

UNIVERSITÀ DEGLI STUDI DI SALERNO



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

Corso di Laurea in Ingegneria Informatica

ELABORATO DI TESI

Conteggio automatico di persone tramite una rete neurale integrata
in una web application

Docente Relatore

Prof. Antonio GRECO

Docente Correlatore

Prof.ssa Alessia SAGGESE

Candidato

Camilla SPINGOLA

Matricola

0612704363

Anno Accademico 2020/2021

*Possa ciascuno di voi, nonostante tutte le distrazioni generate dalla tecnologia
avere successo nel trasformare le informazioni in conoscenza,
la conoscenza in comprensione e la comprensione in saggezza*
(Edsger Wybe Dijkstra)

Sommario

<i>Introduzione</i>	<i>3</i>
1. Raccolta, Analisi e Specifica dei requisiti.....	5
1.1. Descrizione della realtà di interesse	5
1.2. Analisi dei requisiti	5
1.2.1. Glossario dei termini	6
1.2.2. Strutturazione in Frasi	6
1.2.3. Individuazione delle operazioni principali.....	7
1.3. Classificazione dei requisiti.....	7
1.3.1. Requisiti funzionali.....	8
1.3.2. Requisiti non funzionali.....	9
1.4. Casi d'uso	9
2. Progettazione	11
2.1. Progettazione dei dati	11
2.1.1. Progettazione Concettuale	11
2.1.2. Progettazione Logica	17
2.1.3. Normalizzazione	23
2.2. Progettazione delle applicazioni	24
2.2.1. Caratteristiche comuni individuate	24
2.2.2. Homepage.....	24
2.2.3. Registrazione	26
2.2.4. Test	26
2.2.5. Result.....	27
3. Implementazione.....	28
3.1. Implementazione del back-end	28
3.1.1. Implementazione della base di dati.....	28
3.1.2. Implementazione dell'applicazione	32
3.2. Implementazione del front-end.....	37
3.2.1. Caratteristiche comuni tra le interfacce	38
3.2.2. Implementazione della Homepage.....	38
3.2.3. Implementazione della pagina di Registrazione	39
3.2.4. Implementazione della pagina per effettuare il test e visualizzare il risultato	40
4. Integrazione dei vari moduli, Visualizzazione e Descrizione del risultato finale	41
4.1. Visualizzazione e descrizione del risultato.....	41
4.1.1. Visualizzazione Homepage	41
4.1.2. Visualizzazione pagina di Registrazione	47
4.1.3. Visualizzazione pagina di Test	49
4.1.4. Visualizzazione pagina di Result.....	50
Conclusioni.....	51
Bibliografia e Sitografia.....	52

Introduzione

Il contributo che il presente lavoro intende offrire al panorama delle web application risulta originale per l'utilizzo delle reti neurali integrate nel sistema, al fine di effettuare la classificazione di immagini.

La pervasività delle immagini nell'interazione tecnologica, ampliando la platea degli utenti anche non addetti ai lavori, coinvolge le reti neurali in un'interazione uomo-macchina sempre più consapevole e raffinata. Infatti, se l'approccio iniziale dell'utente alle innovazioni si limita spesso ad uno scopo ludico o strettamente finalizzato alla ricerca di un risultato, la rivendicazione di una fruizione più consapevole si manifesta come esigenza in molteplici settori, determinando una ricaduta positiva sulla crescita esponenziale delle tecnologie sottese.

Si moltiplicano di conseguenza le ricerche che mirano all'efficienza e all'implementazione degli strumenti, sempre in un'ottica che garantisca un'interazione piacevole, rapida e immediatamente spendibile. Tale esigenza impone agli esperti del settore un aggiornamento costante delle tecnologie software ricorrendo ai contributi dell'intelligenza artificiale e raffinandone costantemente le performance.

In quest'ottica il portale web realizzato classifica attraverso una serie di reti neurali aspetti specifici delle immagini organizzate in macrocategorie, come il numero di persone presenti al loro interno. Il processo ovviamente consente all'utente di agire da protagonista, soddisfacendo la sua naturale tendenza all'autonomia, ma al contempo fornisce una comparazione puntuale con le reti neurali selezionate. L'immagine-stimolo ottiene, quindi, un'analisi molteplice e varia proveniente da una pluralità di angolazioni e il contributo delle reti neurali, fungendo da potenziatore dell'apparato sensoriale, soddisfa il desiderio di guardare oltre ciò che appare.

Navigando nel portale, l'utente, dopo aver effettuato l'autenticazione, procede alla scelta della macrocategoria (Face Counting, People Counting, Vehicle Counting e Face Analysis), e dell'immagine ad essa associata, da sottoporre ad analisi, avviando un'attività di selezione tra le reti neurali segnalate come compatibili. I valori impostati dall'utente per le varie categorie saranno, al termine del processo, comparati con quelli suggeriti dalla rete neurale, fornendo un aggiornamento dei margini di insuccesso caratteristici della rete neurale per ogni specifica categoria classificata. Ovviamente l'operazione può essere reiterata un numero illimitato di volte per ciascun utente.

La validazione del processo e del prodotto adottati dipendono sostanzialmente da due ragioni. La prima più utilitaristica, in quanto se si supera la soglia dell'impiego ludico da parte dei non addetti ai lavori, si ottiene uno strumento adeguato a testare l'efficacia e l'attendibilità delle reti neurali, monitorando la precisione dei risultati ottenuti in rapporto al valore di threshold selezionato. L'altro aspetto che rende attuale il prodotto è legato all'uso di tecnologie software innovative e diffuse e alle potenzialità che il portale offre per riflettere e condividere i contributi dell'intelligenza artificiale allo scenario tecnico contemporaneo.

Struttura della tesi

Il *Capitolo 1* descrive la fase di raccolta e analisi dei requisiti funzionali e non funzionali del sistema da realizzare, in modo da evitare errori che impediscano la conformità alle richieste iniziali. A tale scopo si è cercato di risolvere eventuali ambiguità associate alla realtà di interesse descritta in

linguaggio naturale e di calibrare la progettazione in relazione ad una potenziale utenza destinataria del prodotto.

La fase di definizione e di progettazione, affrontata nel *Capitolo 2*, ha avuto come scopo la definizione dell'architettura del sistema, tramite l'individuazione dei componenti necessari e delle loro caratteristiche. L'articolazione si snoda in due ambiti: una progettazione dei dati e una progettazione delle applicazioni. L'obiettivo della prima è stato individuare la struttura che i dati dovessero avere, quello della seconda è stato invece definire le caratteristiche del programma applicativo.

L'implementazione del sistema, descritta all'interno del *Capitolo 3*, è suddivisa in rapporto alla componente front-end e alla componente back-end. Per ciascuna fase sono stati descritti gli applicativi software utilizzati, avendo cura di motivare le logiche sottese alle scelte effettuate.

La fase conclusiva del processo, riportata nel *Capitolo 4*, ha previsto l'integrazione dei vari moduli realizzati allo scopo di validare e descrivere il prodotto finale, anche ai fini di programmare eventuali implementazioni future.

1. Raccolta, Analisi e Specifica dei requisiti

Le attività di raccolta e di analisi accurata dei requisiti del sistema software in questione sono state fondamentali al fine di porre delle solide basi per la fase progettuale. Tale fase di gestione dei requisiti è indubbiamente tra le più critiche, in quanto le scelte e gli eventuali errori commessi durante tale fase inevitabilmente possono influire negativamente sulla progettazione, rischiando la realizzazione di un sistema non conforme alle richieste iniziali, e dunque non utilizzabile. Fondamentale per un corretto svolgimento di tali passaggi risulta essere l'individuazione dei requisiti, ovvero le descrizioni dei servizi di sistema e i vari vincoli da rispettare, al fine di individuare cosa il sistema software debba fare, tralasciando, in questa prima fase di analisi, le modalità di svolgimento di tali azioni. L'attività di raccolta dei requisiti si pone l'obiettivo di individuare in maniera completa i problemi che l'applicazione deve risolvere e le caratteristiche che tale applicazione dovrà avere. Le caratteristiche del sistema riguardano sia gli aspetti statici e dunque i dati, sia gli aspetti dinamici e dunque le operazioni su questi ultimi.

1.1. Descrizione della realtà di interesse

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

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

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

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

1.2. Analisi dei requisiti

Per ovviare a possibili fraintendimenti e ambiguità derivanti dalle descrizioni effettuate in linguaggio naturale che spesso caratterizzano le specifiche dei requisiti, è stato fondamentale effettuare una

profonda analisi del testo contenente le specifiche, al fine di filtrare gli eventuali termini ambigui presenti, raffinando concetti poco chiari e cercando di riportare tutte le specifiche ad un uguale livello di astrazione.

1.2.1. Glossario dei termini

Per il raggiungimento degli obbiettivi sopra proposti è stato estremamente utile definire un glossario che, per ciascuno dei termini principali individuati all'interno delle specifiche, prevedesse una breve descrizione, possibili sinonimi ed eventuali altri termini, anch'essi appartenenti al glossario, con i quali esistesse una relazione.

Termine	Descrizione	Sinonimi	Collegamenti
Immagine	Fotografia virtuale scelta dall'utente per sottoporla a classificazione	-	Macrocategoria, Classificazione
Macrocategoria	Categoria molto ampia che può raggruppare diverse categorie più specifiche.	-	Categoria, Immagine, Rete neurale
Categoria	Elemento costitutivo di una macrocategoria.	-	Macrocategoria, classificazione
Classificazione	Operazione di specificazione delle caratteristiche di un'immagine effettuata da una rete neurale	Tentativo	Rete neurale, Immagine, Categoria
Rete neurale	Modello computazionale che classifica immagini di una determinata macrocategoria	-	Macrocategoria, Classificazione

Tabella 1. Glossario dei termini

1.2.2. Strutturazione in Frasi

Al fine di riportare tutte le specifiche ad un uguale livello di astrazione, evitando termini troppo generici o troppo specifici che rendono i concetti poco chiari e al fine di ottenere delle frasi con una struttura standard, dunque non contorte ma semplici, chiare e con il medesimo stile sintattico, si è proceduto a realizzare una strutturazione in frasi delle proprietà di interesse per i concetti principali individuati.

- **Frasi di carattere generale:**

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

- **Frasi relative a immagine**

Per l'immagine si vuole rappresentare l'URL, la valorizzazione delle categorie associate effettuata dall'utente.

- **Frase relative a macrocategoria**

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

- **Frase relative a categoria**

Relativamente alla categoria si vuole rappresentare il nome e la macrocategoria di appartenenza.

- **Frase relative a classificazione**

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

- **Frase relative a rete neurale**

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

1.2.3. Individuazione delle operazioni principali

La fase di individuazione delle operazioni principali ha avuto come obiettivo individuare una serie di operazioni che avrebbero potuto essere svolte sui dati effettuando anche una stima riguardo alla frequenza di svolgimento, dato fondamentale nella fase di progettazione logica della base di dati.

- **Operazione 1:** Stampa i valori delle prestazioni di una rete neurale (in media 20 volte al giorno)
- **Operazione 2:** Aggiornamento del valore della prestazione di una rete neurale (in media 40 volte al giorno)
- **Operazione 3:** Caricamento di una nuova immagine e valorizzazione delle categorie annesse (in media 30 volte al giorno)
- **Operazione 4:** Selezione di una rete neurale (in media 30 volte al giorno)
- **Operazione 5:** Stampa per ciascuna macrocategoria le categorie e le reti neurali associate (in media 75 volte al giorno)
- **Operazione 6:** Stampa il risultato della classificazione effettuata da una rete neurale su una specifica immagine (in media 10 volte al giorno)
- **Operazione 7:** Registrazione di un nuovo utente (2 volte al giorno)

1.3. Classificazione dei requisiti

I requisiti possono genericamente essere raggruppati e classificati secondo molteplici criteri, ovvero il tipo di caratteristica che descrivono, la sorgente del requisito o l'oggetto a cui sono rivolti. In tale fase si è proceduto a una classificazione relativa al tipo di caratteristica descritta, distinguendo fra loro i requisiti funzionali da quelli non funzionali.

1.3.1. Requisiti funzionali

I requisiti funzionali mirano alla descrizione delle funzioni che un sistema deve implementare ed eseguire. Tali requisiti provvedono dunque a descrivere una funzionalità che deve essere implementata, ma non le modalità. Sono stati utili al fine di definire quale dovesse essere il comportamento del sistema a fronte di particolari input e come esso dovesse reagire in determinate situazioni. Ci si è assicurati che tali requisiti fossero coerenti, completi e che non generassero situazioni di inconsistenza causate specifiche tra loro in conflitto.

Si riportano di seguito i requisiti individuati.

- **Caricamento di un'immagine:**
Il portale web deve consentire ad un utente registrato e autenticato di poter caricare, per un numero illimitato di volte, un'immagine a sua scelta, per poter effettuare la classificazione.
- **Scelta di una macrocategoria:**
Il portale web deve consentire all'utente registrato e autenticato di poter scegliere tra le macrocategorie disponibili, quella su cui effettuare il test.
- **Valorizzazione di un'immagine:**
Il portale deve consentire all'utente registrato e autenticato, in fase di test, dopo aver scelto una macrocategoria e aver caricato per quest'ultima una foto, di poter scegliere, per ciascuna categoria associata alla macrocategoria selezionata, i valori descrittivi dell'immagine.
- **Scelta di una rete neurale:**
Il portale web deve consentire all'utente registrato e autenticato di poter, caricata l'immagine, scegliere la rete neurale, tra quelle disponibili, adatte alla macrocategoria in questione, alla quale far effettuare la classificazione, chiarendo anche il threshold da utilizzare.
- **Visualizzazione della classificazione effettuata dalla rete neurale:**
Il portale deve consentire all'utente che ha caricato e valorizzato l'immagine di poter visualizzare i risultati restituiti dalla rete neurale nel corso della classificazione effettuata sulla medesima immagine.
- **Visualizzazione delle prestazioni di una rete neurale:**
Il portale deve consentire la consultazione delle prestazioni di ciascuna rete neurale, per le categorie su cui effettua classificazioni, sul numero di immagini classificate.
- **Salvataggio dei dati del Test:**
Il portale deve garantire che, ogni qualvolta venga eseguito un caricamento, la corrispondente valorizzazione delle categorie da parte dell'utente e la classificazione da parte della rete neurale selezionata, tali valori siano memorizzati all'interno del database in maniera opportuna.
- **Tracciamento degli utenti:**
Il portale deve consentire l'esecuzione del test al solo utente autenticato, deve dunque prevedere un meccanismo di gestione degli utenti, assicurando che la visualizzazione e le funzionalità siano condizionate e soprattutto deve prevedere dei meccanismi per la gestione e per l'autenticazione dell'utente.
- **Gestione utenti:**
Il portale deve consentire all'amministratore di poter gestire gli utenti.

1.3.2. Requisiti non funzionali

I requisiti non funzionali descrivono dei vincoli che devono essere rispettati o delle caratteristiche e dei livelli di qualità che devono essere garantiti, non è però importante il modo in cui tali vincoli o tali caratteristiche siano realizzati. Si è individuata la necessità di soddisfare alcuni vincoli in maniera tale da rendere il portale utile per gli scopi prefissati, tali requisiti infatti sono spesso estremamente critici e vincolanti per l'utilizzo del sistema.

Si riportano di seguito i requisiti non funzionali su cui è stata posta attenzione:

- **Manutenibilità:**

Il portale deve essere implementato in modo tale da essere manutenibile, garantendo modularità e facilità di comprensione. A tal fine è necessaria una suddivisione delle operazioni da svolgere in aree funzionali. La realizzazione di un software modulare, porta infatti con sé l'enorme vantaggio di essere facilmente manutenibile, ovvero qualora venisse identificata una correzione o una modifica da apportare sarebbe più facile individuare la porzione di codice interessata, unicamente coinvolta nella correzione. La facilità di comprensione del software può invece essere assicurata tramite commenti e documentazioni da corredare al codice.

