.
- view: vista che si vuole invocare. Questo parametro può essere anche il risultato del metodo `as_view()` o del metodo `django.urls.include()`.
- kwargs: permette di passare parametri aggiuntivi alla vista o al metodo utilizzato per chiamarla.
- name: stringa che denomina il pattern URL.
- Django scorre i pattern in ordine e si ferma al primo che corrisponde all'URL richiesto
- Determinato il pattern, Django importa ed invoca la vista specificata nella funzione `path()` del pattern URL passandole alcuni argomenti tra cui un'istanza della classe `HttpRequest`
- Se nessun pattern URL corrisponde o se viene sollevata una eccezione, Django invoca una vista di gestione degli errori.

Una pratica modulare ed efficiente consigliata da Django è quella di evitare che il file globale `urls.py`, contenente gli indirizzi del progetto, gestisca gli URL e le viste di tutte le applicazioni esistenti. Si delega, quindi, la gestione al rispettivo file `urls.py` specifico per ogni app. Per tale motivo è stato creato il file `urls.py` nella cartella dell'applicazione `ImageClassifier`. Al suo interno sono stati importati la funzione `path()` da `django.urls` ed il modulo `views` dalla cartella corrente. È stata definita, inoltre, una vista denominata `urlpatterns` finalizzata a contenere le funzioni `path()` per la gestione degli URL dell'applicazione. Infine, nel file `urls.py` all'interno del progetto è stata inserita, nella variabile `urlpatterns`, un'istanza della funzione `path()`. Tale istanza ha come primo argomento l'URL relativo all'applicazione, ossia `"ImageClassifier/"`, e come secondo la funzione `include()` in cui è stato specificato il file da includere, ossia `"ImageClassifier.urls"`.

Attraverso l'utilizzo dell'interfaccia grafica PgAdmin, che gestisce i database di PostgreSQL, è stato realizzato il database denominato IC e sono stati eseguiti i trigger descritti all'interno del Capitolo 2. Il progetto è in grado di dialogare con tale database grazie alla configurazione avvenuta nella sezione DATABASES all'interno del file `settings.py`. In tale sezione sono stati impostati i parametri necessari alla connessione con il database. Per sincronizzare il tutto è stata eseguita, da riga di comando, l'operazione `$python manage.py migrate`.

Per poter aggiungere o modificare le relazioni presenti all'interno del database è possibile definire nuovi modelli o modificare quelli esistenti all'interno del file `models.py`. Per poter rendere effettive queste operazioni è necessario aggiungere l'applicazione nella sezione `INSTALLED_APPS` del file `setting.py`.

Per rappresentare lo schema logico ottenuto durante la fase di progettazione del database nel Capitolo 2, sono stati definiti i modelli necessari al corretto funzionamento dell'applicazione. I modelli sono classi Python i cui attributi rappresentano i campi delle relazioni del database. Ogni tupla di una relazione possiede un campo ID, ossia un numero progressivo generato automaticamente, utilizzato come riferimento univoco.

Per poter comunicare una modifica all'interno dei modelli e, conseguentemente, apportarla al database, è necessario attuare le migrazioni. Le migrazioni costituiscono un meccanismo di modifica del database efficace e potente in quanto permettono di apportare cambiamenti senza generare perdite di dati.

I modelli creati ed i dati presenti al loro interno possono essere gestiti attraverso un'interfaccia di amministrazione messa a disposizione da Django. All'interno di tale interfaccia sono presenti, oltre ai modelli creati, i campi `Groups` e `Users` forniti da `django.contrib.auth` che permettono rispettivamente di controllare i permessi attribuiti ad un gruppo di utenti e gli utenti stessi. Per poter accedere all'interfaccia è necessario creare un superuser che possieda i permessi di amministratore.

In Django le pagine web ed i contenuti mostrati agli utenti sono gestiti attraverso il meccanismo delle view.

Quando viene richiesto uno specifico URL, attraverso una richiesta GET o POST, viene invocata una view implementata attraverso le funzioni Python specifiche della logica applicativa. Una view è una funzione che prende in ingresso una richiesta rappresentata da un oggetto `HttpRequest` e restituisce una risposta rappresentata da un oggetto `HttpResponse` o solleva un'eccezione.

Siccome le view non devono implementare il layout delle pagine, vengono utilizzati i template Django. Un template Django è un file html che contiene al suo interno comandi specifici che implementano il contenuto dinamico ottenuto attraverso le view.

I template rappresentano unicamente il contenuto delle pagine web. In questo modo si ottiene un disaccoppiamento tra contenuto e logica di programmazione. I template hanno il vantaggio di essere semplici e riutilizzabili e sono caratterizzati dall'ereditarietà e modularità. Ogni template, infatti, può ereditare da un altro, chiamato template padre, e modificarne o condizionarne il contenuto.

I costrutti principali utilizzati all'interno dei template Django sono:

- variabili: consentono di scambiare dati tra template e view.
- tag: consentono di inserire logica di programmazione nel processo di rendering e di produzione del contenuto.
- filtri: trasformano i valori delle variabili e degli argomenti dei tag.
- commenti.

Il collegamento tra template e view avviene quando quest'ultima restituisce un oggetto `HttpResponse` attraverso la funzione `render()`. Tale funzione combina un determinato template ad un context. Un context rappresenta una struttura dati, in particolare un dizionario, che associa a variabili utilizzate nel template i valori ottenuti nella view.

