



**Politecnico
di Torino**

Politecnico di Torino

Ingegneria Informatica

A.a. 2022/2023

Sessione di laurea Dicembre 2023

Estensione di una piattaforma web di e-voting tramite supporto OCR

Relatori:

Luigi De Russis

Mirko Landolfo

Candidato:

Vittorio Magelli

Ringraziamenti

Desidero dedicare questa tesi a mio papà che purtroppo non c'è più e se è vero che mi guarda spero sia orgoglioso di me per aver concluso questo percorso di studi e soprattutto per l'uomo che sto diventando.

Ringrazio la mia famiglia ed in particolare mia mamma e i miei fratelli Matteo e Benedetta per essermi sempre stati accanto e avermi aiutato nei momenti difficili.

Grazie ai miei amici che riescono ad alleggerire il peso della quotidianità e i problemi con la loro simpatia. In particolare grazie ad Alberto, Davide e Stefano, mi sono sempre stati vicino e spero continuino a farlo.

Ringrazio l'azienda Pro Logic Informatica per la fiducia che mi ha dato, un grazie in particolare a Mirko Landolfo per l'aiuto fornitomi durante lo sviluppo di questa tesi e ai miei colleghi tutti per avermi fatto sentire a mio agio dal primo giorno.

Infine grazie a mio nonno Paolo, che nonostante l'età è ancora il punto di riferimento mio e di ogni membro della mia famiglia. Mi ha insegnato i valori dell'impegno, del sacrificio e della famiglia e spero di poter essere un giorno per i miei figli e i miei nipoti quello che lui è per noi.

Alla mia famiglia

Indice

Elenco delle figure	VII
1 Introduzione	1
1.1 Contesto	1
1.2 Goal	2
1.3 Struttura della tesi	2
2 Studio del dominio	4
2.1 Tipologie di elezione	5
2.1.1 Europee	5
2.1.2 Politiche	6
2.1.3 Regionali	7
2.1.4 Comunali	8
2.1.5 Referendum	8
2.2 Raccolta dati in Eleweb	9
2.2.1 Affluenze	10
2.2.2 Raggruppamenti	10
2.2.3 Liste	11
2.2.4 Preferenze	11
3 Progettazione del sistema	13
3.1 Studio fonogrammi	13
3.2 Training	16
3.3 OCR	17
3.4 Backend	17
4 Progettazione Training	19
4.1 Tecnologie utilizzate	19
4.1.1 PHP	19
4.1.2 Canvas	19
4.2 Logica del training	20

4.2.1	Funzione di training	20
4.3	La costruzione dei rettangoli	24
4.3.1	Le funzioni mousemove, mouseup e mousedown	25
4.4	Operazioni concesse	26
4.4.1	Esecuzione training	27
4.4.2	Caricamento training	29
4.4.3	Annullamento training	29
5	Progettazione OCR	30
5.1	Tecnologie utilizzate	30
5.1.1	Jscanify	30
5.1.2	Scelta della tecnologia	31
5.1.3	Doctr	31
5.2	Funzionamento OCR	33
5.2.1	Il problema delle dimensioni	33
5.2.2	L'algoritmo di confronto	37
5.2.3	Il rilevamento degli errori	39
6	Prototipo Training	41
6.1	Cruscotto dei sub-eventi	41
6.1.1	Affluenze	42
6.1.2	Liste	43
6.1.3	Raggruppamenti	43
6.1.4	Preferenze	44
6.2	L'inizio del training	45
6.3	La visualizzazione delle istruzioni	48
6.3.1	Visualizzazione tramite alert	48
6.3.2	Visualizzazione tramite label	49
6.3.3	Visualizzazione tramite elenco laterale	49
6.4	La forma del cursore	50
7	Prototipo OCR	54
7.1	Pulsante attivazione OCR	54
7.2	Lo scanning dell'immagine	56
7.3	L'inserimento automatico dei dati	58
8	Valutazione	60
8.1	Pianificazione	60
8.2	Attività	61
8.3	Svolgimento e Risultati	62
8.3.1	Possibili modifiche	63