- **Scalabilità:**

Requisito fondamentale per il portale è la scalabilità, anch'essa derivante dalla manutenibilità del software. Qualora il requisito di scalabilità fosse soddisfatto potrebbero facilmente essere aggiunte funzionalità e requisiti ulteriori a posteriori, senza grandi stravolgimenti del codice, se non per le porzioni aggiuntive.

- **Usabilità:**

Requisito di assoluta importanza è l'usabilità dell'applicativo. A tale scopo è importante che l'interfaccia sia ben progettata e organizzata, in modo da ordinare le varie operazioni che possono essere svolte.

1.4. Casi d'uso

Si è proceduto all'individuazione dei casi d'uso, ovvero delle interazioni tra utente e sistema che consentono all'utente di raggiungere un obiettivo o di svolgere un compito, procedendo attraverso una descrizione di tipo testuale. Si riportano di seguito alcuni dei casi d'uso analizzati.

- **Svolgimento di un test**

1. L'utente seleziona una macrocategoria.
2. Il sistema visualizza la schermata per effettuare il test.
3. L'utente inserisce le informazioni richieste dalla schermata (carica un'immagine, inserisce i valori richiesti per le categorie, seleziona la rete neurale, seleziona il valore di threshold).
4. L'utente clicca sul bottone conferma.
5. Il sistema salva nel database le informazioni inserite e mostra all'utente il risultato della classificazione, oppure notifica all'utente la mancata compilazione di alcuni campi.

- **Registrazione utente**

1. L'utente seleziona la voce "Registrati".
2. Il sistema visualizza la schermata di Registrazione.
3. L'utente inserisce le credenziali richieste (nome, cognome, username, e-mail, password e riconferma della password).
4. L'utente clicca sul pulsante "Registrati".
5. Il sistema salva nel database le informazioni inserite e consente all'utente di effettuare il login, oppure in caso di fallimento notifica all'utente eventuali problematiche da correggere.

2. Progettazione

La fase di definizione e di progettazione, affrontata a seguito dell'individuazione dei requisiti, ha avuto come scopo la definizione dell'architettura del sistema. Tale fase di progettazione ha riguardato la descrizione dell'architettura software che si voleva realizzare, la divisione e la progettazione dettagliata delle componenti client e server e la descrizione delle varie componenti software, delle interfacce utilizzate e delle relazioni tra esse definite. Si è proceduto dunque all'individuazione dei componenti necessari e delle loro caratteristiche. La progettazione si è articolata in una progettazione dei dati e in una progettazione delle applicazioni. L'obiettivo della prima è stato individuare la struttura che i dati dovessero avere, quello della seconda è stato invece definire le caratteristiche del programma applicativo. Queste due attività hanno avuto uno sviluppo in cascata, seppure la loro complementarità sia stata imprescindibile.

2.1. Progettazione dei dati

Così come generalmente accade nei sistemi informativi, comprendenti tipicamente anche i programmi applicativi, le interfacce con l'utente e gli altri programmi di servizio, anche in questo caso, la base di dati costituiva solo una delle componenti dell'intero sistema. Il ruolo centrale che i dati avevano nel sistema di interesse, però, giustifica la scelta di predisporre una progettazione autonoma per la base di dati. Proprio per questo motivo, anche in accordo con l'approccio comune allo sviluppo dei sistemi informativi basato sui dati, l'attenzione è stata focalizzata in prima battuta sulla progettazione della base di dati e solo successivamente sulla realizzazione delle applicazioni che dovessero utilizzarla.

Individuate le proprietà e le funzionalità richieste al prodotto, tramite un'attività di raccolta e di analisi dei requisiti, si è proceduto alla fase di progettazione della base di dati, tramite il supporto di una metodologia di progettazione. La metodologia di progettazione utilizzata è una metodologia che nell'ambito delle basi di dati si è consolidata negli anni, essa ha previsto una decomposizione dell'intera attività progettuale in passi fra loro successivi e indipendenti, ciascuno di essi caratterizzato da strategie, da criteri da cui farsi guidare per effettuare una scelta in caso di più alternative e da alcuni modelli di riferimento a cui ispirarsi. Le fasi principali eseguite in cascata rispondono al criterio di separazione tra le decisioni relative a "cosa" rappresentare in una base di dati, da quelle relative a come farlo. Il primo aspetto è stato curato durante la fase di progettazione concettuale, il secondo, invece, durante la fase di progettazione logica. Questo secondo aspetto generalmente viene curato anche durante la fase di progettazione fisica, che però in questa attività realizzativa non è stata affrontata. L'applicazione di tale metodologia di progettazione ha permesso di ottenere un prodotto che fosse di qualità in termini di correttezza, completezza ed efficienza.

2.1.1. Progettazione Concettuale

La progettazione concettuale si è posta come obiettivo la rappresentazione delle specifiche informali della realtà di interesse in termini di una descrizione formale e completa, indipendente però dai criteri di rappresentazione utilizzati nel sistema di gestione di basi di dati. Durante tale fase il

modello concettuale utilizzato è stato il modello Entità-Relazione, ormai standard di riferimento nelle metodologie di progetto di basi di dati.

Schema concettuale

La strategia di progetto utilizzata durante la fase di progettazione dello schema E-R è stata una strategia mista, che combinasse al suo interno i benefici della strategia top-down e quelli della strategia bottom-up. Dunque, i concetti più rilevanti sono stati rappresentati all'interno di uno schema scheletro, in modo da avere una visione unitaria, sia pure astratta, dell'intero progetto. A partire dallo schema scheletro è poi stata effettuata una decomposizione dei requisiti e, per tutti i sottoschemi ottenuti, si è proceduto secondo una metodologia bottom-up e inside-out al raffinamento dei concetti presenti, in base alle loro specifiche e all'aggiunta di nuovi concetti che chiarissero specifiche non ancora trattate. L'ultimo step ha previsto l'integrazione dei vari sottoschemi in uno schema generale.

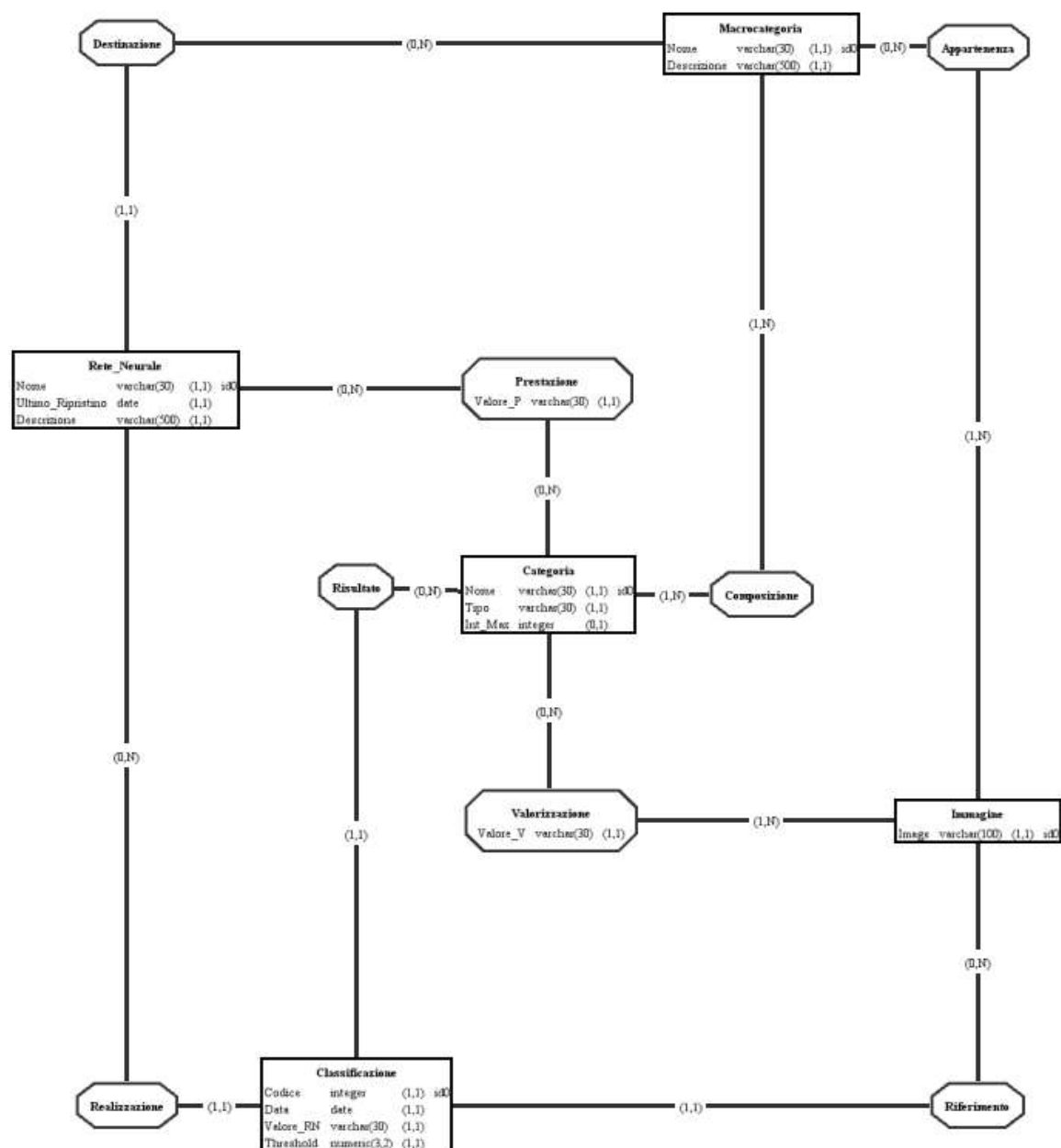


Figura 1. Schema E-R

È evidente, come solitamente accade, che per le specifiche inizialmente individuate le rappresentazioni potessero essere molteplici e tra loro non comparabili. In ogni caso, durante in fase di progettazione concettuale, si sono seguite le “regole concettuali” del modello E-R. Dunque, si sono rappresentate come entità tutti quei concetti che avessero proprietà significative o che descrivessero una classe di oggetti con esistenza autonoma, allo stesso modo tutti quei concetti aventi una struttura semplice e non in possesso di proprietà rilevanti sono stati rappresentati come attributi di un altro concetto a cui facessero riferimento. Sono stati invece rappresentati come relazioni tutti quei concetti che non avessero essi stessi le caratteristiche di un’entità ma che invece ne associassero due o più.

Tramite il rispetto della metodologia progettuale sopra descritta, lo schema concettuale ottenuto, rappresentato in figura, risulta rispettare i requisiti di qualità, ovvero la correttezza, la completezza, la leggibilità e la minimalità. La correttezza riguarda sia l’aspetto semantico che sintattico. La completezza implica la rappresentazione di tutti i dati di interesse e la possibilità di esecuzione di tutte le operazioni a partire dai concetti descritti nello schema. La minimalità prevede invece che tutte le specifiche sui dati siano rappresentate una sola volta all’interno dello schema, ovvero che non vi siano ridondanze, Nel caso dello schema prodotto è in realtà presente una forma di ridondanza che per valutazioni successivamente documentate si è preferito mantenere.

Design Pattern

Durante la progettazione dello schema concettuale, è stato fatto utilizzo oltre che di una metodologia di progetto, come precedentemente illustrato, anche di design pattern, ovvero di soluzioni progettuali a problemi comuni, riscontrabili durante la fase di progettazione concettuale dei dati. Nella realtà di interesse trattata è stato necessario applicare il design pattern riguardante la reificazione di una relazione ternaria. Si riportano di seguito la visualizzazione delle entità e delle associazioni coinvolte nell’applicazione di tale pattern.

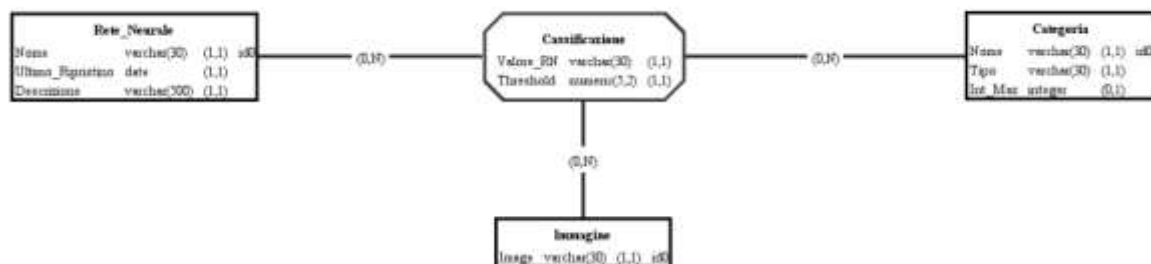


Figura 2. Schema precedente all’applicazione del pattern di reificazione di relazione ternaria

L’applicazione di tale pattern è stata fondamentale nel momento in cui ci si è resi conto che la classificazione fosse un concetto di per sé autonomo, e dunque da dover rappresentare come un’entità piuttosto che come un’associazione. Si è proceduto in tale fase anche all’aggiunta di un ulteriore attributo, estremamente importante all’interno della gestione dell’applicativo, come si vedrà in seguito, ovvero la data di svolgimento di tale classificazione. Come si vede nella rappresentazione successiva all’applicazione del pattern, e così come è previsto dal medesimo, l’associazione reificata avrebbe dovuto essere identificata univocamente dall’identificatore esterno formato dalle entità *Rete_Neurale*, *Immagine* e *Categoria*.

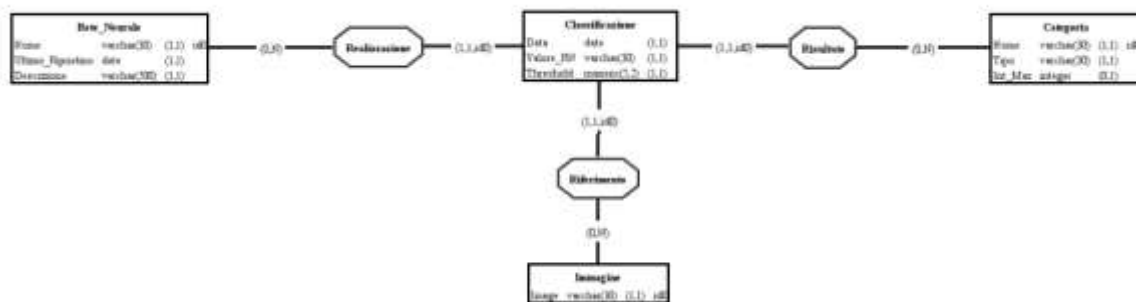


Figura 3. Schema successivo all'applicazione del pattern di reificazione di relazione ternaria

Un identificatore di tale genere avrebbe di fatto violato la possibilità, espressa tra i requisiti del portale, di consentire che la stessa immagine fosse classificata più volte dalla medesima rete neurale e sulle medesime categorie. Si è di conseguenza proceduto all'introduzione di un attributo *Codice* per individuare l'entità *Classificazione*, così come si può vedere all'interno dello schema concettuale.

Documentazione dello schema concettuale

Nonostante lo schema concettuale ottenuto sia esemplificativo della realtà di interesse, lo si è corredato con una documentazione di supporto che ne facilitasse l'interpretazione e che descrivesse proprietà aggiuntive dei dati, non esprimibili direttamente tramite i costrutti del modello. Tale documentazione è stata espressa in forma di regole aziendali. Nello specifico, le regole aziendali di tipo descrittivo, necessarie per la documentazione dei vari concetti rappresentati nello schema, sono state prodotte facendo uso di un dizionario dei dati. Tale dizionario risulta essere composto da una tabella per la descrizione delle entità dello schema, dei loro attributi e dei possibili identificatori e da un'altra per la descrizione delle relazioni, dei loro attributi oltre che delle entità coinvolte insieme alla loro cardinalità di partecipazione.