Un'altra funzione delle view che restituisce un oggetto `HttpResponse` è `redirect()`. Tale funzione riporta all'indirizzo corrispondente all'argomento passato come parametro. Quest'ultimo può essere un modello, una vista o un URL assoluto o relativo

All'interno della cartella dell'applicazione sono state create due ulteriori cartelle denominate rispettivamente "templates" e "static". Nella prima sono stati inseriti i template da implementare, costituiti dalle pagine html create in precedenza. Nella seconda, invece, sono stati inseriti i file css e javascript necessari al corretto funzionamento del sito e le immagini statiche visualizzate nelle pagine.

3.1. HOMEPAGE

In accordo con quanto detto nella fase di progettazione della Homepage nel Capitolo 2, sono stati aggiunti nell'header del corrispondente template, denominato `Homepage.html`, i pulsanti `Accedi` e `Registrati`. Tali elementi sono stati realizzati attraverso l'uso di ancore e sono stati inseriti all'interno di una table costituita da una riga e due colonne. Inoltre, è stato sostituito il logo preesistente con quello del portale web. Quest'ultimo è stato realizzato attraverso un'ancora che, se cliccata, richiama la view `Homepage` che ha il compito di renderizzare il template `login.html`, facendo aggiornare la pagina.

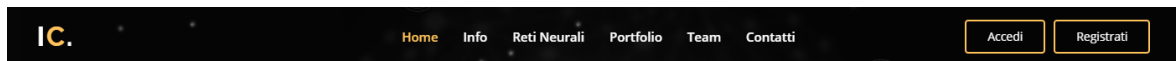
La posizione degli elementi dell'header è stata impostata agendo sui padding e sui margini.

All'interno del footer è stato inserito un form che consente ad un utente già registrato di inserire il proprio username e la propria password per accedere al portale web e sbloccare la possibilità di eseguire i test delle reti neurali.

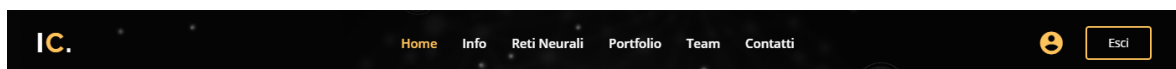
Per poter raggiungere comodamente tale form, è stato inserito all'interno dell'attributo href del pulsante Accedi l'id del div contenente il form. L'attributo href del pulsante Registrati, invece, contiene l'URL della view chiamata Registrazione che ha il compito di renderizzare il template Registrazione.html.

Nella sezione Home i 4 pulsanti che consentono la scelta della macrocategoria sono stati realizzati con delle ancore disabilitate e sono stati animati in modo tale da far cambiare il colore del bordo al passaggio del mouse attraverso l'utilizzo della pseudo-classe hover nel css.

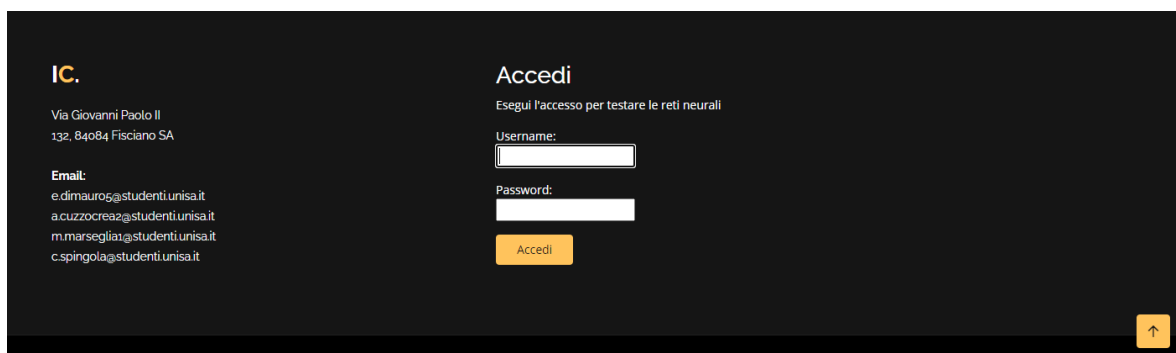
All'interno di questo template, per garantire una visualizzazione vincolata degli elementi determinata dall'autenticazione o meno degli utenti che la visitano, è stato utilizzato un tag che fornisce la struttura di controllo "if" e la classe models.User forniti da Django. In particolare, il tag si occupa di controllare l'attributo in sola lettura is_authenticated in modo da verificare se l'utente che sta visitando la pagina ha eseguito l'accesso. Nel caso in cui il valore dell'attributo è False l'utente visualizza gli elementi precedentemente descritti. Nel caso in cui, invece, il valore dell'attributo è True vengono apportate le seguenti modifiche: i pulsanti Accedi e Registrati vengono sostituiti rispettivamente da un'icona ed il pulsante Esci; nel footer non è più presente il form ed i 4 pulsanti nella sezione Home vengono abilitati.