9	Conclusioni	66
9.1	Sviluppi futuri	67
9.2	Best practices	68
A	Script per la valutazione del prototipo	69
A.1	Presentazione	69
A.2	Istruzioni	70
A.3	Attività	70
A.3.1	Attività 1: Iniziare un training	70
A.3.2	Attività 2: Eseguire un training	71
A.3.3	Attività 3: Salvare il training	71
A.3.4	Attività 4: Caricare un training	71
A.3.5	Attività 5: Annullare il training	71
A.3.6	Attività 6: Iniziare una scansione OCR	71
A.3.7	Attività 7: Verifica inserimento dati	72
A.4	Debriefing	72
B	Questionari SUS relativi al training	73
	Bibliografia	79

Elenco delle figure

3.1	Confronto affluenza politiche tra due comuni	14
3.2	Confronto foglietto liste tra due comuni	15
3.3	Diagramma e-r backend	18
4.1	Strutture all'interno di trainingData con inizializzazione dei campi .	22
4.2	Flow chart funzione training	23
4.3	Coordinate non normalizzate	24
4.4	Coordinate normalizzate	25
5.1	Esempio fasi elaborazione OCR	32
5.2	Confronto training con fonogramma più ambiente esterno	34
5.3	Confronto training con fonogramma tagliato	35
5.4	Confronto tra la foto originale e quella scannerizzata	36
5.5	Esempio sfalsatura training rispetto a immagine ocr	38
6.1	Fieldset pulsanti sub-eventi	42
6.2	Affluenze parziali	42
6.3	Affluenze finali	43
6.4	Liste	43
6.5	Raggruppamenti	44
6.6	Preferenza con training esistente	44
6.7	Preferenza senza training esistente	45
6.8	Alert errore inserimento immagine	45
6.9	Effetto dei bottoni zoom-in e zoom-out sul canvas	46
6.10	Alert istruzioni training	47
6.11	Esempio di annullamento training	48
6.12	Istruzioni affluenze finali	50
6.13	Esempio di tooltip	52
6.14	I puntatori nell'applicativo	53
7.1	Esempio pagina sub-evento web application	55
7.2	Alert sezione non inserita	55

7.3	Esempio di overlay per il sub evento raggruppamenti	57
7.4	Alert di attesa	58
7.5	Esempio inserimento dati via OCR	59

Capitolo 1

Introduzione

1.1 Contesto

L'articolo 48 della Costituzione Italiana specifica che "il voto è personale ed uguale, libero e segreto, Il suo esercizio è un dovere civico". Inoltre anche nel primo e terzo articolo della Costituzione troviamo un riferimento al diritto di voto, in cui si afferma che "la sovranità appartiene al popolo, che lo esercita nelle forme e nei limiti della Costituzione".

Il fatto che il voto in Italia sia regolato per lo più da leggi costituzionali fa intuire quanto sia importante, e quanto sia fondamentale che i dati vengano raccolti in maniera corretta e comunicati al ministero nella maniera più sicura e precisa possibile.

E' in questo contesto che nasce Eleweb, una piattaforma di e-voting e per la raccolta di dati scrutinati dai presidenti di seggio[1] sviluppata dall'azienda Pro Logic Informatica s.r.l. che integra l'interfaccia con il Ministero dell'interno, contenente algoritmi di calcolo e verifica della corretta trasmissione dei dati elettorali. Eleweb permette di effettuare la raccolta dati scrutinati dai presidenti di seggio e gestisce ogni tipologia di evento elettorale a livello Nazionale (Camera e Senato o Europee) e Locale (Regionali, Provinciali e Comunali)[2]. Durante le elezioni offre supporto per l'analisi organizzativa e la raccolta dati sul territorio.

E' possibile inserire dati riguardanti sub-eventi ovvero le affluenze, i raggruppamenti, le liste ed esprimere le preferenze (tutti questi aspetti verranno approfonditi nel capitolo 2).

In questo lavoro di tesi è stato dunque ideato un modulo per il riconoscimento OCR (Optical Character Recognition) delle informazioni inserite nelle "Comunicazioni del seggio agli uffici elettorali", per velocizzare la raccolta e trasmissione dei dati, effettuando una semplice foto alle comunicazioni per inserire le informazioni all'interno di Eleweb.