Entità	Descrizione	Attributi	Identificatore
Macrocategoria	Insieme delle categorie che caratterizzano un'immagine	Nome, Descrizione	Nome
Rete_Neurale	Modello computazionale utilizzato per classificare le immagini	Nome, Ultimo_Ripristino, Descrizione	Nome
Immagine	Immagine che l'utente classifica e desidera far classificare alla rete neurale	Image	Image
Classificazione	Risultato dell'analisi della rete neurale	Codice, Data, Valore_RN, Threshold	Codice
Categoria	Caratteristica dell'immagine da classificare	Nome, Tipo, Int_Max	Nome

Tabella 2. Dizionario dei dati – Entità

Relazioni	Descrizione	Entità Coinvolte	Attributi
Destinazione	Associa la rete neurale alla macrocategoria a cui è destinata	Rete_Neurale (1,1), Macrocategoria (0,N)	
Composizione	Associa la macrocategoria alle categorie di cui è composta	Macrocategoria (1,N), Categoria (1,N)	
Appartenenza	Associa l'immagine alla macrocategoria a cui appartiene	Macrocategoria (0,N), Immagine (1,N)	
Realizzazione	Associa la classificazione alla rete neurale da cui è stata realizzata	Rete_Neurale (0,N), Classificazione (1,1)	
Riferimento	Associa la classificazione all'immagine a cui fa riferimento	Immagine (0,N), Classificazione (1,1)	
Valorizzazione	Associa la categoria all'immagine valorizzata	Immagine (1,N), Categoria (0,N)	Valore_V
Risultato	Associa alla categoria il risultato della classificazione	Classificazione (1,1), Categoria (0,N)	
Prestazione	Associa alla rete neurale il valore delle proprie prestazioni relativo alla categoria	Rete_Neurale (0,N), Categoria (0,N)	Valore_P

Tabella 3. Dizionario dei dati – Relazioni

Le regole di vincolo e di derivazione sono state invece utilizzate per documentare vincoli espressi con i costrutti del modello E-R, per descrivere vincoli non esprimibili direttamente nello schema, e per illustrare un concetto ottenibile tramite inferenza o un calcolo aritmetico da altri concetti dello schema. Tali regole sono state espresse tramite asserzioni atomiche.

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

Tabella 4. Regole di vincolo

All'interno delle seguenti regole di derivazione con N si fa riferimento al numero di classificazioni effettuate dalla specifica rete neurale per la specifica categoria.

Regole di derivazione
(RD1) 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 $
(RD2) L'attributo Valore_P della Prestazione, quando i valori sono stringhe, a partire dalla prima classificazione, è derivabile attraverso la formula:
$\frac{1}{N} \sum_{n=0}^{N-1} (Valore_V_n == Valore_RN_n)$
(RD3) 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:
$Valore_P = \frac{Valore_P * (N - 1) + Valore_V - Valore_RN }{N}$
(RD4) 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:
$Valore_P = \frac{Valore_P * (N - 1) + (Valore_V == Valore_RN)}{N}$

Tabella 5. Regole di derivazione

2.1.2. Progettazione Logica

La fase di progettazione logica ha avuto come obiettivo la costruzione di uno schema logico in grado di descrivere correttamente ed efficientemente tutte le informazioni contenute nello schema Entità-Relazione prodotto in fase di progettazione concettuale. La progettazione logica si è articolata in due fasi, una prima fase di ristrutturazione dello schema Entità-Relazione e una seconda fase di traduzione verso il modello logico. Lo scopo della ristrutturazione è stato raggiungere un'ottimizzazione dello schema e semplificare la fase successiva di traduzione, questo sia per risolvere le problematiche relativamente alla presenza di costrutti del modello Entità-Relazione che non hanno una traduzione naturale nei modelli logici, sia perché la progettazione logica costituisce la base per la realizzazione dell'applicazione e deve quindi tener conto delle sue prestazioni. Per la fase di traduzione è stato invece necessario fare riferimento ad un particolare modello logico, e a tale fine è stato scelto il modello relazionale.

Ristrutturazione dello schema E-R

La fase di ristrutturazione dello schema E-R ha previsto preliminarmente una valutazione sul carico applicativo. Nello specifico, infatti la fase di ottimizzazione prevista ha tenuto conto di alcuni indici di prestazione, quali il costo delle operazioni e l'occupazione di memoria, valutabili tramite informazioni sul volume dei dati e sulle caratteristiche delle operazioni. Le informazioni sul volume dei dati sono state ottenute valutando il numero di entità e di associazioni dello schema e le dimensioni di ciascun attributo. Le caratteristiche delle operazioni fanno riferimento invece al tipo di operazione (interattiva o batch), alla frequenza con cui viene svolta e ai dati coinvolti.

Si è proceduto quindi alla compilazione di una tavola dei volumi che illustrasse, per tutti i concetti dello schema, il volume previsto a regime. Durante la definizione dei volumi sono state fatte alcune assunzioni, ovvero poiché le macrocategorie sono formate da un numero variabile di categorie, e poiché ciascuna rete neurale associata ad una specifica macrocategoria effettuerà una classificazione per ciascuna categoria, è stato ipotizzato che in media per ogni immagine una rete neurale debba effettuare due diverse classificazioni. Ulteriore ipotesi, a partire dalla quale sono stati valutati i volumi è che a regime ciascuna rete neurale abbia effettuato 300 classificazioni.

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

Tabella 6. Tavola dei volumi

Durante la fase di analisi dei requisiti si era già proceduto con l'individuazione delle operazioni principali e della corrispondente frequenza

Per fare una stima del reale carico della base di dati ci si è potuti concentrare sulle sole operazioni in accordo alla cosiddetta regola "ottanta-venti" tale per cui l'ottanta per cento del carico all'interno di una base di dati viene generato dal venti per cento delle operazioni.

Terminata la stima sugli indici di prestazione ha avuto inizio la fase di ristrutturazione dello schema E-R. Nello specifico tale ristrutturazione si è articolata nelle sole fasi di analisi delle ridondanze e di scelta degli identificatori principali. Non vi è stata necessità di procedere anche con l'eliminazione delle generalizzazioni e al partizionamento o accorpamento di entità o associazioni. Non vi erano infatti generalizzazioni nello schema concettuale di partenza, tutti concetti rilevanti nella realtà di interesse erano correttamente divisi e distinti e inoltre non erano presenti attributi multi-valore.

Nell'analisi delle ridondanze, ovvero di quei dati ottenibili tramite una serie di operazioni da altri dati, la valutazione è stata svolta sull'attributo "Valore_P" nell'entità *Prestazione*. Tale valore poteva infatti essere derivato per inferenza tramite le formule indicate all'interno delle regole di derivazione. Tale ridondanza rientra negli attributi derivabili da attributi di altre entità o associazioni. Per tale analisi, e sulla necessità o meno di preservare tale attributo, sono state prese in considerazione l'operazione 1 e l'operazione 2, valutandone il numero di accessi da effettuare sia in presenza della ridondanza che senza.

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

- **Operazione 1: Stampa prestazioni: stampa i valori delle prestazioni di una rete neurale**
Per tale operazione sono stati valutati il numero di accessi da effettuare in lettura e in scrittura sia in presenza della ridondanza che senza. Bisogna ovviamente tenere conto del fatto che è stato ipotizzato che per ciascuna immagine, ogni rete neurale debba effettuare due diverse classificazioni, dunque saranno due le prestazioni associate a ciascuna rete neurale.

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

- **Operazione 2: Aggiorna prestazione (aggiorna il valore della prestazione di una rete neurale per una categoria in seguito ad una classificazione)**

Per tale operazione sono stati valutati il numero di accessi da effettuare in lettura e in scrittura sia in presenza della ridondanza che senza. Ovviamente in caso di assenza della ridondanza l'operazione di aggiornamento delle prestazioni di una rete neurale per una specifica categoria non viene svolta.

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 è stata fatta una valutazione sul numero di accessi e sull'occupazione di memoria che tale dato comportava, in relazione al 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;
- In assenza di ridondanza il costo delle operazioni è di 30100 accessi giornalieri;

Pertanto, si decide di mantenere la ridondanza in quanto riduce il numero di accessi in maniera sostanziale a fronte di un'occupazione di memoria è trascurabile.

La fase di scelta degli identificatori principali è stata fondamentale per arrivare alla traduzione verso il modello relazionale perché in tale modello le chiavi sono state ovviamente usate per stabilire legami tra dati in relazioni diverse. In tali scelte è stata posta particolare attenzione alla semplicità, nello specifico si sono preferiti identificatori formati da uno o pochi attributi, garantendo così che le strutture ausiliarie (gli indici) create dai sistemi di gestione di basi di dati fossero di dimensioni ridotte, ottenendo un risparmio di memoria nella realizzazione dei legami logici e una facilitazione delle operazioni di join. Allo stesso modo, onde evitare di avere chiavi con molti attributi, si sono preferiti identificatori interni piuttosto che esterni.

Per il soddisfacimento delle succitate motivazioni, lì dove non ci fossero identificatori candidati tali da poterle soddisfare, si è proceduto all'introduzione di un ulteriore attributo all'entità tale da contenere dei codici generati appositamente per identificare le occorrenze dell'entità.

Le entità per cui è stata fatta tale aggiunta sono state:

- *Immagine*: per la quale in precedenza era stato individuato l'identificativo "Image"
- *Rete_Neurale*: per la quale in precedenza era stato individuato l'identificativo "Nome"

Al termine dei vari passi compiuti durante la fase di ristrutturazione dello schema Entità-Relazione si è giunti allo schema ristrutturato finale, di seguito riportato, dal quale partire per la fase di traduzione verso il modello relazionale.

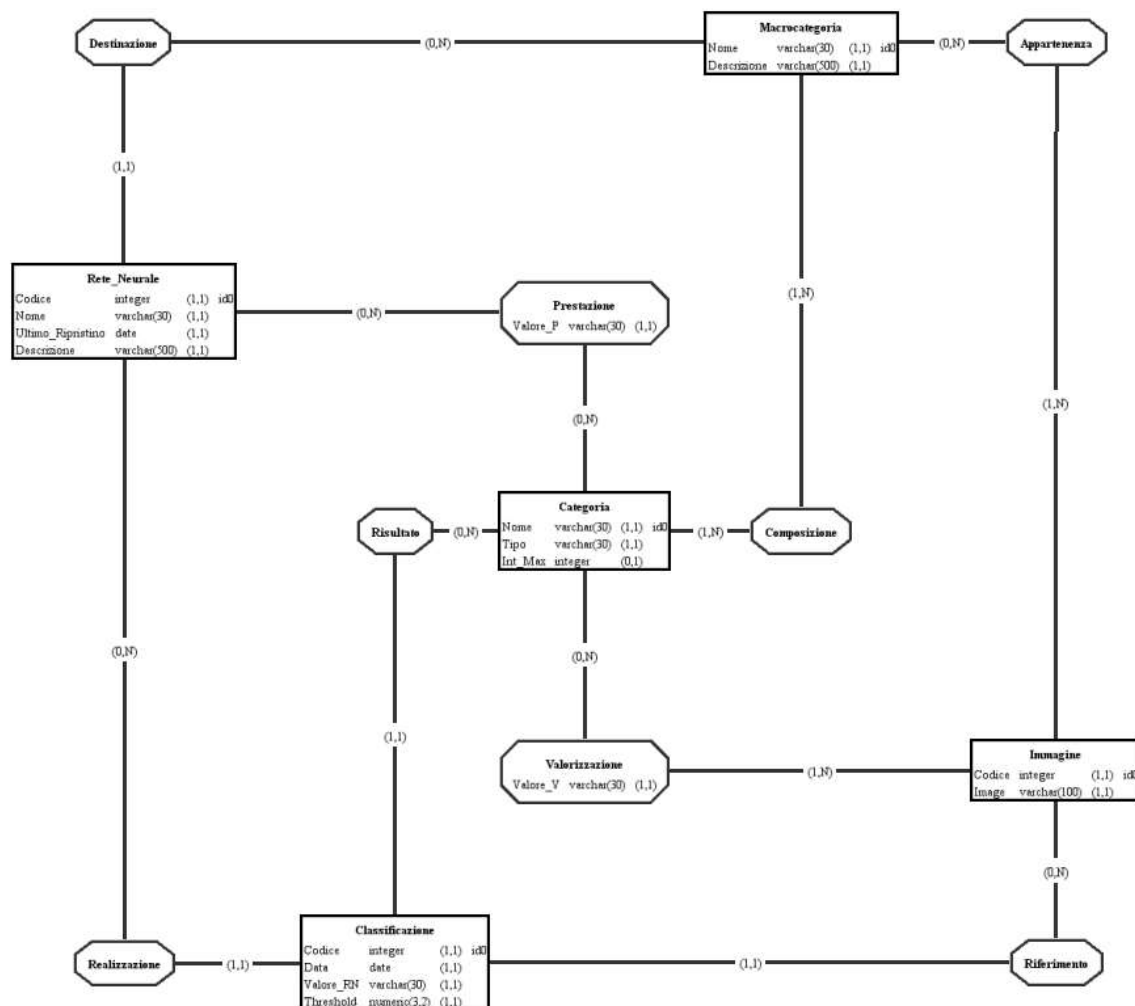


Figura 2. Schema ER Ristrutturato

Traduzione verso il modello logico

La seconda fase della progettazione logica ha riguardato la traduzione, ovvero la realizzazione, a partire da uno schema E-R ristrutturato, di uno schema logico equivalente, in grado cioè di rappresentare le medesime informazioni. Lo schema logico ottenuto, a seguito della corretta traduzione delle cardinalità delle associazioni individuate durante la progettazione concettuale, prevede nove relazioni. Per queste vanno ricordate alcune chiavi alternative, quali: l'attributo *Image* per la relazione *Immagine* e l'attributo *Nome* per la relazione *Rete_Neurale*.

IMMAGINE (Codice, Image)

APPARTENENZA (Macrocategoria, Immagine)

MACROCATEGORIA (Nome, Descrizione)

RETE_NEURALE (Codice, Nome, Ultimo_Rispristino, Descrizione, Macrocategoria)

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

CATEGORIA (Nome, Tipo, Int_Max)

VALORIZZAZIONE (Categoria, Immagine, Valore_V)

COMPOSIZIONE (Macrocategoria, Categoria)

PRESTAZIONE (Rete_Neurale, Categoria, Valore_P)

Lo schema logico realizzato ha ereditato buona parte della documentazione dello schema concettuale, ovvero le regole aziendali. Per quanto riguarda invece la descrizione dei vincoli di integrità referenziale introdotti dalla traduzione è stato adottato un formalismo grafico che fosse anche in grado di rappresentare le relazioni con i relativi attributi. Tale visualizzazione viene di seguito riportata.