Header prima dell'accesso



Header dopo l'accesso

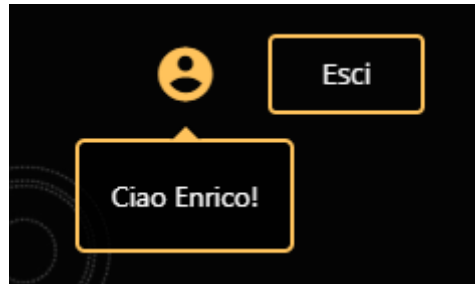


Footer prima dell'accesso



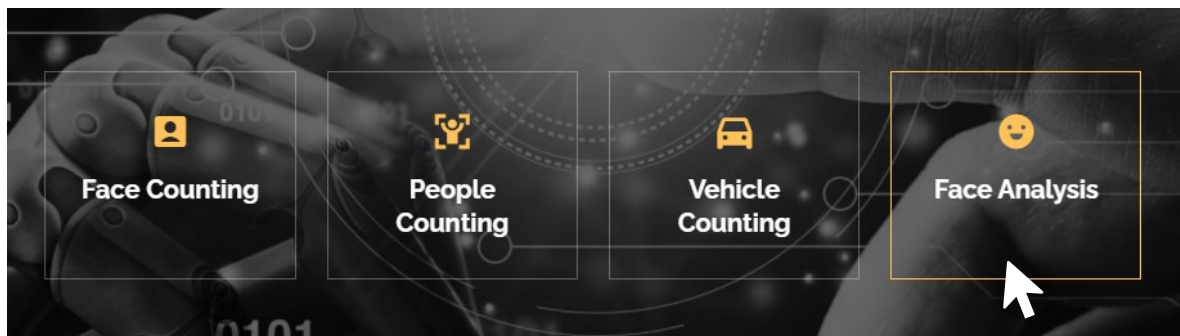
Footer dopo l'accesso

L'icona che sostituisce il pulsante Accedi, se cliccata, fa visualizzare all'utente un popover con un messaggio di saluto personalizzato con il nome utente.

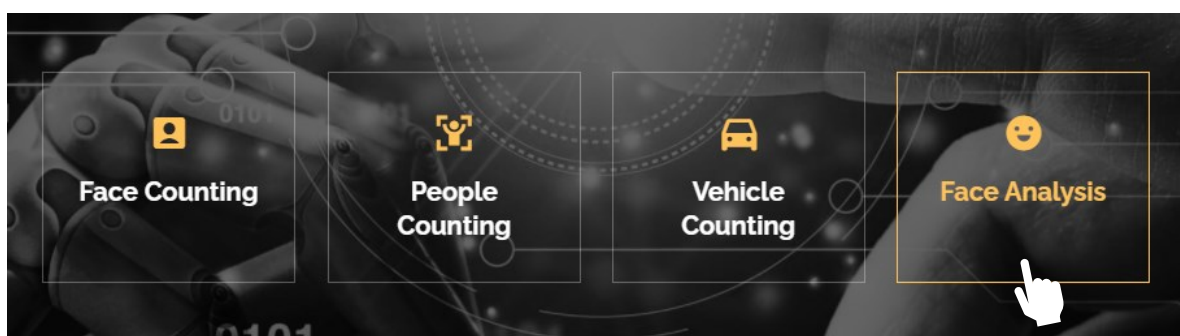


Popover personalizzato

Il pulsante Esci permette di far terminare la sessione dell'utente che lo clicca. I 4 pulsanti nella sezione Home acquisiscono l'attributo href contenente l'URL della view chiamata Test, con l'aggiunta della rispettiva macrocategoria sotto forma di query string, che ha il compito di renderizzare il template Test.html e delle animazioni che permettono di modificare il colore delle scritte tramite degli script javascript attuati dagli eventi "onmouseover" ed "onmouseout".



Pulsante prima dell'accesso



Pulsante dopo l'accesso

Per poter gestire la visualizzazione attraverso il controllo dell'autenticazione è stato utilizzato il template `login.html` che eredita `Homepage.html`. Questo nuovo template si occupa della visualizzazione del form per l'immissione delle credenziali da parte dell'utente e della gestione di tali dati.

Per poter garantire la corretta visualizzazione del template è stata utilizzata, nella prima istanza della funzione `path()` all'interno del file `urls.py` dell'applicazione, la classe `LoginView`.

Di default, tale classe se chiamata attraverso il metodo `GET` reindirizza al template `registration/login.html` al quale vengono passate alcune variabili:

- `form`: oggetto `Form` che rappresenta l'`AuthenticationForm`. La classe `Form`, fornita da Django, specifica i campi di un form, le label, il loro layout, le loro caratteristiche e permette di verificare la validità dei dati immessi
- `next`: URL al quale si viene reindirizzati in caso di corretto inserimento delle credenziali
- `site`: modello per memorizzare il dominio ed il nome degli attributi del sito web
- `site_name`: alias per il nome del valore assegnato a `site`

Per sostituire tale template con uno creato appositamente è possibile utilizzare la funzione `as_view()` assegnando al parametro `template_name` l'URL del template scelto. Esistono ulteriori parametri opzionali che possono essere passati a tale funzione, come, ad esempio, `extra_context` che permette di specificare un dizionario di dati utile nel template specificato.

Se la classe `LoginView` viene chiamata attraverso il metodo `POST`, ossia successivamente alla compilazione del form di autenticazione, vengono verificate le credenziali immesse dall'utente ed in caso di corretto inserimento avviene il reindirizzamento all'URL specificato nella variabile `next`. Se le credenziali sono sbagliate, invece, possono essere applicate procedure relative all'errore e si viene reindirizzati alla stessa pagina raggiunta attraverso il metodo `GET`.

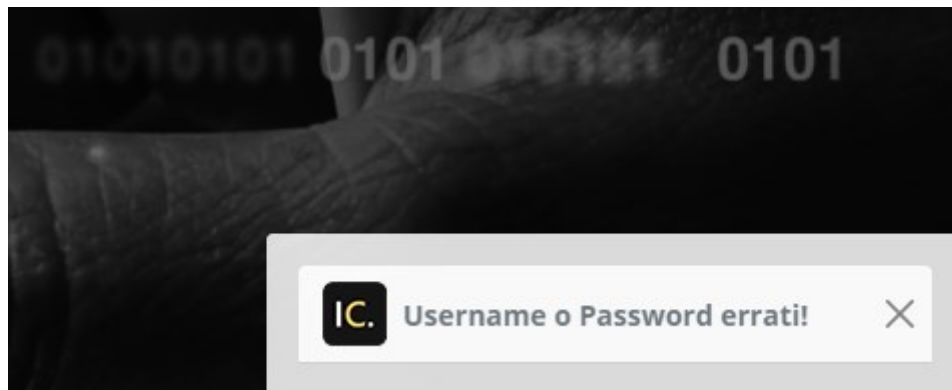
Nel caso in cui la variabile `next` sia vuota viene utilizzato l'URL specificato all'interno della variabile `LOGIN_REDIRECT_URL` presente nel file `settings.py`.