1.2 Obiettivo

Partendo dai presupposti introdotti nella sezione Contesto l'obiettivo di questa tesi è l'integrazione di un modulo di scansione OCR all'interno di Eleweb, applicando metodi e metodologie di ultima generazione di User Experience e User Interface(UX/UI).

Nello specifico si presterà particolare attenzione alla qualità con cui il modulo OCR riconoscerà i caratteri, gestendo le difficoltà nel loro riconoscimento e cercando di fornire informazioni all'utente della qualità con cui le informazioni sono state ritrovate.

Il modulo OCR sarà diviso in due parti:la prima sarà all'interno di Eleweb e consisterà in una fase di configurazione. La seconda invece sarà l'implementazione vera e propria del riconoscimento OCR sulla web app perché i comuni necessitano di fotografare i fonogrammi tramite tablet o cellulare per velocizzare l'inserimento.

Per quanto riguarda la prima parte sarà importante studiare l'interfaccia e rendere le operazioni di configurazione facili,veloci e intuitive per l'utente. Per la seconda parte sarà invece necessario trovare tra i numerosi scanner OCR open source, un tool che sia il giusto compromesso tra velocità e qualità di riconoscimento.

1.3 Struttura della tesi

La tesi inizia con una breve introduzione ad Eleweb, una descrizione generale delle varie tipologie di elezione in Italia e sul come vengono gestite all'interno dell'applicativo. Successivamente vengono descritte le implementazioni del training e del modulo OCR vero e proprio all'interno della web application, per poi passare allo sviluppo dei prototipi. Infine sono stati svolti dei test di usabilità agli Amministratori Comunali, per la valutazione del prototipo.

- Nel capitolo 2 "Studio del dominio" vi è una panoramica generale delle elezioni in Italia, è stato descritto per ogni tipologia di elezione quali dati sono raccolti, inoltre è stato descritto come Eleweb raccoglie e divide i dati.
- Nel capitolo 3 "Progettazione del sistema" viene descritto cos'è un fonogramma e perché si è optato per l'utilizzo di un training prima dell'inizio dell'elezione, come è stato ideato il database e com'è strutturato il modulo OCR.
- Nel capitolo 4 "Progettazione Training" vengono riportate le tecnologie utilizzate e la logica dietro al training, quali operazioni sono concesse e come sono sviluppate.
- Nel capitolo 5 "Progettazione OCR" vengono elencate le tecnologie usate nella web application e nell'OCR,inoltre sono riportate le varie problematiche

relative allo sviluppo del modulo, e viene spiegato come interagire con esso e come è stato sviluppato.

- Nel capitolo 6 "Prototipo Training" viene spiegato com'è stato progettato il prototipo del training, con particolare attenzione alle scelte legate all'UX/UI. Viene fatto ampio utilizzo di figure dell'applicativo per mostrare il risultato ottenuto.
- Nel capitolo 7 "Prototipo OCR" viene spiegato come è stato progettato il prototipo dell'OCR, come nel capitolo 6 viene posta particolare attenzione alle scelte legate all'UX/UI e vengono utilizzati numerosi esempi grafici.
- Nel capitolo 8 "Valutazione" si descriverà come è stato eseguito il test di valutazione e usabilità del prototipo, analizzando le metodologie, i partecipanti al test e i risultati ottenuti
- Nel capitolo 9 "Conclusioni" vengono discussi i risultati ottenuti, descritte le modifiche che verranno apportate all'applicativo e vengono definite alcune best practices che i comuni dovrebbero adottare per sfruttare a meglio il prodotto realizzato.

Capitolo 2

Studio del dominio

Per riuscire a sviluppare correttamente il modulo OCR è fondamentale strutturare un'interfaccia flessibile che possa adattarsi a ogni tipo di elezione, in quanto a seconda dell'evento elettorale vengono raccolte diverse quantità e tipologie di dati. Sulla base del tipo di elezione Eleweb gestisce la raccolta dati in massimo quattro categorie dette sub-eventi: affluenze, raggruppamenti, liste, preferenze. Ognuna di queste 4 categorie può riferirsi a entità diverse a seconda dell'elezione, ad esempio nelle elezioni amministrative il sub-evento raggruppamenti è riferito ai candidati sindaco e la categoria liste alle liste comunali, mentre per le politiche i raggruppamenti si riferiscono effettivamente ai raggruppamenti elettorali.