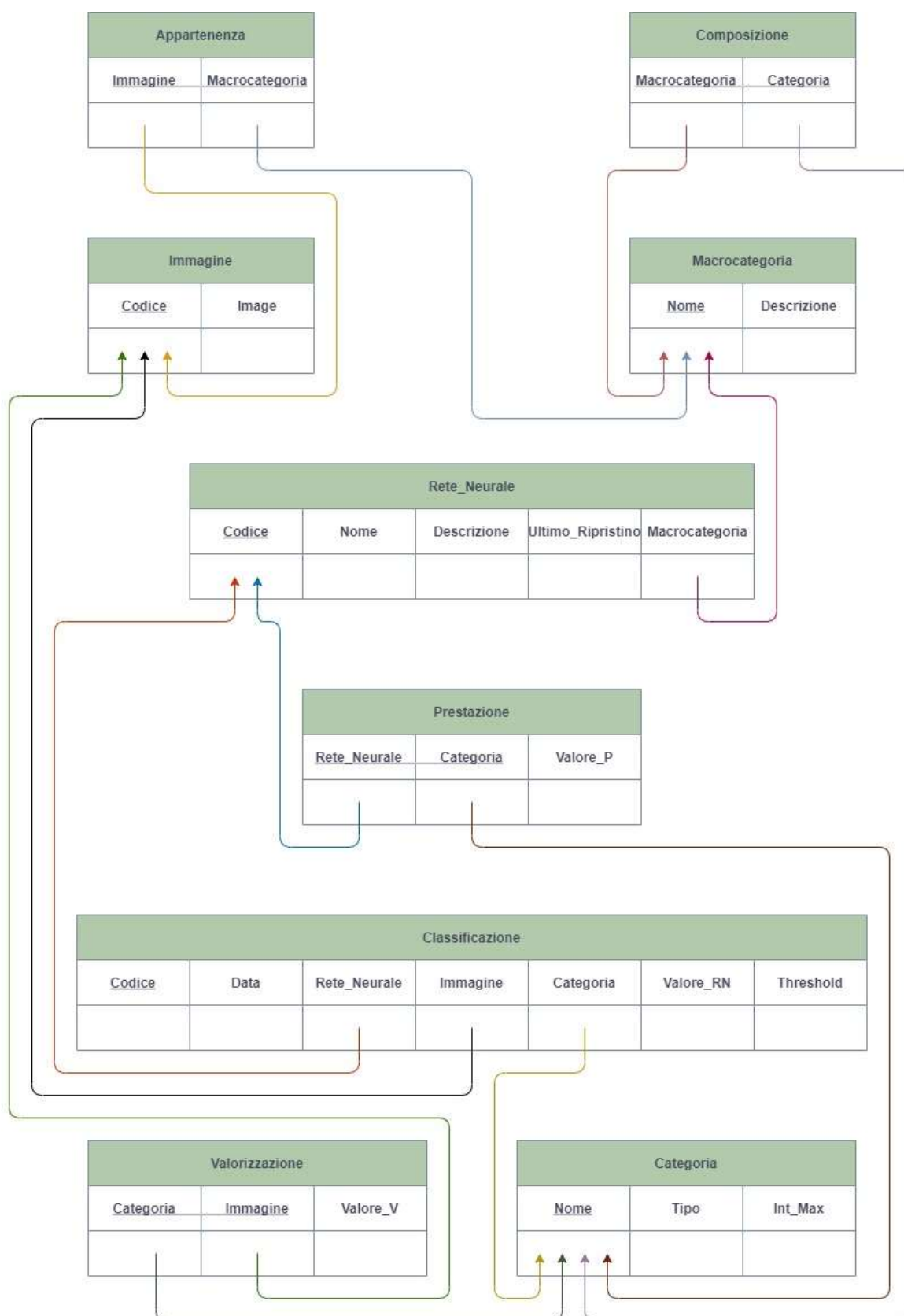


Figura 5. Documentazione dello schema logico

2.1.3. Normalizzazione

Sullo schema della base di dati progettato si è proceduto ad una certificazione di qualità, ovvero alla verifica del soddisfacimento delle forme normali, in modo tale da valutare la necessità di intervenire o meno con un'operazione di normalizzazione onde evitare la presenza di ridondanze o di anomalie durante le operazioni di aggiornamento, di cancellazione o di inserimento.

Per la verifica di tali forme normali si sono individuate, per ciascuna relazione, le dipendenze funzionali non banali, ovvero quei vincoli che descrivono legami di tipo funzionale tra gli attributi della relazione, tali da descrivere proprietà significative dell'applicazione rappresentata.

IMMAGINE (Codice, Image)

Codice → Image

Image → Codice

MACROCATEGORIA (Nome, Descrizione)

Nome → Descrizione

RETE_NEURALE (Codice, Nome, Ultimo_Rispristino, Descrizione, Macrocategoria)

Codice → Nome, Ultimo_Rispristino, Descrizione, Macrocategoria

Nome → Codice, Ultimo_Rispristino, Descrizione, Macrocategoria

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

Codice → Data, Rete_Neurale, Immagine, Categoria, Valore_RN

CATEGORIA (Nome, Tipo, Int_Max)

Nome → Tipo, Int_Max

VALORIZZAZIONE (Categoria, Immagine, Valore_V)

Categoria, Immagine → Valore_V

PRESTAZIONE (Rete_Neurale, Categoria, Valore_P)

Rete_Neurale, Categoria → Valore_P

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

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

Dall'analisi delle dipendenze funzionali, possiamo affermare che tutte le relazioni rispettano i requisiti della **terza forma normale** in quanto si trovano in seconda forma normale e ogni attributo non primo (attributo non appartenente a nessuna chiave candidata della relazione) dipende in modo non transitivo da ogni chiave della relazione.

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 e inoltre ogni determinante delle dipendenze funzionali individuate è una chiave candidata. Dunque, non vi è stata la necessità di procedere con alcuna decomposizione.

2.2. Progettazione delle applicazioni

Terminata la fase di progettazione della base di dati, supporto imprescindibile per un corretto funzionamento del sistema, si è proceduto alla progettazione dell'applicazione comprendente l'interfaccia con l'utente e i vari programmi di servizio. A tale fine è stato necessario scandire delle fasi di descrizione e di progettazione tali da soddisfare sia le esigenze dei moduli di front-end sia di back-end.

Tale fase progettuale ha riguardato l'ideazione e la predisposizione delle interfacce utente che sarebbero poi state utilizzate, sia dal punto di vista della visualizzazione grafica che delle relazioni che avrebbero dovuto legarle. Durante tale fase è stata svolta un'analisi finalizzata all'inquadramento del numero di pagine necessarie al soddisfacimento di tutte le funzionalità individuate e, conseguentemente, per ciascuna di esse le opportunità da offrire all'utente che vi interagisse. Per quanto riguarda l'aspetto relativo alla visualizzazione grafica, al fine di avere un'idea della struttura delle diverse pagine web è stata utilizzata l'applicazione Marvel. Gli obbiettivi perseguiti durante tale ideazione hanno portato alla necessità, in fase realizzativa, di prevedere diverse pagine web, ciascuna con funzionalità e scopi differenti, i cui contenuti visualizzabili agli utenti fossero condizionati dal fatto che questi ultimi fossero autenticati o meno. Per tali pagine sono state individuate alcune caratteristiche comuni e altre invece particolarizzate per ciascuna visualizzazione.

2.2.1. Caratteristiche comuni individuate

Tutte le pagine sono state progettate in modo tale da avere un header ed un footer, tali che il primo fosse sempre visualizzabile nella pagina (sticky-header) e consentisse all'utente di ritornare alla homepage, mentre il secondo contenesse contatti utili per l'utente. In parte la homepage si discosta dalle altre, in queste due sezioni. Al fine di rendere la grafica delle pagine coinvolgente ed omogenea si è scelto di utilizzare codifiche ricorrenti nelle forme e nei colori per i componenti con medesime finalità. In particolare, per le varie icone e pulsanti è stato scelto il colore giallo, utilizzato anche per evidenziare l'interazione del visitatore con i vari componenti cliccabili, realizzati di base con il colore bianco. Le scritte di tipo informativo sono realizzate con il colore nero, fatte eccezione quelle corredate a delle immagini di sfondo. Sia le immagini utilizzate eventualmente come sfondi, che in generale tutte quelle presenti all'interno del sito sono ricche di una forte componente esplicativa in accordo alla sezione all'interno della quale sono integrate, ponendo nella maggior parte dei casi attenzione su una delle tematiche attualmente più comune e fortemente relazionata al sito web, ovvero il confronto uomo-macchina e l'enorme rincorrersi di intelligenza artificiale e naturale. Tali scelte durante la progettazione sono state mosse dal desiderio di assicurare omogeneità tra le diverse pagine, chiarezza simbolica e semplicità di interpretazione.

2.2.2. Homepage

L'Homepage rappresenta una vera e propria pagina di presentazione del portale oltre che un passaggio obbligato per qualsiasi utente. Nella homepage, tramite l'interazione con apposita barra di navigazione sarà possibile per l'utente consultare diverse sezioni all'interno delle quali reperire diverse informazioni.

- Alla voce “Home” l’utente visualizzerà, le quattro macrocategorie: Face Counting, People Counting, Vehicle Counting e Face Analysis, tramite le quali, se autenticato, potrà effettuare la sua scelta.
- Alla voce “Info” sarà possibile per l’utente consultare una descrizione accurata del portale web e delle funzionalità da esso messe a disposizione.
- Alla voce “Reti Neurali” l’utente potrà consultare una breve descrizione di cosa siano le reti neurali, con la possibilità di accedere ad una documentazione più approfondita, oltre che il nome ed una breve descrizione delle reti neurali di cui il portale dispone.
- Alla voce “Portfolio” sarà possibile visualizzare delle immagini di esempio, ovvero delle immagini appartenenti alle diverse macrocategorie. In tale sezione, tramite un filtro predisposto, l’utente potrà scegliere se visualizzare tutte le immagini o soltanto quelle contenenti uno specifico soggetto da lui scelto, oltre che ingrandirle e consultarle nel dettaglio. Al di sotto di tale sezione vi è la possibilità di consultare sia il numero di immagini attualmente presenti all’interno del database suddivise secondo la macrocategoria di appartenenza, sia una descrizione delle diverse macrocategorie presenti all’interno del portale con le relative categorie di riferimento.
- Alla voce “Team” vi è la possibilità di conoscere i membri del team gestore del portale, con la possibilità di contattarli tramite i Social Networks più diffusi quali Twitter, Facebook, Instagram e LinkedIn.
- Alla voce “Contatti” è possibile visualizzare sulla mappa il luogo di riferimento come sede del team che ha ideato il portale e un contatto a cui fare riferimento per segnalare eventuali problematiche riscontrate sul portale, al quale sarà possibile automaticamente inviare un’e-mail tramite l’apposito pulsante “Contattaci”.

La barra di navigazione con la quale interagire per reperire le informazioni sopracitate è inquadrata all’interno dell’header della homepage all’interno del quale è previsto il logo del portale e i pulsanti per effettuare l’accesso e la registrazione. Tramite il pulsante “Registrati” l’utente sarà reindirizzato alla pagina di registrazione, tramite il pulsante “Accedi” verrà reindirizzato al footer della homepage stessa, dove previo inserimento dei propri username e password potrà procedere all’autenticazione. Il corpo centrale della homepage precedentemente descritto, ovvero quello consultabile tramite l’interazione con la barra di navigazione, ha una visualizzazione indipendente dal fatto che l’utente sia autenticato o meno, ma solo l’utente autenticato potrà effettuare la scelta della macrocategoria ed essere reindirizzato alla pagina di test.

L’header e il footer hanno invece una visualizzazione differente per i due utenti. Nel caso in cui l’autenticazione sia stata effettuata, nell’header i due pulsanti precedentemente descritti saranno sostituiti da un’icona-profilo e dal pulsante “Esci” per effettuare la disconnessione. Invece all’interno del footer non sarà più presente la sezione per effettuare l’autenticazione.

Con lo stesso obiettivo si è predisposto un pulsante scroll-to-top che consentisse all’utente in qualsiasi punto della pagina, di essere riportato in cima.

2.2.3. Registrazione

La pagina di registrazione è stata progettata in modo da consentire all'utente di sottoscrivere i propri dati e le proprie credenziali da utilizzare successivamente per l'accesso. Il body della pagina prevede diversi campi da compilare per l'utente, ovvero il nome, il cognome, la e-mail, l'username e la password per la quale è prevista anche una riconferma. Vi sono i due pulsanti "Registrati" e "Reset", il primo per la sottoscrizione dei campi compilati, il secondo per resettare i campi inseriti. In fase di ideazione di tale pagina è stata posta grande attenzione anche sull'aspetto riguardante i controlli da implementare affinché i dati inseriti dall'utente siano effettivamente corretti. In tale fase i controlli che ci si è posti l'obiettivo di realizzare nel corso dell'implementazione successiva sono stati:

- l'assenza di numeri e caratteri speciali nei campi del nome e del cognome, ci si è cioè assicurati giustamente che fossero formati da soli caratteri alfabetici;
- la presenza all'interno dell'e-mail del carattere "@" e di un dominio, nell'ordine corretto;
- il rispetto per la password di determinati prerequisiti, ovvero una lunghezza minima di sei caratteri, la presenza di almeno un carattere maiuscolo e l'assenza di spazi bianchi;
- la coerenza tra la password inserita e la riconferma di quest'ultima;
- il completamento di tutti i campi prima della sottoscrizione.

In tale fase è stata prevista anche la presenza di una progress-bar che servisse per validare la robustezza della password inserita dall'utente.

Dunque, la sottoscrizione da parte dell'utente può concludersi soltanto qualora tutti i controlli sopra elencati abbiano un riscontro positivo, si procederebbe altrimenti a notificare all'utente eventuali problematiche.

In seguito alla corretta registrazione dell'utente si è previsto che automaticamente venga effettuata l'autenticazione e venga reindirizzato alla Homepage, senza essere costretto ad effettuare l'accesso.

2.2.4. Test

Il body della pagina di test è invece strutturato in modo tale da essere diviso in due aree simmetriche lungo l'altezza raccordate sotto una sezione comune che contenga il nome della macrocategoria scelta nella homepage. I campi la cui compilazione viene richiesta all'utente saranno distribuiti tra le due aree, quella di sinistra e quella di destra. Nella prima vi sarà la possibilità di caricare un'immagine appartenente alla macrocategoria selezionata, tramite l'apposito pulsante "Carica Immagine". La seconda invece è organizzata in due diverse sezioni, riguardanti aspetti differenti, che richiedono all'utente di specificare alcuni parametri, riguardanti la valorizzazione delle categorie e i parametri relativi alla rete neurale (nome e threshold).

Per i campi da completare presenti in queste due sezioni sono state scelte delle strutture specifiche, ovvero per le categorie che richiedevano l'inserimento di un valore numerico si è progettato l'utilizzo di un input di tipo numerico con uno step di un'unità e il cui massimo valore esprimibile fosse condizionato dal tipo di categoria in questione. Per quanto riguarda invece le categorie quali etnia ed emozione è stato progettato l'utilizzo di una select, in modo da mostrare all'utente tutte le opzioni possibili tra cui scegliere. Per quanto riguarda invece la categoria genere, essendo i valori possibili vincolati a due sole opzioni si è preferito l'utilizzo di un input di tipo radio. Tale progettazione è in linea a quanto ideato nel corso della progettazione concettuale della base di dati, durante la quale per l'entità categoria è stato previsto l'attributo "Tipo" al fine di consentire all'amministratore,

all'aggiunta di una nuova categoria, di decidere il tipo di controllo da associargli all'interno del test. La scelta del valore di threshold può essere effettuata tramite un input di tipo range, per il quale lo step scelto è pari a 0.01, in seguito all'analisi delle funzioni associate alle reti neurali fornite. Ciascuna delle sezioni sopra descritte è stata strutturata con la possibilità di scorrere i vari campi da compilare, in modo tale che seppure il numero di categorie e quindi di valori da esprimere, dovesse essere diverso, le dimensioni e la visualizzazione rimarrebbe inalterata.

Affinché l'utente possa visualizzare il risultato delle classificazioni effettuate dalla rete neurale sull'immagine da lui caricata è necessario che clicchi sul pulsante "Conferma" a patto di aver però compilato tutti i campi richiesti. Durante la fase progettuale sono state previste eventuali notifiche per l'utente qualora i campi non fossero tutti correttamente compilati.

2.2.5. Result

Tale pagina è quella che consente all'utente di fare un confronto tra la valorizzazione da lui effettuata sull'immagine e la classificazione effettuata dalla rete neurale sulle medesime categorie.