Per la realtà di interesse, siccome è stato utilizzato un template differente da quello di default, è stata utilizzata la funzione `as_view()` ed è stato inserito al suo interno il parametro `template_name` valorizzato con la stringa `"ImageClassifier/login.html"`. Al parametro `root` della funzione `path()` è stato assegnato il valore `"Homepage"`. In questo modo ogni richiesta relativa alla Homepage produce la visualizzazione del template `login.html`.

All'interno della funzione `as_view()` è stato definito, inoltre, il parametro `extra_context` con lo scopo di passare al template un dizionario costituito da 3 liste definite sempre all'interno del file `urls.py`. Tali liste sono state popolate attraverso i dati restituiti da query eseguite sui modelli `Macrocategoria`, `Appartenenza` e `Rete_Neurale` presenti nel database e che rappresentano le informazioni necessarie per implementare una visualizzazione dinamica di alcuni elementi del template `Homepage.html`. Più precisamente tali elementi dinamici sono i nomi e le descrizioni delle macrocategorie e delle reti neurali ed il numero di immagini classificate per ogni macrocategoria.

Il template `login.html` è costituito da un blocco che, per l'ereditarietà, verrà inserito in maniera condizionata all'interno del template `Homepage.html`. Il blocco contiene il codice che permette la visualizzazione e la compilazione di un form costituito dai campi `username` e `password` e da un pulsante di invio denominato `Accedi`. Le label e gli input dei campi del form sono stati gestiti attraverso l'oggetto `Form` passato al template. L'attributo `action` del form è stato valorizzato con una stringa vuota in quanto nel file `settings.py` è stata specificata la variabile `LOGIN_REDIRECT_URL` valorizzata con l'URL relativo alla Homepage. Nel blocco viene gestito anche l'eventuale errore restituito nel caso in cui l'accesso non vada a buon fine.

In tal caso, infatti, l'utente viene riportato alla Homepage come utente non loggato e visualizza, in aggiunta, un toast contenente un messaggio di errore.



L'attributo href del pulsante Esci, visualizzato da un utente loggato, reindirizza all'URL “/ImageClassifier/logout”. Tale URL è stato specificato come parametro root della seconda istanza della funzione path() all'interno di urls.py. Il parametro view, invece, è il risultato della funzione as_view() della classe LogoutView.

Tale classe si occupa di terminare la sessione dell'utente che la utilizza. Di default, il template al quale si viene reindirizzati è registration/logged_out.html. Per sostituire tale template è possibile agire in modo simile a quanto descritto precedentemente per il caso della classe LoginView quando viene chiamata attraverso il metodo POST. In questo caso, infatti, è possibile specificare un URL all'interno della variabile LOGOUT_REDIRECT_URL presente nel file settings.py.

Nella realtà di interesse è stata valorizzata tale variabile con l'URL relativo alla Homepage.



Sezione Home

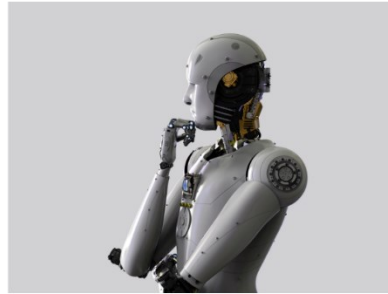
INFO SCOPR(IC.)I

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

Sezione Info

RETI NEURALI SCOPRI LE NOSTRE RETI NEURALI

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

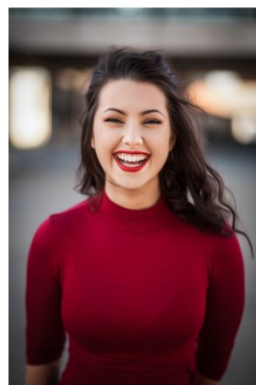
[Scopri Di Più](#)

Sezione Reti Neurali

PORTFOLIO

SCOPRI IL NOSTRO PORTFOLIO

TUTTE VOLTI PERSONE VEICOLI



Un po' di numeri...

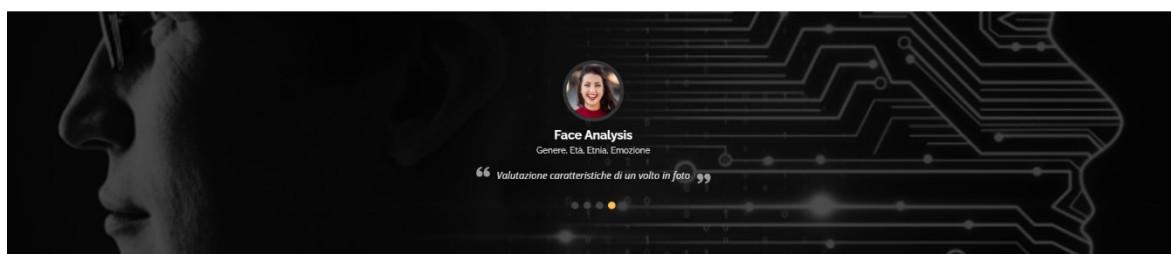
...grazie al contributo dei nostri utenti le reti neurali sono state testate, per ogni macrocategoria, sul seguente numero di immagini:

15
Face Counting

20
People Counting

25
Vehicle Counting

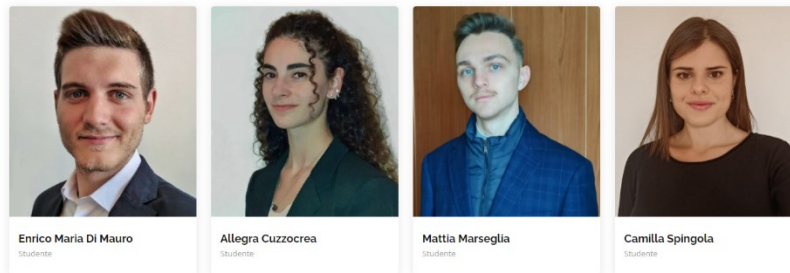
30
Face Analysis



Sezione Portfolio

TEAM

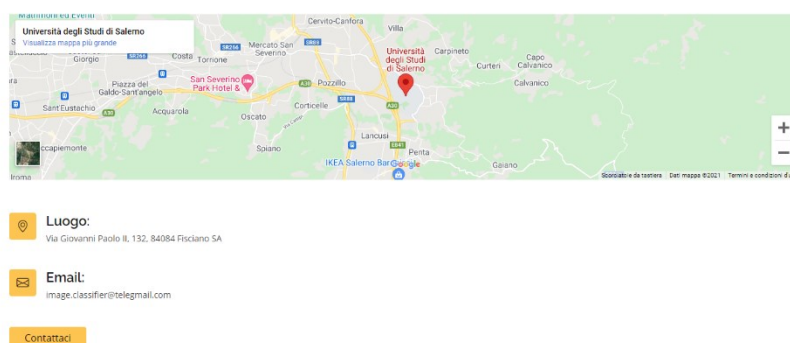
SCOPRI IL NOSTRO TEAM



Sezione Team

CONTATTI

CONTATTACI

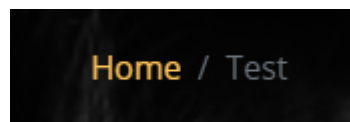


Sezione Contatti

3.2. TEST

Dalle descrizioni della progettazione delle pagine di Test e Result presenti nel capitolo 2 si evince che esse presentano gli stessi header e footer. Per questo motivo, è stato ritenuto opportuno inserire il codice in comune in un template denominato common.html che viene esteso dai template Test.html e Result.html. In questo modo, quando verrà eseguito il rendering del template Test.html, quest'ultimo verrà inserito in common.html permettendo la visualizzazione di tutti gli elementi implementati. La stessa cosa accade per il template Result.html.

In common.html è stata implementata una breadcrumb nell'header utilizzando una struttura fornita dalla Documentazione di Bootstrap.



Breadcrumb della pagina Test

È stata, inoltre, impostata l'immagine di sfondo, attraverso la classe "sfondo_test", e modificata attraverso alcuni comandi CSS eseguiti sulla classe "sfondo".

Come deciso in fase di progettazione, il form è stato incluso in una card al centro della pagina ed è stato creato un riquadro destinato a contenere l'immagine inserita dall'utente.

Per implementare il form, è stato creato un nuovo file Python nella cartella dell'applicazione, denominato forms.py. In questo modo è possibile avere più moduli per uno stesso form.

Per ogni campo da visualizzare è stato creato un modulo, rappresentato da una classe, che eredita dalla classe `ModelForm` di Django.

Quest'ultima è una sottoclasse della classe `Form` e permette di creare una classe `Form` da un modello Django. In questo modo, il modulo viene costruito sulla base dei campi del modello esistente.

All'interno della classe creata è stata inserita una meta-classe contenente due meta-informazioni: il modello sul quale si basa il modulo ed i campi a cui si è interessati.

Per assegnare caratteristiche ai campi è possibile utilizzare l'attributo `widgets` della classe `Meta`. Tale attributo è un dizionario che associa ai nomi dei campi il tipo di input da utilizzare con i relativi attributi.

Se tali caratteristiche sono ottenute dal risultato di query, invece, è necessario sovrascrivere il metodo speciale `__init__()` ed utilizzare il parametro `self`, che rappresenta l'istanza dell'oggetto a cui si fa riferimento. Inoltre, i parametri passati da una view possono essere utilizzati in un modulo inserendoli, a partire dal secondo argomento, nella definizione di `__init__()`.

In particolare, nella pagina destinata al test della macrocategoria gestita, ossia *Face Analysis*, i campi da visualizzare sono:

- inserimento dell'immagine.
- valorizzazione delle categorie Età, Genere, Etnia ed Emozione.
- scelta della Rete Neurale.
- impostazione del Threshold.

Il modulo finalizzato all'inserimento dell'immagine è costruito sul modello *Immagine* e possiede `image` come campo da visualizzare attraverso un `FileInput`. Su quest'ultimo è stato gestito l'evento `onchange` mediante una funzione Javascript denominata `readURL()`. Tale funzione si occupa di far visualizzare l'immagine caricata all'interno riquadro predisposto. In seguito, è stata realizzata una label con le caratteristiche di un pulsante, conforme allo stile delle pagine del portale, per sostituire l'input prodotto dal `FileInput`.

Per la modularità, l'implementazione dei campi del form prevede un meccanismo di gestione di qualsiasi categoria che debba essere valorizzata attraverso una scelta tra più opzioni. Tale meccanismo presuppone la creazione di relazioni contenenti i valori tra cui effettuare la scelta. All'interno del database tali relazioni devono essere denominate come le categorie che rappresentano. Esempi di categorie implementate in questo modo sono *Genere*, *Etnia* ed *Emozione*. Il modulo finalizzato alla valorizzazione della categoria *Età* è costruito sul modello *Valorizzazione* e possiede `valore_v` come campo da visualizzare attraverso un `NumberInput`. La numerazione è stata impostata tra 0 ed il valore del campo `int_max` associato alla categoria.