Il riconoscimento OCR nasce dall'esigenza di velocizzare l'inserimento dati all'interno di Eleweb. Attualmente avviene attraverso la copiatura dei fonogrammi (detti comunemente "foglietti") che sono una serie di fogli contenenti i dati ricavati attraverso lo spoglio della sezione elettorale che vengono compilati dai presidenti di seggio.

Per spiegare meglio perché l'OCR è necessario prendiamo l'esempio del **Comune di Milano**. In questo comune ci sono più di 1300 sezioni, se il comune incaricasse una persona per sezione di prendere i dati scritti all'interno del fonogramma e inserirli nella piattaforma dovrebbe assumere circa 1300 persone solo per questo specifico incarico, andando a investire molto denaro per effettuare questa operazione. Quello che il comune allora ha ideato è utilizzare i vigili, ognuno raccoglie fonogrammi da più sezioni e li porta in motocicletta in un centro specifico di raccolta dove vi sono degli incaricati che inseriscono i dati. Il numero di questi incaricati è ovviamente di gran lunga minore a 1300, però devono inserire una grande quantità di informazioni all'interno dell'applicativo. Utilizzare l'OCR potrebbe non solo velocizzare il loro compito ma anche renderlo più efficiente.

Nel proseguo di questo capitolo vedremo quali sono le elezioni configurabili in Eleweb, quali dati vengono raccolti per ogni evento elettorale, dividendoli

nei quattro sub-eventi affluenze, raggruppamenti, liste e preferenze e come sono configurabili.

2.1 Tipologie di elezione

Le tipologie di elezione sono multiple. A livello nazionale troviamo le Europee e le Politiche e a queste si aggiungono i referendum, mentre a livello locale abbiamo Regionali, Provinciali e Comunali.

Le Europee sono ogni 5 anni, sono il più grande evento elettorale transnazionale al mondo e servono per votare i rappresentanti dell'Italia al parlamento europeo. Le Politiche sono anch'esse ogni 5 anni e sono in realtà composte da due eventi elettorali diverse ovvero la votazione per la Camera e per il Senato, essendo eventi diversi, vengono viste in maniera separata all'interno della piattaforma. Le Regionali sono ogni 5 anni e si vota divisi per regioni per eleggere il presidente della giunta e il consiglio regionale. Le Comunali sono anche dette amministrative e si svolgono ogni 5 anni per eleggere il sindaco e la giunta comunale, ovviamente si vota a livello di singoli comuni.

Mentre la comunali e le regionali non coinvolgono l'intera penisola perché non tutti i comuni/regioni votano lo stesso anno, le Europee e le politiche sono a livello nazionale e dunque bisogna considerare uno stress non indifferente per i server di Eleweb, che serve più di 200 comuni, per un totale di oltre 2000 sezioni.

Ora vedremo per ogni tipologia di elezione quali sono i tipi di dati elettorali rilevanti che vengono inseriti nella sezione "raccolta dati" di Eleweb. Verrà fatta un'analisi generale di cosa si vota, ma non si presterà particolare alle soglie di sbarramento, all'età minima e altri particolari, ma sarà uno studio incentrato sul come e sul cosa si vota.

2.1.1 Europee

Le elezioni europee^[3] sono fatte per scegliere i rappresentanti che dovranno fare gli interessi dell'UE. Ogni paese elegge i suoi rappresentanti all'interno del parlamento Europeo. Il numero dei deputati e delle deputate varia da nazione a nazione a seconda della loro popolazione. Come riportato al secondo paragrafo dell'articolo 14 del Trattato sull'Unione europea *"Il Parlamento europeo è composto di rappresentanti dei cittadini dell'Unione. Il loro numero non può essere superiore a settecentocinquanta, più il presidente. La rappresentanza dei cittadini è garantita in modo degressivamente proporzionale, con una soglia minima di sei membri per Stato membro. A nessuno Stato membro sono assegnati più di novantasei seggi"* . Attualmente dei 750 seggi, ne vengono attribuiti 705 e di questi, 76 sono assegnati a deputati e deputate Italiane.