Sempre con lo spirito di ricerca di un'omogeneità grafica al fine di rendere l'interazione più intuitiva, tale pagina web è stata progettata con l'obiettivo di ottenere un'affinità grafica con la pagina in cui viene svolto il test. Dunque, il corpo centrale è stato anch'esso strutturato in due diverse aree, tali per cui sulla sinistra l'utente può visualizzare l'immagine caricata, mentre sulla destra può consultare diverse informazioni organizzate in tre distinte sezioni. Le prime due sezioni, organizzate con una struttura tabellare completamente identica, conterranno i valori espressi, per ciascuna categoria classificata, dall'utente e dalla rete neurale. Tramite la consultazione di queste due sezioni si potrà effettivamente avere un confronto sul reale distacco che vi è, eventualmente in termini numerici, tra i diversi valori su ciascuna categoria. La terza e ultima sezione, anche essa basata su strutture tabellari, consentirà la visualizzazione per ciascuna categoria dell'indice di errore compiuto dalla rete neurale sul numero di immagini classificate, espresso o in termini numerici o in termini percentuali. Tale valore viene ricavato in tempo reale sfruttando le formule individuate in fase di progettazione della base di dati per l'aggiornamento del dato ridondante che, per motivazioni di semplicità delle operazioni, è stato mantenuto. Nel caso in cui la categoria di riferimento fosse esprimibile tramite un numero, tale valore sta a rappresentare di quanto in media si distacca la classificazione della rete neurale dalla valorizzazione dell'utente su quella specifica categoria. Qualora invece la categoria non prevedesse valori numerici, l'errore sarebbe espresso in percentuale, come indice di insuccesso medio sul numero di immagini classificate.

All'interno di tale sezione saranno inoltre visualizzabili due parametri fondamentali alla corretta interpretazione dei dati mostrati, ovvero il threshold selezionato dall'utente in fase di caricamento dell'immagine e il numero di classificazioni effettuate fino a quel momento dalla rete neurale. Il primo parametro influenza, infatti, il risultato espresso dalla rete neurale, il secondo invece può essere utile per la corretta interpretazione dell'indice di performance della rete.

Da tale pagina, tramite il pulsante "Carica Nuova Immagine" l'utente viene reindirizzato alla pagina di test per poter nuovamente caricare un'immagine appartenente alla medesima macrocategoria della precedente ed effettuare nuovamente il test.

3.Implementazione

La fase di implementazione si è sviluppata a partire dalle scelte, dalle valutazioni e dai risultati che hanno caratterizzato la fase progettuale. Tutti i risultati ottenuti in precedenza, sia inerenti ai dati che inerenti applicazioni, hanno rappresentato solide fondamenta su cui procedere per la realizzazione del sistema. Proprio per questo motivo si è voluto strutturare la fase implementativa, articolandola sui medesimi fronti che avevano caratterizzato la progettazione. A tal proposito la fase di progettazione dei dati è stata fondamentale per realizzare un'implementazione della base di dati corretta e conforme alle specifiche, così come la fase di progettazione delle applicazioni è stata imprescindibile per l'individuazione degli aspetti da implementare nella web application, sia per la componente front-end che back-end. La fase di implementazione si è dunque concentrata sull'implementazione del back end, riguardante sia la base di dati che la logica applicativa, e sull'implementazione del front-end, mirato al soddisfacimento dei requisiti di interfaccia.

Ciascuna di queste operazioni è realizzata tramite il supporto offerto dai diversi applicativi software utilizzati, estremamente diffusi e all'avanguardia.

3.1. Implementazione del back-end

Il back-end rappresenta probabilmente la parte fondamentale di un'applicazione, senza di esso infatti non sarebbe possibile utilizzare tutte le funzionalità presenti sulle interfacce. Rappresenta sostanzialmente l'infrastruttura base del software. La fase di implementazione del back-end si è dunque articolata in una fase di implementazione della base di dati e in una fase di implementazione dell'applicazione.

3.1.1. Implementazione della base di dati

La fase di progettazione della base di dati ha portato all'ottenimento dello schema logico, all'individuazione delle varie relazioni sulle quali, tramite le apposite documentazioni, erano stati individuati i diversi vincoli, intrarelazionali e interrelazionali. I vincoli intrarelazionali sono quei vincoli il cui soddisfacimento è definito rispetto a singole relazioni della base di dati, appartengono a questa categoria i vincoli di tupla, i vincoli di dominio e i vincoli di chiave. I vincoli interrelazionali sono invece quelli che coinvolgono più relazioni e appartengono a questa categoria i vincoli di integrità. La fase di implementazione della base di dati si è posta come obbiettivo non soltanto la traduzione, tramite l'apposito linguaggio, dello schema logico precedentemente ottenuto in tutte le sue parti, ma anche e soprattutto la verifica della correttezza dei moduli previsti. Al fine di soddisfare questo obbiettivo è stato necessario prevedere la creazione di tutte le relazioni individuate, effettuare una traduzione dei vari vincoli su esse individuati e realizzarne il popolamento, al fine di poter validare il prodotto.

Il database di riferimento utilizzato è stato PostgreSQL tramite la sua interfaccia grafica PgAdmin.

Creazione della base di dati

In fase di creazione della base di dati, si è proceduto alla creazione, in linguaggio SQL, delle relazioni definite in fase di traduzione, e alla specifica di parte dei vincoli su esse definiti.

I vincoli la cui gestione è stata messa a punto in questa sede sono stati quelli riguardanti tramite l'individuazione delle chiavi primarie, degli attributi che ammettessero valori nulli e quelli invece obbligatori, e degli eventuali valori di default associati. Sono stati gestiti inoltre, tra i vincoli interrelazionali, i vincoli di integrità referenziale per i quali sono state definite le corrispondenti politiche di reazione.

Per politiche di reazione si intende la scelta, tra varie alternative possibili, delle reazioni da mettere in atto in risposta alle violazioni generate da modifiche sulla tabella esterna (tabella referenziata). Gli eventi che possono essere causa di tali violazioni sono eventi di cancellazione e di modifica, contrassegnate dagli eventi “on delete” e “on cascade” rispettivamente. Si sono individuate le alternative, spesso differenti a seconda del tipo di operazione, che maggiormente fossero adeguate alla situazione analizzata.

- Per la relazione *Appartenenza*, si è chiarito che eventuali modifiche o cancellazioni sulle relazioni referenziate, ovvero macrocategoria ed immagine dovessero ripercuotersi sulla relazione referenziante.
- Per la relazione *Rete_neurale*, tramite la clausola “restrict”, si è chiarita come fosse impossibile eliminare una macrocategoria sulla relazione referenziata, mentre che eventuali modifiche dovessero ripercuotersi sulle tuple della relazione referenziante.
- Per la relazione *Classificazione*, tramite la clausola “no action” si è chiarita l'impossibilità di procedere all'eliminazione di una rete neurale, di una categoria o di un'immagine sulle tabelle referenziate, mentre eventuali modifiche si ripercuoterebbero sugli attributi della tabella referenziante.
- Per la relazione *Valorizzazione*, tramite la clausola “no action”, si è chiarita l'impossibilità di procedere all'eliminazione della categoria nella relazione referenziata, mentre è concessa l'eliminazione di un'immagine così come le modifiche su entrambe, che si ripercuoterebbero a cascata sulla tabella referenziante.
- Per la relazione *Composizione*, si è chiarita l'impossibilità di procedere all'eliminazione di una categoria referenziata, è invece possibile effettuare cancellazioni delle macrocategorie e modifiche su entrambe, che si ripercuoteranno a cascata sulle tuple della relazione sopra citata.
- Per la relazione *Prestazione*, tramite la clausola “no action” si è chiarita l'impossibilità di procedere all'eliminazione di una categoria, sono ammesse invece, e si ripercuoteranno a cascata sulle tuple della relazione, eventuali modifiche sulle categorie e sulle reti neurali oltre che cancellazioni di queste ultime.

Implementazione dei trigger

Al fine di soddisfare alcune delle regole aziendali e di derivazioni individuate, è stato fondamentale implementare dei trigger che rendessero la base di dati attiva, capace cioè di reagire a determinati eventi definiti dall'amministratore, tramite l'esecuzione di opportune azioni. L'impiego dei trigger è

risultato necessario per la definizione di alcune regole conformi al paradigma Evento-Condizione-Azione (ECA), in modo da definire per ciascuna di esse un evento che, qualora la condizione fosse soddisfatta, determinasse l'esecuzione dell'azione associata.

Il soddisfacimento delle regole di vincolo relativamente alle cardinalità minime ha comportato l'implementazione di tre trigger, tutti aventi come modalità quella *after*, come evento di riferimento l'inserimento su una tabella target, con livello di granularità di tupla (*row-level*), in modo da determinarne l'attivazione per ogni tupla coinvolta. Proprio per lo scopo a cui questi trigger sono stati destinati è stato fondamentale, tramite la clausola *deferrable initially deferred*, renderli con una modalità differita, in maniera tale che la loro valutazione avvenisse alla fine della transazione, a seguito cioè del comando di *commit work*. Si riporta di seguito una breve descrizione dei trigger realizzati.

- *ins_macrocategoria*: trigger che soddisfa la regola di vincolo RV5, assicurandosi che all'inserimento di una nuova macrocategoria, sia presente per quest'ultima almeno una categoria associata. Al momento di inserimento di una nuova macrocategoria, all'interno della funzione associata al trigger si controlla che vi sia almeno una tupla nella relazione *Composizione* in cui la macrocategoria corrisponda a quella attualmente analizzata. All'interno di tale relazione sono infatti presenti le associazioni tra ciascuna macrocategoria e le rispettive categorie. Qualora il controllo non avesse esito positivo, verrebbe lanciata un'eccezione, con relativo messaggio di errore, che necessariamente causerebbe il *rollback* di tutte le operazioni eseguite all'interno della transazione.
- *ins_categoria*: trigger che soddisfa la regola di vincolo RV6, assicurandosi che all'inserimento di una categoria sia presente per quest'ultima almeno una macrocategoria a cui associarla. Al momento di inserimento di una nuova categoria, all'interno della funzione associata al trigger, si controlla che esista almeno una tupla all'interno della relazione *composizione* tale per cui la categoria sia la medesima di quella che si stava inserendo. Qualora tale controllo non dovesse andare a buon fine verrebbe lanciata un'eccezione con annesso messaggio di errore e questo determinerebbe il *rollback* di tutte le operazioni eseguite all'interno della transazione.
- *ins_immagine*: trigger che soddisfa le regole di vincolo RV7 e RV8, assicurandosi che ogni immagine inserita appartenga ad almeno una macrocategoria e che abbia almeno una categoria valorizzata. Nello specifico, all'inserimento di una nuova immagine, all'interno della funzione associata al trigger, si controlla che quest'ultima sia associata ad una macrocategoria. Successivamente si controlla che per tutte le categorie appartenenti alla macrocategoria in questione, sia stata fatta una valorizzazione relativa all'immagine caricata. In caso di fallimento di uno dei due controlli viene lanciata un'eccezione con annesso messaggio di errore personalizzato a seconda della violazione rilevata.

Durante tale fase è inoltre stata fatta una scelta in merito alla modalità con cui implementare l'aggiornamento delle prestazioni della rete neurale, per una specifica categoria, a seguito di una nuova classificazione. Sicuramente tale aggiornamento avrebbe potuto essere effettuato tramite una funzione inserita all'interno del programma applicativo. Si è però preferito implementare tale funzionalità tramite un trigger. La motivazione di tale scelta risiede nella volontà di rendere la base di dati completamente consistente in maniera autonoma, aggiungendo ad essa un ulteriore livello di indipendenza, oltre quello fisico e logico, ovvero l'indipendenza della conoscenza.

Tale trigger, denominato *modify_performance*, viene attivato in seguito ad un inserimento nella relazione *Classificazione*, con una granularità di tupla. All'interno della funzione associata a tale trigger, vengono selezionati, associati alla specifica immagine oggetto della classificazione e per la specifica categoria in esame, sia il valore inserito dall'utente che quello individuato dalla rete neurale. Vengono prelevati inoltre l'attuale valore della prestazione della rete neurale per la categoria in esame, e il numero di classificazioni la cui data sia successiva a quella di ultimo ripristino (saranno considerate solo queste classificazioni nell'aggiornamento delle prestazioni e non in generale tutte quante quelle effettuate). Soltanto qualora la data di svolgimento della classificazione fosse successiva alla data di ultimo ripristino della rete neurale si procederebbe, tramite le formule individuate nelle regole di derivazione e tradotte in linguaggio PL/pgSQL, all'aggiornamento del valore corrente delle prestazioni.

Proprio perché l'attributo *Ultimo_Ripristino* della relazione *Rete_Neurale* serve ad effettuare una sorta di reset della rete neurale e più specificatamente delle sue prestazioni, è stato implementato un ulteriore trigger per soddisfare tale requisito. Il trigger *reset_performance*, attivato in seguito ad un'operazione di modifica sull'attributo *Ultimo_Ripristino*, ha una granularità di riga. La funzione ad esso associata prevede che i valori delle prestazioni di quella rete neurale siano resettati, ovvero siano posti a NULL.

Popolamento della base di dati

Il popolamento della base di dati è stato un passaggio fondamentale per la validazione dei moduli predisposti. Al fine di rispettare i vincoli aziendali individuati e le soluzioni rappresentate dai trigger per la gestione delle cardinalità minime, l'inserimento è stato implementato in modo tale da prevedere l'inserimento di ciascuna macrocategoria, delle relative categorie e delle corrispondenti composizioni all'interno di una medesima transazione. La scelta di inglobare tali operazioni all'interno di una transazione, ha fatto sì che qualora all'interno di uno dei trigger fosse sollevata l'eccezione, poiché il trigger e le operazioni che lo attivano sono considerati come facenti parte di un'unica transazione, si determinasse un rollback totale tale da annullare tutte le operazioni effettuate in quella transazione. Si è proceduto poi all'inserimento delle varie reti neurali e delle varie prestazioni. Durante la fase progettuale e ancor prima, durante la fase di analisi delle specifiche, si erano evidenziate le funzionalità che il portale avrebbe dovuto garantire all'utente. Dunque, per testare l'effettiva robustezza e adeguatezza dello schema logico realizzato e dei trigger annessi, sono stati implementati, all'interno di specifiche transazioni, degli inserimenti che simulassero le richieste che il database avrebbe dovuto soddisfare a regime, in una situazione di reale funzionamento. Si è proceduto dunque a simulare l'inserimento di un'immagine, alla definizione della macrocategoria di appartenenza, all'inserimento dei valori associati all'immagine per le specifiche categorie definite sulla macrocategoria e all'inserimento della classificazione che la rete neurale avrebbe fatto, utilizzando ovviamente, in questa fase, dei valori casuali. Grazie a tale simulazione si è potuto constatare che l'implementazione realizzata riusciva a soddisfare le operazioni principali individuate e che le soluzioni adottate per la gestione di cardinalità minime, aggiornamento delle prestazioni e ripristino di queste ultime fossero effettivamente solide e funzionanti.

Questa garanzia di correttezza ha fatto sì che i prodotti ottenuti in tale fase implementativa potessero essere integrati con i moduli successivamente realizzati, soprattutto i trigger.

3.1.2. Implementazione dell'applicazione

Per l'implementazione dell'applicazione è stato utilizzato il framework Django. Tale framework open source rappresenta un importante supporto allo sviluppo Web, cerca infatti di velocizzarla tramite l'offerta di diversi aspetti e moduli già implementati. Attraverso l'impiego di Django si è certi di realizzare un prodotto che sia robusto sul piano della sicurezza, evitando errori generalmente possibili come SQL injection e gestendo in maniera sicura l'aspetto dell'autenticazione. Nella sostanza offre quindi la possibilità di realizzare soluzioni altamente scalabili e flessibili.

Django è fondato sul paradigma "Model-Template-View". Nello specifico, dunque, tale fase implementativa si è strutturata in una serie di passi organizzati in modo da essere effettivamente conformi a tale paradigma. L'implementazione è stata dunque ripartita in una fase di implementazione dei modelli, una fase di implementazione dei template e una fase di implementazione delle viste. Per la realizzazione di tale progetto è stata effettuata la creazione di un'app Django, ovvero un'applicazione riusabile che consente un insieme di funzionalità finalizzate a scopi ben precisi, in sostanza ciascun progetto Django può essere costituito da una o più app.