Il modulo finalizzato alla valorizzazione della categoria *Genere* è costruito sul modello *Valorizzazione* e possiede `valore_v` come campo da visualizzare attraverso un `RadioSelect`. Le scelte sono state prelevate dalla rispettiva relazione presente nel database.

Il modulo finalizzato alla valorizzazione delle categorie *Etnia* ed *Emozione* è costruito sul modello *Valorizzazione* e possiede `valore_v` come campo da visualizzare attraverso un `ModelChoiceField`. Le scelte sono state prelevate dalle rispettive relazioni presenti nel database.

Il nome della categoria da cui prelevare le scelte per ogni modulo è stato passato nella definizione del metodo `__init__()`.

Il modulo finalizzato alla selezione della Rete Neurale è costruito sul modello *Classificazione* e possiede `rete_neurale` e `threshold` come campi da visualizzare rispettivamente attraverso un `ModelChoiceField` ed un `NumberInput`. I nomi delle reti neurali disponibili per la macrocategoria scelta, il cui nome è stato passato nella definizione del metodo `__init__()`, sono state prelevate dalla relazione *Rete Neurale* presente nel database.

La view Test definita per gestire la visualizzazione e la valutazione del form esegue istruzioni diverse in base al tipo di richiesta con cui viene invocata:

- GET: la view preleva dall'URL la macrocategoria inserita attraverso la query string presente nell'ancora del pulsante della Homepage. Viene renderizzato il template Test.html passando come parametro context della funzione render() i campi del form da visualizzare. I nomi delle categorie associate alla macrocategoria scelta vengono ottenuti tramite l'esecuzione di una query sulla relazione Macrocategoria presente nel database.
Se nell'URL non è presente una query string, la view reindirizza alla Homepage.
- POST: la view preleva i dati immessi dall'utente, inizializza ed invoca la rete neurale selezionata ed inserisce nel database le tuple opportune. Infine, renderizza il template result.html passando alla funzione render() un dizionario costituito dalle variabili necessarie per la visualizzazione del risultato del test.

Nel template Test.html sono stati inseriti i tag HTML per la creazione del form e le variabili Python necessarie alla visualizzazione dei moduli creati nel file forms.py.

Il meccanismo dei tasti che condiziona la visualizzazione dei riquadri contenenti i campi del form è stato realizzato attraverso l'utilizzo di ancore. Queste ultime sono caratterizzate dall'attributo data-bs-toggle, a cui è stato assegnato il valore "collapse", e dall'attributo href valorizzato con l'id del div corrispondente al riquadro associato.

Per rispettare la scelta compiuta durante la fase di progettazione, è stata utilizzata all'interno del riquadro contenente le categorie da valorizzare la proprietà CSS overflow-y alla quale è stato assegnato il valore auto. Ciò permette di ottenere una scroll-bar nel momento in cui gli elementi presenti all'interno del riquadro superino la dimensione verticale di quest'ultimo.

Infine, nel file urls.py dell'applicazione, è stata inserita un'istanza di path() che all'URL 'Test' invoca la vista Test.

IC.

Home / Test

FACE ANALYSIS

Carica Immagine

Classifica le categorie

Età: 0 - 120

Genere: ☐ M ☐ F

Firma:

Rete Neurale: --Seleziona--

Threshold: 0.4

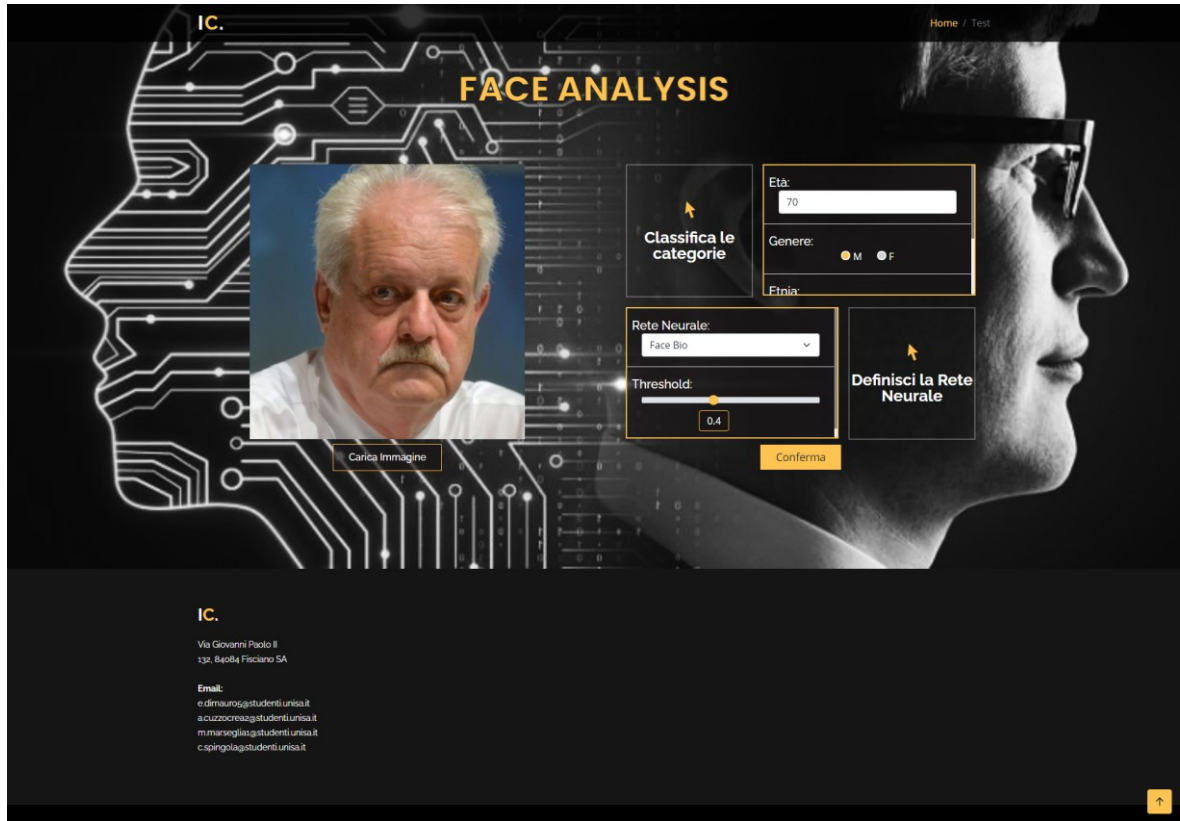
Definisci la Rete Neurale

Conferma

IC.

Via Giovanni Paolo II
132, 04014 Frosinone SA

Email:
e.dimauro@studenti.unisa.it
a.cuzzocrea@studenti.unisa.it
m.mariagrazia@studenti.unisa.it
c.spingola@studenti.unisa.it



Pagina Test dopo la valorizzazione

3.3. RESULT

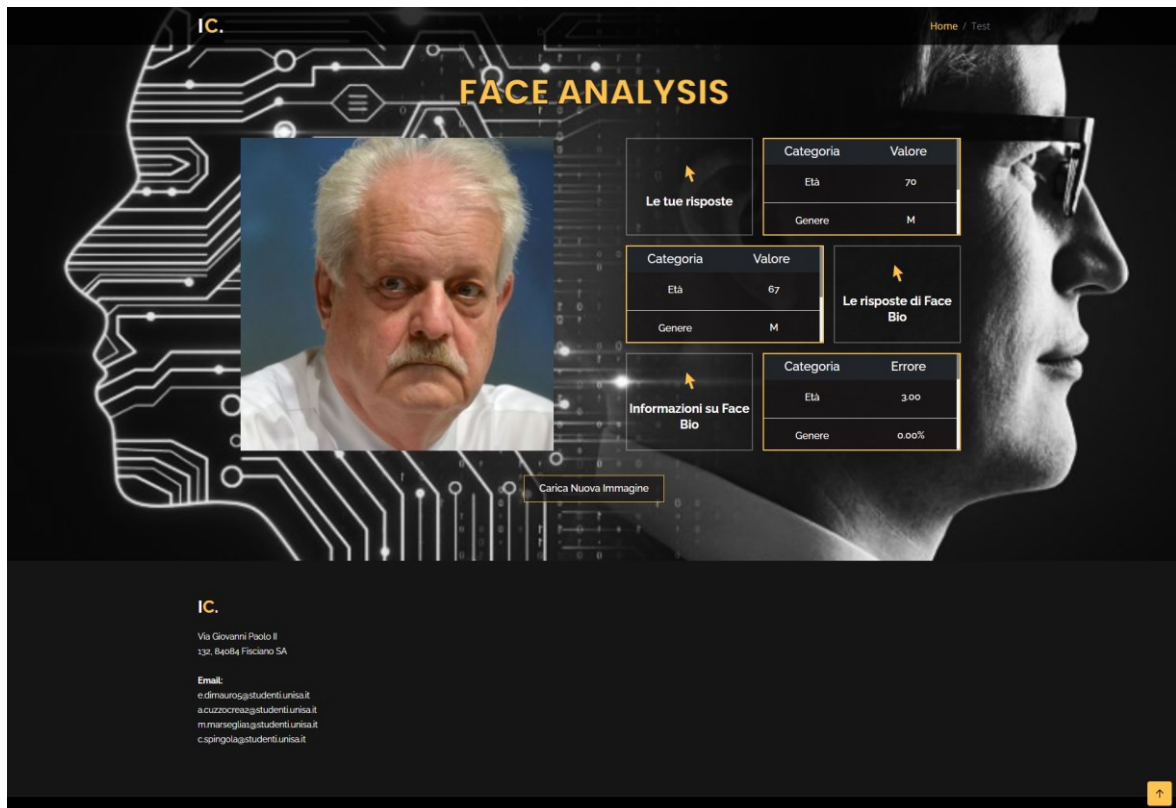
In accordo con quanto progettato nel capitolo 2, la pagina Result permette di visualizzare il risultato del test effettuato.

Come è stato esposto nella sezione sull'implementazione della pagina di test, l'Header ed il Footer sono contenuti nella pagina common.html.

L'implementazione grafica del template Result.html è stata realizzata inserendo alcuni elementi presenti nel template Test.html precedentemente realizzato.

Uno di questi è la scroll-bar disponibile nel momento in cui gli elementi presenti all'interno di uno dei riquadri superino la sua dimensione verticale.

I valori visualizzati nei riquadri sono stati inseriti nel template attraverso le variabili Python valorizzate, al momento del rendering, con i valori prelevati dal dizionario passato dalla view Test, descritta nella sezione precedente.



Pagina Result

3.4. REGISTRAZIONE

È stato creato un div per inserire il titolo “REGISTRAZIONE” ed il form. È stato inserito un bordo arrotondato agli angoli e di colore giallo assegnando dei valori alle proprietà border-color e border-radius del div all’interno del file CSS.