I cittadini di età superiore a 18 anni, ad eccezione di Austria(16) e Grecia(17) votano per partiti politici **nazionali**, ma una volta eletti la maggior parte delle deputate e dei deputati scelgono di aderire a un partito transnazionale. Il sistema di voto non è univoco, ma ogni paese è libero di scegliere il proprio sistema, a patto di rispettare alcune restrizioni[4]:

- Le elezioni si svolgono in un periodo di quattro giorni, da Giovedì a Domenica.
- Ogni cittadino esprime al più una votazione.
- Il numero di deputati eletti da un partito è proporzionale ai voti che riceve.

Concentriamoci ora su come avvengono le elezioni in Italia in modo da poter comprendere quali dati elettorali sono rilevanti e vanno inseriti in Eleweb per la trasmissione al Ministero. La legge numero 18 del 1979 [5] regola come vengono eletti i membri italiani del parlamento europeo, gli elettori possono votare una e una sola tra le liste dei candidati, il voto è valido solo per la lista, non per i candidati che possono essere scelti attraverso l'esercizio del voto di preferenza che può essere usato per votare da uno a 3 candidati, fanno eccezione le liste di minoranze linguistiche per cui si può esprimere la preferenza per solo un candidato. A partire dal 2014 è possibile esprimere il voto di preferenza per più di un candidato a patto che i candidati siano di sesso diverso

Dunque in sintesi possiamo concludere che per quanto riguarda le elezioni europee le informazioni rilevanti al fine della raccolta dati riguardano le liste e le preferenze, inoltre vengono anche segnate le affluenze, vi sono affluenze dette parziali, che vengono inserite nella piattaforma durante i giorni dell'elezione e le affluenze finali, che vengono inserite al termine degli scrutini.

2.1.2 Politiche

Le elezioni politiche in Italia permettono ai cittadini di eleggere coloro che eserciteranno il potere legislativo, ovvero i membri del Parlamento. Il parlamento è costituito da due differenti camere, il Senato della Repubblica e la Camera dei Deputati. L'elezione è per suffragio universale e avviene in contemporanea ma avviene in modalità distinte, per la Camera votano i cittadini maggiorenni raggruppati in circoscrizione, mentre la divisione per il Senato è su base regionale. Dunque nonostante il periodo elettorale sia lo stesso le votazioni sono due e vengono viste su Eleweb come due eventi elettorali diversi, con un inserimento dati distinto. La legge elettorale che regola la votazione del Parlamento è il così detto *Rosatellum* più formalmente "Legge Rosato" che prevede un sistema misto proporzionale(prevalentemente) e maggioritario. L'Elettore può votare in modi distinti:

- votando la lista, il cittadino dà il suo voto a essa e al candidato a essa collegato per il collegio uninominale
- votando il candidato uninominale senza esprimere il voto per la lista. Nel caso in cui il candidato sia relativo a più liste il voto viene ripartito tra esse, se invece il candidato è relativo a un'unica lista il voto viene attribuito a essa.

Le politiche non prevedono che il cittadino eserciti un voto di preferenza perché i candidati sono già indicati dai partiti in un ordine da loro stabilito. Non è permesso il voto disgiunto (tra lista e candidato uninominale).

In conclusione le informazioni rilevanti al fine della raccolta dati riguardano affluenze parziali e finali, raggruppamenti e liste. Non sono previste preferenze.

2.1.3 Regionali

Le elezioni regionali, non riguardano tutto il paese ma le singole regioni, in Italia dunque ci sono 20 elezioni regionali. Il sistema di votazione è simile ma non uguale per tutte le regioni. Vi sono delle regioni a statuto ordinario e delle regioni a statuto speciale che hanno un sistema di elezione diverso, inoltre anche all'interno delle regioni ordinarie vi sono delle differenze. Quello che accomuna tutte le regioni è che la votazione viene fatta per eleggere il presidente della giunta regionale e in tutte le regioni a statuto ordinario la votazione è a turno unico (ovvero non prevede un ballottaggio) eccezion fatta per la Toscana. In genere all'interno della votazione a prescindere dalla regione è possibile esprimere sia il voto per le liste provinciali che per il presidente di giunta. Se si vota solo per la lista provinciale in automatico il voto viene assegnato anche al presidente di giunta corrispondente. In alternativa si può votare solamente per il presidente di giunta, nel caso il voto viene ripartito anche sulla coalizione che lo sostiene. Si può inoltre esprimere il voto di preferenza che in alcune regioni è singolo come nel caso del Piemonte o multiplo come nel caso di Lombardia e Marche che prevedono fino a due voti di preferenza a patto che riguardino candidati di sesso differente. Dunque mentre le politiche e le europee prevedono un'unica metodologia di votazione per tutti i comuni e le regioni d'Italia, la metodologia di votazione in Italia per le regionali varia di regione in regione e per questo è ancora più importante la corretta configurazione dell'evento elettorale all'interno dell'applicativo.