Models

In tale fase implementativa si è fatto uso dei modelli Django, i quali nascono dalla programmazione orientata agli oggetti, ovvero dall'idea di modellare la realtà come un insieme di oggetti, ciascuno caratterizzato da proprietà ed azioni. Si è fatto utilizzo di tali modelli per salvare dei contenuti all'interno del database, in quanto i modelli rappresentano sostanzialmente l'unica e definitiva fonte di informazione sui dati, poiché contengono i campi e i comportamenti essenziali dei dati che si stanno memorizzando. Generalmente ogni modello viene mappato ad una singola tabella del database. Conseguentemente ciò che si è fatto è stato, a partire dalle relazioni individuate nello schema logico, realizzare quelli che potevano essere gli eventuali modelli corrispondenti. Tale scelta è stata animata dal desiderio di uniformarsi quanto più possibile al paradigma su cui si fonda Django, in modo tale da poter sfruttare l'enorme potere di tutte le sue componenti.

Sono inizialmente stati definiti un numero modelli pari alle relazioni individuate nello schema logico, ai quali sono stati dati gli stessi nomi. Ciascuno di tali modelli è stato definito come sottoclasse di *django.db.models.Model*. Per ciascun modello sono stati definiti una serie di campi, ovvero di attributi che fossero corrispondenti agli attributi delle diverse relazioni a cui si faceva riferimento. Ciascun campo del modello è stato definito come un'istanza della classe Field appropriata al tipo individuato per i corrispondenti attributi nelle varie relazioni, in questo modo si è potuto definire il tipo di ciascuna delle colonne delle tabelle del database, oltre che i requisiti minimi di convalida utilizzati dall'amministratore di Django e nei moduli generati automaticamente. Per ciascuno dei campi sono stati valorizzate una serie di opzioni sia tra quelle caratteristiche per lo specifico campo (ad esempio la max-length nel caso dei CharField) sia quelli comuni a tutti i campi, relativi alla possibilità di avere valori nulli o meno, alle chiavi primarie, ai valori di default. Sono stati implementati anche i vincoli di chiave esterna corredandoli con le politiche di reazione all'evento di cancellazione. Tutti questi passi sono stati svolti con l'intento di mantenere la consistenza rispetto alla definizione delle relazioni sullo schema logico.

L'unica nota di differenziazione rispetto a quest'ultimo è stato rappresentato dal fatto che per quelle relazioni che prevedevano delle chiavi primarie formate da due attributi si è proceduto all'utilizzo di un campo di chiave primaria automatico, nello specifico ciò è stato previsto per le relazioni *Appartenenza*, *Valorizzazione*, *Composizione*, *Prestazione*. In fase di definizione dei modelli sono stati forniti i metadati tramite la classe interna *Meta*. I metadati del modello sostanzialmente comprendono tutto ciò che non appartiene ai campi, si sono in questo modo chiariti gli eventuali attributi che insieme dovessero soddisfare il vincolo "unique", tramite l'espressione *unique_together*. Per tutti i modelli creati è stato ridefinito il metodo `__str__`, tale da aggiungere delle funzionalità personalizzate a livello di riga agli oggetti, nello specifico capace di restituire una rappresentazione di qualsiasi oggetto come una stringa.

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

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

View e Template

Nella definizione della logica dell'applicazione è stato fatto uso delle View che hanno rappresentato, durante la fase implementativa, la connessione tra le informazioni contenute all'interno dei modelli e le visualizzazioni da mettere a disposizione dell'utente all'interno dei Template. Le viste sono infatti delle specifiche funzioni Python che accettano una richiesta Web e restituiscono una risposta Web, che può essere il contenuto di una pagina. Data la stretta interconnessione che le viste e i relativi template hanno generalmente, e che hanno avuto anche in questa implementazione, si procederà ad una descrizione parallela, che consenta di comprenderne la reale importanza, apprezzando le finalità raggiunte. Risulta però fondamentale una prima premessa relativamente ad uno strumento messo a disposizione da Django e fortemente utilizzato per soddisfare i requisiti funzionali del portale web realizzato, ovvero i form. La necessità di utilizzare i form e di aderire alle potenzialità offerte da Django, tramite la classe form, accredita ancor di più la scelta di creare i modelli. Si procederà dunque ad una descrizione, per ciascuna delle viste create, alla descrizione delle sue funzionalità e delle interazioni di quest'ultima con i diversi template.

- **Homepage**

La vista Homepage prevede che all'aggiunta al path principale della richiesta effettuata da parte dell'utente di "\Homepage", venga effettuato il render del template "login.html". Nello specifico tale vista è stata creata sfruttando il meccanismo di autenticazione messo a

disposizione da Django. Avendo definito la vista come una *LoginView*, automaticamente, senza aver avuto necessità di implementare ex novo un form che consentisse l'autenticazione dell'utente, all'interno del template "login.html" si è potuto fare riferimento ad un form contenente i campi di username e password. Le potenzialità insite in tale meccanismo, si spingono in realtà ancora oltre, consentendo anche il meccanismo di controllo delle credenziali inserite ed eventualmente, in caso di correttezza, all'automatizzazione del meccanismo di autenticazione dell'utente. Tale template estende nello specifico il template "homepage.html", dunque l'utente verrà in sostanza reindirizzato alla homepage, contenente la sezione per l'autenticazione. Tramite questo meccanismo, sfruttando il campo *extra_content*, si è potuto fornire al template connesso alla vista un dizionario aggiuntivo addizionale ai contenuti di default. Nello specifico tale dizionario è stato formato da una serie di campi e di valori, prelevati in maniera dinamica dal database, con lo scopo di rendere le visualizzazioni di diversi valori e sezioni presenti nella homepage costantemente aggiornate. Si procede a prelevare dinamicamente il numero di immagini appartenenti a ciascuna delle macrocategorie, in maniera tale da avere la sezione riguardante i numeri delle immagini sottoposte al test delle reti neurali correttamente aggiornati. Riguardo alle reti neurali si è predisposto che venissero prelevati anche i nomi delle reti neurali e le relative descrizioni in maniera tale da popolare la sezione della homepage corrispondente. Riguardo alle macrocategorie si sono prelevati i nomi e le relative descrizioni, utilizzate in diversi punti della homepage. Tali valori, prelevati tramite delle query effettuate sul database, sono stati inseriti all'interno di liste popolate tramite il metodo *append()*. Le tre liste (*home_img*, *home_rn* e *home_macro*) sono state poi accedute, all'interno del template tramite l'utilizzo di indici. Il meccanismo di autenticazione messo a disposizione da Django ha consentito di condizionare la visualizzazione della Homepage a seconda di se l'utente fosse autenticato o meno. Nello specifico la sezione per effettuare l'accesso, unica componente del template di login, è visibile soltanto qualora l'utente non fosse autenticato, così come i tasti accedi e registrati, che in caso di utente autenticato vengono rimpiazzati dal tasto esci e dal logo utente. Il controllo sull'autenticazione dell'utente viene effettuato tramite il metodo *user.is_authenticated*. Infatti, Django utilizza sessioni e middleware per agganciare il sistema di autenticazione agli oggetti di richiesta. Questi forniscono un attributo *request.user* su ogni richiesta che rappresenta l'utente corrente. Se l'utente corrente non ha effettuato l'accesso, questo attributo verrà impostato su un'istanza di *AnonymousUser*, altrimenti sarà un'istanza di *User*. Sempre tramite una vista messa a disposizione da Django è stata gestita la disconnessione dell'utente, nello specifico tramite l'applicazione di una *LogoutView*.

- **Registrazione**

La vista Registrazione prevede che all'aggiunta al path principale della richiesta effettuata da parte dell'utente di "\Registrazione", venga effettuato il render del template "Registrazione.html".

All'interno della registrazione l'utente deve poter completare i campi richiesti e successivamente sottoscriverli. La necessità di utilizzare un form si è tramutato nella creazione di una classe *AccountForm* che estendesse la classe *UserCreationForm*. Quest'ultima viene messa a disposizione da Django per la realizzazione del form usato in fase di registrazione dell'utente. È stato necessario estenderla in quanto non prevedeva tutti i campi e dunque tutti gli input previsti in fase di progettazione, si è proceduto dunque ad aggiungere i campi di nome, cognome ed e-mail, mentre è stato riutilizzato il meccanismo,

già implementato in tale form, di validazione delle password, ovvero il confronto tra la password e la riconferma inserita dall'utente. Specificati dunque i campi di tale form, all'interno della vista si è proceduto alla creazione di un'istanza di tale form, conservata all'interno di una variabile. Ora qualora il metodo con cui venga effettuata la richiesta della pagina sia il metodo POST, sintomo che i campi sono stati compilati e sottomessi, la variabile form sarà popolata con le risposte inserite da parte dell'utente, ottenute tramite *AccountForm(request.POST)*. Si controlla la validità dei campi inseriti tramite il metodo *is_valid()* e si procede al salvataggio di tali campi all'interno del database, nella relazione Users, automaticamente configurata e collegata da Django stesso. Si procede successivamente all'autenticazione diretta dell'utente, senza che questo debba effettuarla da sé, tramite le funzioni *authenticate()* e *login()*, e si procede a reindirizzare l'utente alla homepage. Qualora invece il form non fosse stato sottomesso, dunque la richiesta fosse stata eseguita con il metodo GET oppure il form non fosse valido, l'utente visualizzerebbe la pagina di registrazione con il form ancora da compilare. Infatti, la variabile contenente il form, sia completato che incompleto, viene passata al template di registrazione, in cui si procede alla stampa dei suoi contenuti tramite la variabile *field*, contenente i campi appartenenti al form.

- **Test**

La vista Test prevede, a seconda del metodo con cui viene effettuata la richiesta, metodo GET o metodo POST, la gestione dei due template “test.html” e “result.html”. Bisogna sottolineare che prima della definizione di tale vista, è stato posto il decoratore *@login_required* con l'intento di limitare l'accesso a tali pagine web ai soli utenti autenticati, e di reindirizzare alla homepage eventuali altre richieste.

Tale vista serve sostanzialmente alla gestione del vero e proprio core della web application, infatti qui l'utente deve poter effettuare le sue scelte per visualizzarne poi i risultati. Poiché in fase implementativa si era scelto che l'utente selezionasse attraverso la homepage la macrocategoria su cui voler effettuare il test, è stato in prima battuta necessario gestire il meccanismo tale per cui il valore della macrocategoria cliccata da parte dell'utente sia passato alla vista tramite il metodo GET, inserendo cioè il nome della macrocategoria scelta all'interno dell'URL.

Al fine di consentire all'utente di caricare l'immagine, effettuare la valorizzazione delle categorie ad essa annesse e selezionare la rete neurale a cui far effettuare la classificazione, è stato necessario prevedere l'utilizzo dei form. Poiché in fase progettuale si era deciso che per le diverse tipologie di categorie, gli input all'interno dei form fossero di tipi diversi, e poiché la strutturazione dei modelli avrebbe richiesto che si facesse riferimento a più modelli contemporaneamente, si sono sostanzialmente create tre diverse tipologie di form per le categorie. Tale scelta ha preso corpo dall'idea di base di consentire, all'aggiunta di una nuova categoria da parte dell'amministratore, la scelta del tipo di input da mostrare all'utente in fase di valorizzazione. I tre form previsti, ovviamente espandibili, sono stati tali da consentire tre diverse tipologie di controlli: Select, Radio e l'input di tipo Number.

Ai form per la selezione delle macrocategorie sono stati aggiunti quelli per il caricamento dell'immagine, per la scelta della rete neurale e del threshold.

Si riportano di seguito i form implementati, che estendono tutti la classe *forms.ModelForm*.

ImagineForm: prevede un input di tipo file, limitato al caricamento delle sole immagini.

ClassificazioneForm: form destinato alla scelta della rete neurale e del valore di threshold da associare. La scelta della rete neurale viene effettuata tramite un controllo di tipo Select, all'interno del quale, come opzioni, vengono mostrate le reti neurali prelevate dal database, corrispondenti alla macrocategoria selezionata dall'utente. Il valore di threshold viene selezionato tramite un input di tipo range, con valori variabili tra 0 e 1. La creazione di tale form prevede che venga passato il nome della categoria su cui si sta effettuando il test, in modo tale che tra le reti neurali mostrate all'utente vi siano solo quelle corrispondenti alla macrocategoria in questione.

ValorizzazioneNumberForm, ValorizzazioneSelectForm, ValorizzazioneRadioForm: sono i tre form utilizzati per la gestione delle valorizzazioni delle categorie. Così come è intuibile dal nome, generano un modello di form contenente rispettivamente un input di tipo Number, una Select o un Radio. Il metodo per la creazione di tali form è stato sovrascritto, prevedendo che, in fase di creazione, dovesse essere passato come parametro in ingresso la categoria di riferimento. La conoscenza di tale categoria è fondamentale sia per categorie con valorizzazioni numeriche che non. Qualora la categoria fosse di tipo numerico, e si facesse dunque riferimento al ValorizzazioneNumberForm, il nome della categoria sarebbe fondamentale per prelevare dal database il valore dell'attributo *Int_Max* associato e individuare dunque il massimo valore accettabile. Qualora la categoria non fosse di tipo numerico, il nome di questa sarebbe fondamentale in quanto, avendo creato dei modelli contenenti le opzioni disponibili per ciascuna categoria, che avessero lo stesso nome delle categorie, si possono tranquillamente prelevare dinamicamente le opzioni da mostrare all'utente dal database.

La vista prevede un'esecuzione differente a seconda del metodo con cui viene effettuata la richiesta.

Se la richiesta è di tipo GET ed è stato selezionato un valore per la macrocategoria, allora si preleva il nome della macrocategoria selezionata dall'URL, si prelevano dal database tutte le categorie associate a quella macrocategoria e per ciascuna di esse si preleva l'attributo *Tipo*. Tale attributo avrà un valore corrispondente al tipo di controllo che si desidera per quella categoria, ovvero "Select", "Number" o "Radio". Come si visualizza nel codice di seguito riportato, tutte le associazioni tra la macrocategoria selezionata e le categorie relative saranno memorizzate all'interno del set *comp_set*. Iterando su quest'ultimo, verranno prelevate per ciascuna categoria il nome e il tipo, si procederà alla creazione di istanze dei form corrispondenti, conservati all'interno di *form_set*, per essere poi passati al template "test.html" ed essere stampati, al fine di essere compilati da parte dell'utente.

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

Figura 6. Codice della vista test per richieste di tipo GET

Qualora il metodo della richiesta fosse il metodo POST, significherebbe che il form è stato compilato e inviato. Verrebbero prelevati i valori dell'immagine caricata, della rete neurale e del threshold scelto e tramite un meccanismo simile al codice sopra riportato verrebbero recuperati anche i valori espressi dall'utente in relazione alle diverse categorie, come si vede nel codice sottostante, dove *valore_v* sta ad indicare il contenuto della valorizzazione effettuata dall'utente sulla categoria.

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

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

Figura 7. Codice della vista test per richieste di tipo POST

Si procede successivamente al salvataggio dei valori all'interno del portale in accordo alle specifiche. All'interno della medesima transazione, a seconda della rete neurale selezionata, viene ricavato, per ciascuna categoria, il valore espresso da quest'ultima, viene dunque salvata la corrispondente istanza di classificazione effettuata dalla rete neurale, viene memorizzata l'immagine (nel caso non sia già presente) e l'appartenenza alla macrocategoria di riferimento (nel caso non sia già presente). Per ciascun elemento presente in *form_set* viene creata un'istanza di *Valorizzazione* all'interno del database. In seguito al salvataggio di questi valori, tali informazioni, con in aggiunta il numero di classificazioni effettuate dalla rete neurale, saranno passati all'interno di un dizionario al template per la visualizzazione del risultato, dove saranno mostrati all'utente in modo tale da rispettare le specifiche iniziali.