Per la creazione del form è stato utilizzato il file Python forms.py precedentemente creato nell’applicazione. È stata creata una classe dal nome AccountForm che eredita la classe UserCreationForm. Quest’ultima è un ModelForm per la creazione di un nuovo utente. Utilizzando tale classe, gli utenti registrati appariranno nella sezione Users presente nell’interfaccia di amministrazione di Django.

Di default, la classe contiene i campi: username, password1 e password2. Per realizzare il form progettato sono stati aggiunti i campi: first_name, last_name ed email del modello User.

Sovrascrivendo il metodo `__init__()`, ai campi sono stati assegnati gli appositi tipi di input e gli attributi. In particolare, ai campi password è stato assegnato il tipo PasswordInput che oscura i caratteri immessi dall’utente.

La view Registrazione definita nel file views.py renderizza il template Registrazione.html passando alla funzione render() un dizionario contenente il form da visualizzare. Inoltre, se tale vista viene invocata tramite una richiesta POST, vengono prelevati i dati immessi dall’utente e ne viene verificata la validità attraverso la funzione `is_valid()`. Nel caso in cui i dati siano corretti, il nuovo utente viene aggiunto nella sezione Users dell’interfaccia di amministrazione di Django e viene reindirizzato alla Homepage.

Nel template Registrazione.html sono stati inseriti i tag HTML per la creazione del form e i campi della variabile passata dalla relativa view utilizzando strutture di controllo contenute in tag Python.

Il gestore dell'evento onSubmit contenuto nel tag form invoca una funzione Javascript che effettua i controlli necessari per la correttezza dei dati.

Infine, nel file urls.py è stata inserita un'istanza della funzione path() che all'URL 'Registrazione' invoca l'apposita vista.

IC. Home / Registrazione

REGISTRAZIONE

Nome

Cognome

E-mail

Username

Password

La Password deve contenere:

- almeno 6 caratteri
- almeno una maiuscola

Conferma Password

IC.
Via Giovanni Paolo II
132, 84014 Fisciano SA

Email:
e.dimauro@studenti.unisa.it
a.cuzzocrea@studenti.unisa.it
m.marielga@studenti.unisa.it
c.spingola@studenti.unisa.it

↑

Pagina Registrazione

4. CAPITOLO 4

CONCLUSIONI E PROSPETTIVE FUTURE

Alla fine della progettazione e, successivamente, dell'implementazione di tutte le funzionalità descritte nei capitoli precedenti, è stato ottenuto un portale web che rispecchia gli obiettivi prefissati. L'attività è stata svolta in Team. Il mio contributo di sviluppo è stato a 360 gradi partecipando alla progettazione ed all'implementazione dell'intera attività. La mia attenzione è stata posta particolarmente sulla macrocategoria Face Analysis che permette di sfruttare la rete neurale Face Bio per analizzare un volto e determinarne l'età, il genere, l'etnia e l'emozione.

Il Team working è una soft skill indispensabile in sempre più ambiti lavorativi, che consente di valorizzare le proprie capacità mettendole a disposizione del gruppo, per massimizzare i risultati. Grazie ad attente valutazioni effettuate durante le fasi di realizzazione dell'attività di tesi, sono state individuate ed applicate strategie di progetto che hanno permesso l'ottenimento della modularità.

Il vantaggio dell'aver prodotto un codice modulare è quello di poter aggiungere, eliminare e modificare macrocategorie, categorie e reti neurali in maniera semplice, senza compromettere in alcun modo la struttura delle pagine.

L'esperienza di tesi svolta è stata molto formativa dal punto di vista professionale, perché ha permesso di raffinare le proprie conoscenze delle tecnologie software per il web, ossia i linguaggi HTML, CSS e JAVASCRIPT, e di basi dati; ha consentito, inoltre, di valutare le capacità di analisi e gestione dei problemi che possono presentarsi nello sviluppo di un progetto, come avviene nel mondo del lavoro ed ha reso possibile l'acquisizione e la padronanza, seppur in piccola parte, del linguaggio Python e del suo framework Django. Questi ultimi, infatti, sono stati utilizzati per la creazione e lo sviluppo di alcuni dei social network più popolari al mondo, come, ad esempio, YouTube ed Instagram. Python, inoltre, è uno dei linguaggi che viene utilizzato sempre più nel campo della programmazione ad alto livello come machine learning e data analysis.

Con tali presupposti, è facile intuire che la conoscenza di tali linguaggi possa essere un ottimo punto di partenza per il futuro in ambito professionale e lavorativo.

BIBLIOGRAFIA E SITOGRAFIA

Atzeni P., Ceri S., Fraternali P., Paraboschi S., Torlone R., *Basi di dati*, Milano, McGraw-Hill, (2018), V edizione

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

<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

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

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

<https://www.pgadmin.org/docs/>

https://it.wikipedia.org/wiki/Rete_neurale_artificiale

RINGRAZIAMENTI

I miei sinceri ringraziamenti vanno:

al Prof. Antonio Greco, mio Relatore, che in ogni momento ha supportato il mio lavoro e quello di tutto il Team con estrema disponibilità e validi suggerimenti teorici e pratici;

alla Prof.ssa Alessia Saggese, secondo Relatore, che mi ha validamente motivato ed incentivato nella realizzazione del progetto.

Alla mia collega nonché compagna di viaggio Allegra Cuzzocrea va un ringraziamento speciale per la collaborazione, il totale sostegno, la sua infinita disponibilità e la sua inesauribile fiducia.

Un grazie particolare ai colleghi del Team Camilla Spingola e Mattia Marseglia con i quali ho condiviso in maniera proficua idee ed intuizioni per dare forma al prodotto finale.

Un ringraziamento di cuore ai miei genitori, ai miei fratelli, a mia nonna ed i miei zii che mi hanno sempre incoraggiato e seguito durante tutto il percorso universitario.