Possiamo concludere che i dati che vengono raccolti in questa tipologia di elezione sono affluenze parziali e totali, le liste, i raggruppamenti (i candidati presidente) e le preferenze.

2.1.4 Comunali

Le elezioni comunali vengono dette anche amministrative e riguardano l'elezione del sindaco e del consiglio comunale. Vi è un evento elettorale per comune, dunque all'interno di Eleweb devono essere gestite più di 200 elezioni. Le elezioni comunali nei comuni di più di 15.000 abitanti prevedono che ogni elettore abbia diritto a:

- Un voto per il sindaco
- Un voto per una lista tra quelle candidate al consiglio comunale
- Al più due voti di preferenza

Vi è la possibilità di esprimere un voto solo per una lista e non per il candidato sindaco, in questo caso il voto viene automaticamente esteso al candidato sindaco collegato alla lista. Se invece l'elettore esprime solo il voto per il candidato sindaco non viene conteggiato il voto per la lista. Vi è la possibilità di esprimere un voto disgiunto. Nel caso in cui al primo turno nessun candidato abbia raggiunto la soglia del 50% si ricorre al ballottaggio, in cui concorrono i due candidati che hanno ottenuto più voti al primo turno. I comuni con popolazione minore di 15.000 abitanti hanno un sistema elettorale che non prevede il ballottaggio ma a turno unico.

Possiamo concludere che i dati elettorali che vengono raccolti per le amministrative sono affluenze parziali e totali, liste, raggruppamenti e preferenze. Per quanto riguarda invece il possibile ballottaggio abbiamo sempre affluenze parziali e finali e le liste (in questo caso chiamate candidati).

2.1.5 Referendum

Il referendum è una votazione attraverso la quale viene chiesto all'elettore di esprimersi su una singola domanda. Spesso quando vi è un referendum in realtà i cittadini devono esprimersi su più domande, in Eleweb ogni domanda e dunque ogni referendum è un evento elettorale diverso, dunque con una raccolta dati separata dagli altri (è però possibile ribaltare alcuni dati come le affluenze su più eventi elettorali). Esistono 4 tipi di referendum:

- Consultivo
- Confermativo
- Abrogativo
- Propositivo

Il **consultivo** è il meno vincolante, il cittadino esprime la propria opinione sulla domanda, lo stato non è tenuto a tener conto dell'esito del referendum. Il **confermativo** è previsto dopo l'approvazione di una legge da parte del Parlamento. Al cittadino è richiesto di confermare l'entrata in vigore della suddetta, se il referendum ha esito positivo la legge entra in vigore. L'**abrogativo** chiede al cittadino di esprimere la sua volontà sull'annullamento parziale o totale di una legge, in caso di esito positivo la norma sarà rimossa dall'ordinamento. Infine vi è il **propositivo** che parte su proposta dei cittadini che possono raggiungere un determinato numero di firme e presentare una proposta di legge. La proposta può essere approvata in diversi modi: il testo di legge viene proposto agli elettori che possono approvarlo o meno, in alternativa può essere proposto al Parlamento. Se accetta la proposta non vi è bisogno di un referendum altrimenti i cittadini vengono chiamati a votare, se invece il parlamento propone un testo alternativo può essere indetto un referendum.

A prescindere dal tipo di referendum i dati importanti per l'elezione sono sempre gli stessi, a partire dalle affluenze che sono ancora più importanti che per gli altri eventi elettorali in quanto in Italia un referendum per essere valido deve aver raggiunto il quorum del 50%. Oltre alle affluenze vengono conteggiati i dati sulle liste, che per i referendum sono solo 2: sì o no.