3.2. Implementazione del front-end

La fase di implementazione del front-end ha previsto l'utilizzo del framework Bootstrap, ovvero dei modelli di progettazione basati su HTML, CSS e le estensioni JavaScript messe da quest'ultimo a disposizione, al fine di realizzare delle pagine web che fossero responsive, dunque adattabili a diverse visualizzazioni. Il contributo offerto ha sostanzialmente riguardato l'utilizzo di alcuni template grafici a cui ispirarsi e una serie di componenti con funzionalità proprie riusabili, da poter essere integrati nella struttura progettata. Per la realizzazione di interfacce che fossero effettivamente conformi agli obiettivi prefissati è stata talvolta necessaria l'integrazione di funzioni JavaScript, finalizzate alla messa appunto di alcuni controlli, quali per esempio la validazione dell'input in fase di registrazione.

3.2.1. Caratteristiche comuni tra le interfacce

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

All'interno del footer vi sono invece informazioni quali i contatti dei membri del team sviluppatore del portale. La homepage si discosta dalle altre pagine in quanto, nel caso di un visitatore non autenticato, prevede nel footer una sezione per effettuare l'accesso. Presenta invece un header, formato dal logo del portale web, da una barra di navigazione per consultare agevolmente le varie sezioni della pagina, e dai pulsanti per la gestione dei meccanismi di login e di logout.

L'header di ciascuna pagina è stata realizzata in modo tale da essere sticky, semplificando la navigazione nel sito.

Per ciascuna di tali pagine, tramite una serie di ridimensionamenti dei diversi componenti, specificati all'intero dei file CSS collegati e realizzati tramite l'applicazione delle *media queries*, ci si è assicurati che fossero capaci di adattarsi a qualsiasi tipologia di visualizzazione: computer, tablet, cellulari di varie dimensioni.

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

3.2.2. Implementazione della Homepage

La homepage è stata strutturata e organizzata in diverse sezioni.

- Home: è la sezione che viene visualizzata al momento di caricamento della pagina. Prevede una riga, ripartita in più colonne, all'interno delle quali sono contenute delle *icon-box*, ovvero sostanzialmente dei box cliccabili tramite i quali l'utente autenticato può effettuare la scelta della macrocategoria su cui effettuare il test.
- Info: è la sezione all'interno della quale l'utente può reperire informazioni sul portale. È stata organizzata tramite l'impiego della struttura offerta dal container, all'interno del quale, per separare le immagini dai componenti testuali è stata usata una ripartizione tramite la componente di layout *grid* messa a disposizione da Bootstrap.
- Reti Neurali: è la sezione all'interno della quale l'utente può consultare i nomi e le descrizioni delle reti neurali disponibili. Anche queste, contenute all'interno di un container, sono state organizzate in questa disposizione, ripartendo una riga su più colonne. Ciascuno di tali box è un icon-box.
- Portfolio: contiene immagini di esempio fornite all'utente come guida, sulle quali è possibile effettuare una selezione. Ciascuna di tali immagini è organizzata all'interno di colonne di dimensioni differenti, adattabili alla dimensione stessa della pagina.

- **Team:** all'interno di tale sezione si possono conoscere i membri del team, ovvero i loro nomi e eventuali contatti utili. Le diverse *card* sono organizzate su più colonne, allineate all'interno della medesima riga.
- **Contatti:** è la sezione contenente i contatti utili per l'utente e la sede del team. La visualizzazione della mappa è resa possibile tramite l'utilizzo di un contenuto di tipo *frame*. Vi è in questa sezione la possibilità di inviare un'e-mail in automatico all'indirizzo di posta specificato, cliccando sul relativo bottone.

3.2.3. Implementazione della pagina di Registrazione

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

La colonna centrale è infatti organizzata in modo da prevedere una sezione titolo, una sezione formata dai vari campi di input e una sezione contenente i tasti "Registrati" e "Reset". Tali sezioni sono contenute all'interno di un *container* delimitato dallo spazio circostante da un bordo arrotondato di colore giallo, con uno sfondo parzialmente trasparente di colore nero.

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

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

3.2.4. Implementazione della pagina per effettuare il test e visualizzare il risultato

Tali due visualizzazioni hanno una struttura estremamente somigliante, anche nella sezione centrale. Orizzontalmente sono state strutturate su righe successive. La prima sezione prevede il nome della macrocategoria scelta, la sezione subito sottostante prevede gli effettivi contenuti per la compilazione dei campi o la visualizzazione dei risultati, l'ultima sezione prevede nel caso del test i due tasti per caricare l'immagine e per sottomettere i campi inseriti, mentre nel caso di visualizzazione dei risultati è previsto il tasto per il caricamento di una nuova immagine per la classificazione, sulla medesima macrocategoria.

L'area centrale risulta essere formata da una *card*, organizzata in modo tale che la sezione di sinistra preveda la visualizzazione dell'immagine da caricare o già caricata, mentre la sezione di destra preveda delle strutture per l'inserimento dei valori o la visualizzazione dei risultati.

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

Nella pagina in cui vengono visualizzati i risultati vi sono tre dei box precedentemente descritti, a cui sono associate delle sezioni dotate del medesimo meccanismo, una associata alla visualizzazione delle risposte dell'utente, una della rete neurale e una per la visualizzazione di informazioni inerenti alla rete neurale, quali l'indice di performance medio per ciascuna categoria, il threshold selezionato in fase di caricamento e il numero di classificazioni fino a quel momento effettuate dalla rete neurale. Tutte e tre queste sezioni sono al loro interno organizzate in maniera tabellare, prevedendo un header della tabella che renda auto-esplicativa la consultazione dei dati e prevedendo ciascun controllo all'interno di una riga. In caso di aumento dei campi è stata predisposta una scroll-bar che ne consenta la visualizzazione, evitando modifiche nell'assetto della grafica.

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

- *readURL*: Tale funzione consente all'utente di poter visualizzare, nella sezione apposita, l'immagine che ha appena caricato, prima si sottomettere i campi, utilizzando i meccanismi di navigazione nel DOM.
- *loading*: Tale funzione consente di intrattenere l'utente quando, dopo la sottomissione del form e prima della visualizzazione dei risultati, bisogna attendere che la rete neurale effettui la propria elaborazione. Si riesce in questo modo, tramite i messaggi "Aspetta ancora un po..." e "La rete neurale sta caricando...", a comunicare all'utente di attendere non facendo apparire una situazione altrimenti assimilabile ad un blocco del sistema.

4. Integrazione dei vari moduli, Visualizzazione e Descrizione del risultato finale

La fase conclusiva ha riguardato l'integrazione dei vari moduli precedentemente implementati. Dunque, è stato necessario verificare il corretto funzionamento del prodotto ottenuto, sottoponendolo alle condizioni di funzionamento richieste in fase di analisi dei requisiti. Durante tale fase è stato necessario procedere all'integrazione, all'interno del progetto realizzato, delle reti neurali fornite, facendo in modo da instaurare una comunicazione tra la vista che consente all'utente la visualizzazione dei risultati e le funzioni che, ciascuna per ogni rete neurale, consentono la valutazione del risultato richiesto. Si è proceduto dunque alla validazione di un prodotto che risulta essere funzionante, dalla grafica accattivante e in accordo con i requisiti funzionali. Si è cercato di realizzare un'esperienza di navigazione all'interno del portale che fosse piacevole e non banale, anche per chi ne volesse cogliere il solo aspetto "ludico". Anche i requisiti non funzionali possono dirsi quasi completamente soddisfatti. Infatti, fermo restando la natura prototipale del prodotto realizzato, l'impronta strutturale data, estremamente modulare, riusabile ed espandibile, consente in ogni modo la possibilità di arricchimento e di estensione del prodotto in maniera estremamente semplice.

4.1. Visualizzazione e descrizione del risultato

Si procede alla descrizione del risultato finale ottenuto, ovvero delle diverse pagine web tra cui l'utente ha la possibilità di navigare.

4.1.1. Visualizzazione Homepage

- **Header**

L'header della homepage presenta una visualizzazione differenziata per l'utente autenticato e quello non autenticato. Entrambi visualizzeranno il logo del portale e una barra di navigazione che consentirà la consultazione delle varie sezioni all'interno della pagina. Mentre l'utente non autenticato visualizzerà i pulsanti per effettuare l'accesso o essere reindirizzato alla pagina di registrazione, l'utente autenticato visualizzerà il pulsante per effettuare la disconnessione, insieme ad un'icona profilo che gli darà il benvenuto.



Figura 8. Header per utente non autenticato

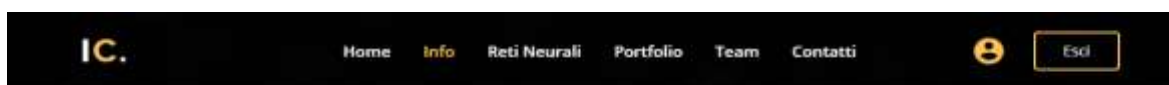


Figura 9. Header per utente autenticato

- **Footer**

Anche il footer presenta una diversa visualizzazione per le due tipologie di utenti. Qualora l'utente non avesse effettuato l'autenticazione, all'interno del footer sarebbe visualizzabile, oltre alle informazioni sui contatti utili, una sezione per poter effettuare l'accesso che provvederebbe a notificare eventuali errori nell'inserimento dei dati all'utente.

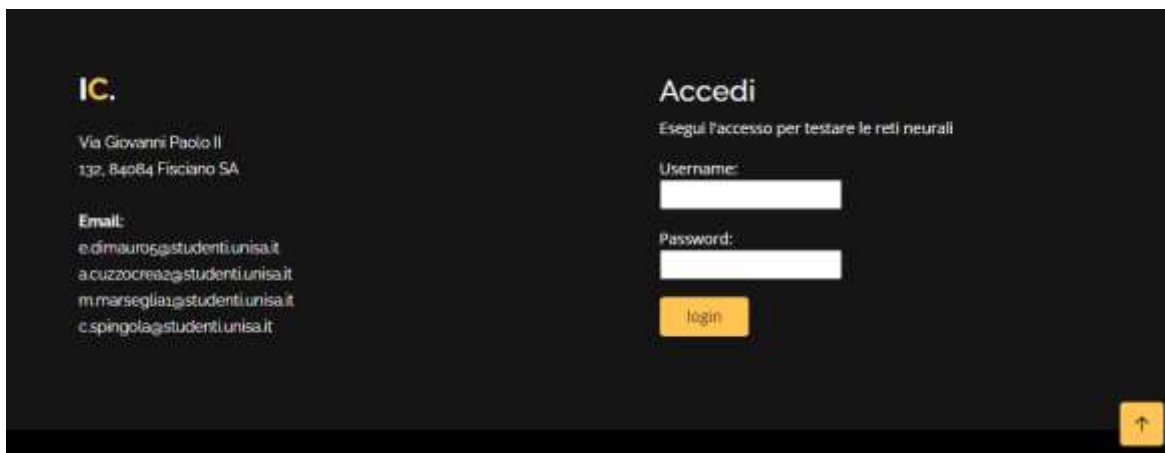


Figura 10. Footer per utente non autenticato.

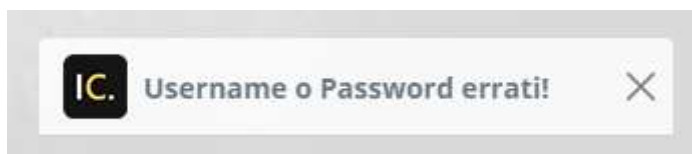


Figura 11. Messaggio di errore in caso di fallimento dell'autenticazione



Figura 12. Footer per utente autenticato

- **Home**

Alla voce "Home" l'utente visualizzerà, con uno sfondo accattivante ed esplicativo dello scopo principale del portale, ovvero mettere a confronto le reti neurali artificiali e naturali, le quattro macrocategorie: Face Counting, People Counting, Vehicle Counting e Face Analysis. I nomi di tali macrocategorie sono inseriti all'interno di box interattivi e cliccabili,

corredati da icone ad esse riferite. Da tale visualizzazione, qualora fosse autenticato, l'utente può effettuare la scelta della macrocategoria e iniziare il test.



Figura 13. Home

- **Info**

All'interno di tale sezione l'utente potrà reperire una serie di informazioni utili in merito alle funzionalità messe a disposizione dal portale.

Nello specifico la possibilità di, previa autenticazione, caricare un'immagine, specificarne le caratteristiche, selezionare una rete neurale e visualizzare il confronto tra la classificazione effettuata su quell'immagine da quest'ultima e quella effettuata da sé stesso.



Figura 14. Info(a)



Figura 15. Info(b)

- **Reti Neurali**

In tale sezione l'nome ed una breve descrizione delle reti neurali di cui il portale dispone e che eventualmente potrà selezionare in fase di caricamento dell'immagine, in modo tale da conoscerne la funzionalità.

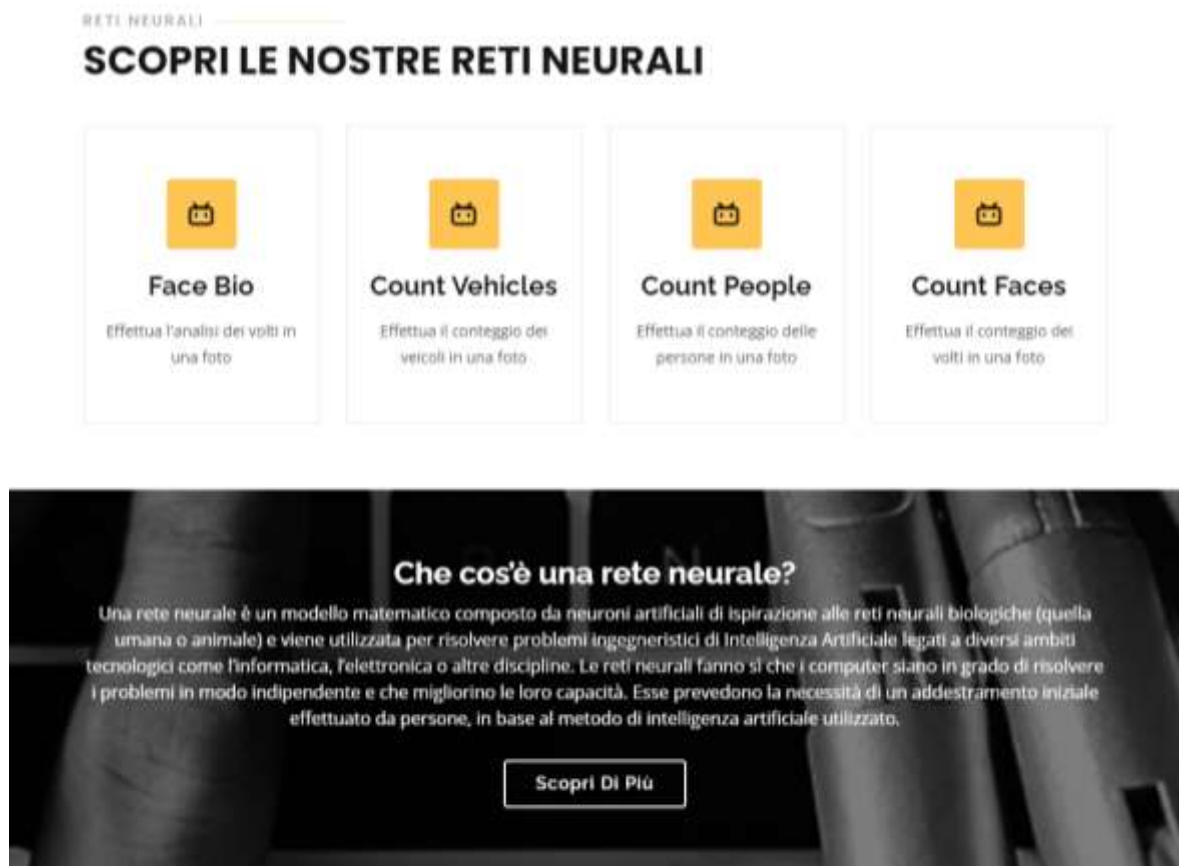


Figura 16. Reti Neurali

- **Portfolio**

All'interno di tale sezione l'utente potrà visualizzare una serie di immagini di esempio di quelle eventualmente caricabili all'interno del database. Dunque immagini contenenti dei volti, appartenenti alle macrocategorie Face Analysis e Face Counting, immagini contenenti gruppi di persone appartenenti alla macrocategoria People Counting ed infine immagini contenenti un numero variabile di veicoli appartenenti alla macrocategoria Vehicle Counting. Tramite un filtro predisposto potrà scegliere se visualizzare tutte le immagini o solo quelle contenenti uno specifico soggetto.