ELEZIONE	AFFLUENZE	RAGGRUPP.	LISTE	PREFERENZE
COMUNALI	X	X	X	X
REGIONALI	X	X	X	X
SENATO	X	X	X	
CAMERA	X	X	X	
EUROPEE	X		X	X
BALLOTTAGGIO	X		X	
REFERENDUM	X		X	

Tabella 2.1: elenco sub-eventi per tipologia di elezione

2.2 Raccolta dati in Eleweb

In Eleweb la raccolta dati avviene in uno specifico modulo denominato appunto "Raccolta Dati". L'utente attraverso le funzioni del modulo può caricare dati provenienti dalle sezioni. Le funzioni per la raccolta sono impostate rispettando il più possibile le consuetudini degli utenti che caricano i dati contenuti nei fonogrammi. A cominciare dai dati di affluenza, per ogni fascia oraria, l'utente carica i dati per ogni sezione elettorale. Per ogni tipologia di evento elettorale l'utente carica i dati relativi ai voti dei raggruppamenti, i voti di lista e i voti di preferenza a seconda della tipologia dell'evento.

Andiamo a vedere ora per ogni tipologia di dato elettorale quali sono le possibili opzioni di configurazione dei sub-eventi per adattarlo all'evento elettorale impostato.

2.2.1 Affluenze

L'utente ha la possibilità di inserire i dati di affluenza man mano che arrivano dal seggio, in quanto le affluenze devono essere comunicate entro 15 minuti dall'orario di raccolta. Si possono avere differenti fasce orarie di raccolta all'interno dell'evento elettorale che devono essere configurate nell'applicativo in modo da poter essere visualizzati sulla videata di raccolta. La tipologia che caratterizza la chiusura definitiva dei seggi (detta affluenza finale) raccoglie i dati definitivi per l'evento elettorale per ogni sezione, questi saranno i dati proposti automaticamente per il caricamento degli scrutini.

Abbiamo dunque capito che vi sono affluenze parziali e affluenza finale, mentre per la seconda viene sempre divisa l'affluenza in maschi femmine e totali, a seconda del comune e della tipologia di elezione l'affluenza parziale potrebbe essere divisa oppure essere indicata solo la totale. Questa particolarità è configurabile in Eleweb e dobbiamo tenerne conto, in quando a seconda di questa scelta dovremo andare a caricare o meno i dati di affluenza maschi e femmine con l'OCR.

Come abbiamo visto per referendum e politiche spesso abbiamo eventi elettorali diversi che hanno i dati di affluenza in comune, Eleweb permette di configurare l'affluenza come ribaltabile su più elezioni, questa informazione però non sarà molto rilevante per il modulo OCR in quanto questa logica è già configurata in fase di raccolta dati.

2.2.2 Raggruppamenti

Questo sub-evento è presente solo nei tipi di elezione per cui è previsto. Per questo sub-evento i voti al candidato (o raggruppamento) sono sempre previsti, poi il sistema può essere impostato per raccogliere altri tipi di dato, come i voti non assegnati al candidato e i voti espressi al solo candidato. Spesso nei fonogrammi alla fine delle "colonne" con i voti validi, i voti contestati e i voti solo candidato, ci sono dei riepiloghi con i totali, questi non vanno presi tramite OCR perché vengono calcolati automaticamente dall'applicativo sulla base dei dati inseriti. Una volta impostati questi valori, possono essere previsti uno o più dei seguenti campi di riepilogo:

- Schede contestate(A)
- Schede bianche(B)
- Schede nulle(C)

- Voti nulli(d)

I voti solo al candidato possono essere riferiti al singolo raggruppamento oppure trovarsi nei campi di riassuntivo in fondo al fonogramma, a seconda dell'impostazione il dato si troverà in posizione diversa e sarà un dato multiplo (uno per candidato) o singolo (riassuntivo) All'interno dell'applicativo la presenza o meno dei questi campi "opzionali" può essere gestita tramite la configurazione elezione, è importante capire a cosa si riferiscono i campi e quando sono presenti all'interno del fonogramma per riuscire a fare un corretto training e rilevamento tramite OCR.