Figura 17. Portfolio



Figura 18. Numero immagini classificate per ciascuna macrocategoria



Figura 19. Descrizione di ciascuna macrocategoria e visualizzazione delle relative categorie

- **Team**

In tale sezione si potranno conoscere i membri del team amministratore del portale, e reperire contatti utili per i social network più diffusi quali LinkedIn, Facebook, Twitter e Instagram.

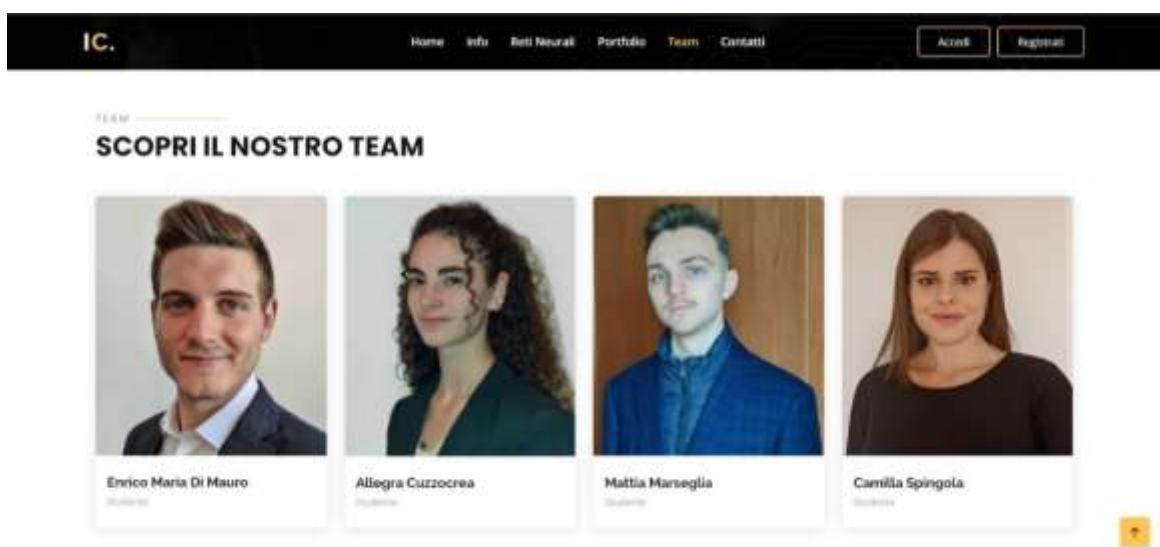


Figura 20. Team

- **Contatti**

In tale sezione è possibile reperire contatti utili a cui inviare automaticamente l'e-mail e eventualmente conoscere la sede del team.

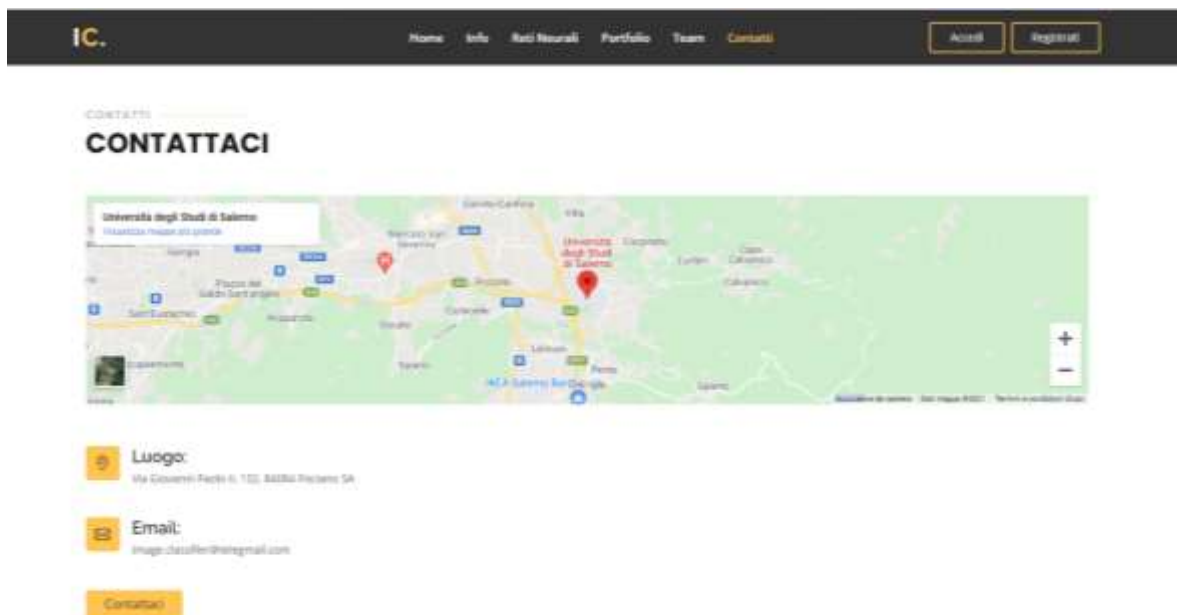


Figura 21. Sede

4.1.2. Visualizzazione pagina di Registrazione

La pagina di registrazione riporta il form contenente i campi richiesti per potersi registrare al sito. Cliccando sul pulsante Registrati senza aver compilato tutti i campi verrà notificato un messaggio di errore tramite un popup. In caso di corretto inserimento di tutti i dati si verrà reindirizzati alla pagina principale con la possibilità di selezionare le macrocategorie. Cliccando sul pulsante reset vengono svuotati tutti i campi.



Figura 22. Registrazione



Figura 23. Progress-bar (Weak)

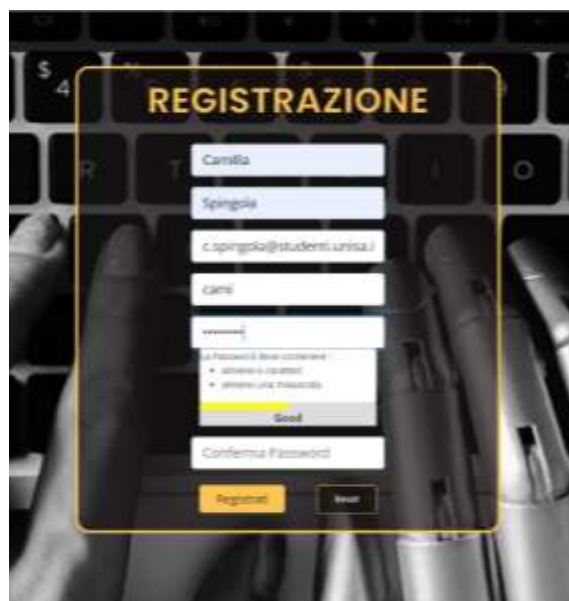


Figura 24. Progress-bar (Good)

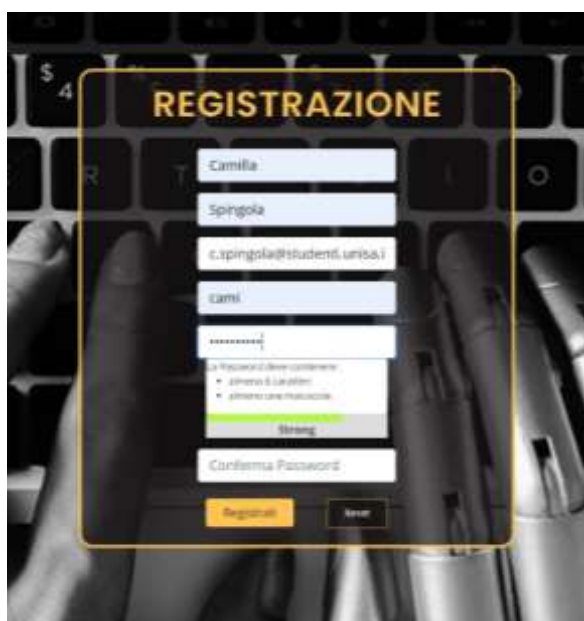


Figura 25. Progress-bar (Strong)

Nella pagina è stata inserita anche una progress-bar indicativa della robustezza della password inserita dall'utente. Nello specifico lì dove i requisiti minimi non fossero soddisfatti il riempimento della barra sarebbe limitato al solo 25% con il colore rosso e corredato dalla scritta "Weak", in caso di soddisfacimento il riempimento sarebbe portato al 50% con il colore giallo e corredato dalla scritta "Good", qualora fosse invece rilevato il soddisfacimento di altri prerequisiti non obbligatori ma indici di una maggiore robustezza come la presenza di un almeno un carattere speciale o di un numero oppure di entrambi, il riempimento sarebbe portato rispettivamente al 75% con il colore verde chiaro e corredato dalla scritta "Strong", o al 100% con il colore verde scuro e la scritta "Very Strong".

4.1.3. Visualizzazione pagina di Test

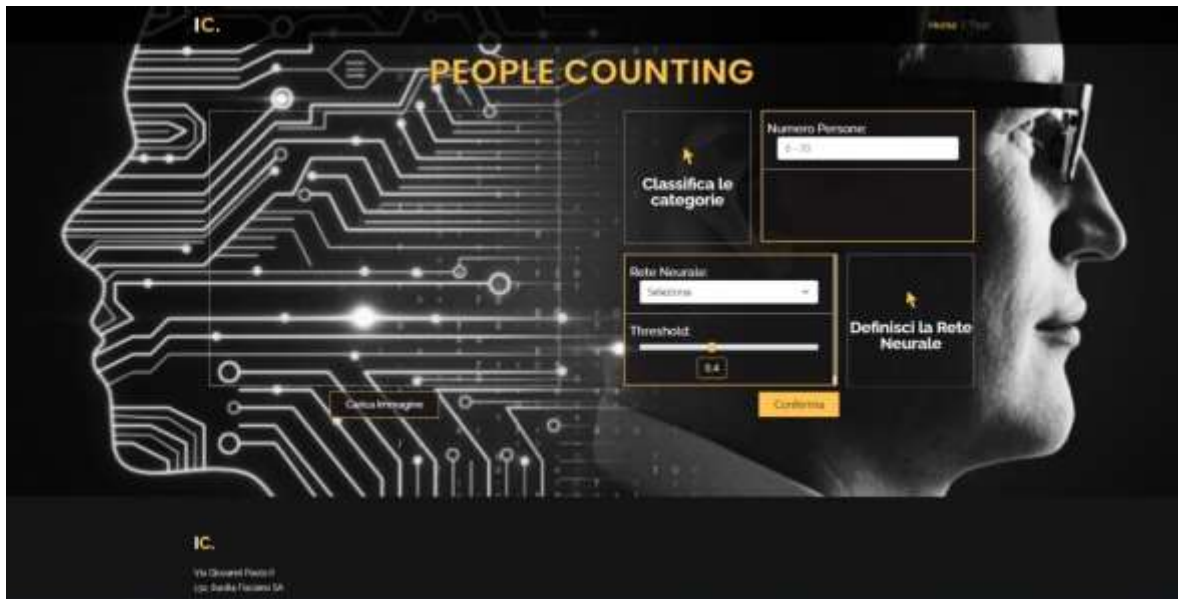


Figura 26. Pagina di test prima del completamento dei campi



Figura 27. Pagina di test dopo il completamento dei campi

In tale pagina l'utente potrà eseguire il test, dopo aver selezionato, all'interno della homepage la macrocategoria. Avrà la necessità, per poter procedere alla conferma e quindi alla visualizzazione del risultato di aver inserito tutti i campi richiesti, eccetto il threshold, già impostato ad un valore di default di 0.4.

In seguito alla sottomissione dei campi inseriti, cliccando il tasto "Conferma", in attesa della valutazione da parte della rete neurale, l'utente verrà intrattenuto da scritte dinamiche.



Figura 28. Pagina di test dopo aver cliccato il tasto di conferma, in attesa del risultato della rete neurale

4.1.4. Visualizzazione pagina di Result



Figura 29. Pagina di result

All'interno di tale pagina l'utente potrà consultare l'esito della classificazione effettuata dalla rete neurale comparandola con la propria valorizzazione. Saranno mostrati i valori delle performance della rete neurale per le varie categorie classificate, aggiornate alle ultime classificazioni, il valore di threshold selezionato e il numero di classificazioni fino a quel momento effettuate dalla rete neurale. L'utente potrà effettuare un nuovo test sulla medesima classificazione cliccando sul pulsante "Carica Nuova Immagine".

Conclusioni

Il prodotto finale inquadrabile nel panorama delle web application consente la classificazione delle immagini organizzate in macrocategorie e la comparazione con gli esiti di un processo che, coinvolgendo le reti neurali, si configura come stimolo a programmare implementazioni e a ipotizzare possibili scenari di applicazione futura. In particolare, il progetto, condotto in team, ha previsto una verticalizzazione delle specifiche realizzative e ha accolto il mio contributo in relazione alla macrocategoria People Counting, orientando ogni fase della ricerca alla realizzazione di un prodotto riusabile ed espandibile sia in fase di progettazione che di implementazione. La consapevolezza che l'innovazione tecnologica richiede una fruizione sempre più diretta da parte dell'utente ha orientato le scelte personali verso una ricerca che garantisse efficienza ma anche un'interazione piacevole e spendibile. L'aspetto più apprezzabile, sia in fase di processo che di realizzazione del prodotto, si è rivelato essere l'uso tecnologie innovative e diffuse che consentono un coinvolgimento emotivo e professionale, stimolando il desiderio di ricerca in un settore estremamente affascinante e denso di opportunità per ulteriori approfondimenti.

Bibliografia e Sitografia

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

Web Languages. URL: <https://www.w3schools.com/>.

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

PgAdmin 4 documentation. URL: <https://www.pgadmin.org/docs/pgadmin4/4.25/index.html>.

Django documentation. URL: <https://docs.djangoproject.com/en/3.2/>.

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

Ringraziamenti

Al termine di questo percorso esprimo la mia gratitudine al Prof. Antonio Greco e alla Prof.ssa Alessia Saggese per la competenza e la professionalità che da sempre offrono ai loro studenti, per il supplemento di disponibilità che mi hanno riservato per la realizzazione di questo lavoro e per l'arricchimento della mia formazione attraverso un percorso di tirocinio che si è rivelato stimolante e impreziosito dalla valida collaborazione dei miei colleghi Allegra, Enrico e Mattia.

Grazie ai miei genitori per aver saputo tenermi la mano senza condizionarmi mai, per avermi insegnato la gioia di un successo ottenuto dopo il sacrificio, ma soprattutto la forza per ricostruire dopo un fallimento. Grazie a mio fratello Valerio per l'affetto da sempre destinatomi, per essere sempre stato un riferimento indispensabile su cui poter contare. Grazie alla mia famiglia, dunque, per me porto sicuro, luogo di confidenze, di fiducia e di amore incondizionato.

Grazie ai miei nonni, a chi oggi avrei voluto fosse ancora qui, a festeggiare con orgoglio il mio traguardo.

Grazie a Mattia, compagno di studi e di tanto altro ancora, per aver condiviso con me i momenti più belli e i più difficili, per esserci stato in questo tempo di fatica e di sogni.

Agli amici Benedetto, Stefano e Antonio, la mia riconoscenza per aver proseguito con me un percorso iniziato tra i banchi di scuola e ai nuovi compagni di studio, Ferdinando e Vito, il mio grazie per aver riempito, seppure attraverso uno schermo, le mie giornate di questo tempo strano.

Infine, alle "Alessie", amiche da sempre, la mia riconoscenza per aver tollerato le mie assenze e le mie distrazioni, lasciando inalterato l'enorme affetto che ci lega.