2.2.3 Liste

La configurazione delle liste è molto simile a quella per i raggruppamenti, in quanto anche sui fonogrammi non si differenziano molto i dati per i due sub-eventi. Il campo voti (voti alla lista) non può essere rimosso. Sono presenti in maniera opzionale i voti non assegnati perché in alcuni fonogrammi è presente. Anche in questo caso, così come per i raggruppamenti è possibile selezionare quali e quanti dei seguenti campi di riepilogo sono presenti nel fonogramma:

- Schede contestate(A)
- Schede bianche(B)
- Schede nulle(C)
- Voti nulli(d)

Inoltre i totali vengono calcolati in automatico dall'applicativo come avveniva nel sub-evento precedente. Una cosa che differisce tra i due sub-eventi è che i voti solo presidente non assegnati sono editabili solo come totale, non per singola lista.

Anche qui bisogna prestare molta attenzione a come sono settati i flag relativi a questi dati, in quanto per capire quali dati prendere dal fonogramma e la loro posizione sul foglio bisogna essere certi di avere impostato bene il sistema in fase di configurazione elezione e aver verificato tutte le possibile casistiche in fase di training

2.2.4 Preferenze

Le preferenze sono il sub-evento un po' più complicato da gestire in quanto prevedono un grande numero di dati perché vi sono dei candidati per ogni lista e dunque va fatto teoricamente un training per lista. Dal punto di vista della configurazione elezione però è molto semplice da gestire, è possibile impostare solo il numero di preferenze esprimibili (dato non rilevante per l'OCR) e la presenza opzionale della colonna voti non assegnati.

EVENTI ELETTORALI	Affluenza	Raggruppamenti
COMUNALI	Affluenze	Candidati SINDACO
REGIONALI	Affluenze	Candidati PRESIDENTE
BALLOTAGGI	Affluenze	
REFERENDUM	Affluenze	
CAMERA	Affluenze	COALIZIONI (solo lettura)
SENATO	Affluenze	COALIZIONI (solo lettura)
EUROPEE	Affluenze	

Tabella 2.2: Tabella relazioni tra eventi elettorali affluenza e raggruppamenti

EVENTI ELETTORALI	Liste	Candidati
COMUNALI	Liste	Voti di Preferenza
REGIONALI	Liste	Voti di Preferenza
BALLOTAGGI	Candidati	
REFERENDUM	Risposte: SI-NO	
CAMERA	Liste-Candidati	
SENATO	Liste-Candidati	
EUROPEE	Liste	Voti di Preferenza

Tabella 2.3: Tabella relazioni tra eventi elettorali e liste e candidati

Capitolo 3

Progettazione del sistema

Alla luce dello studio del dominio abbiamo compreso che il sistema Eleweb è complesso perché gestisce diverse tipologie di elezione. Ogni tipologia è presente in tutta Italia, dunque ogni comune avrà dei fonogrammi(o foglietti). Per progettare il sistema si è partiti dallo studio dei foglietti per capire che approccio utilizzare per l'OCR. Una volta compreso l'approccio si è passato al suo studio e alla sua implementazione e infine alla scelta del miglior OCR sul mercato in relazione ai fattori di costo, velocità e affidabilità della lettura.

3.1 Studio fonogrammi

Le interviste per analizzare i requisiti del sistema sono state realizzate dall'azienda precedentemente all'inizio di questa tesi, dunque le informazioni su quali fossero i bisogni degli utenti ci sono stati forniti direttamente da Pro Logic Informatica. Gli utenti, ovvero i presidenti di seggio, i messi comunali e gli scrutinatori hanno bisogno di un sistema veloce, che semplifichi l'inserimento dati e che non aggiunga ulteriori operazioni da svolgere durante l'evento elettorale. L'idea iniziale era quella di trovare dei pattern comuni nei vari foglietti per riuscire a fare un'analisi statistica di dove sono posizionati i dati, ad esempio da quale parola è preceduto un dato o il posizionamento relativo dei dati tra loro. Prendendo a campione le affluenze è importante capire se le informazioni su votanti maschi femmine e totali sono messe una sotto l'altra o una di fianco all'altra, per i raggruppamenti se vi è un elenco verticale o orizzontale di nomi, ecc.

Partiamo con l'analisi attraverso alcuni esempi di foglietti:

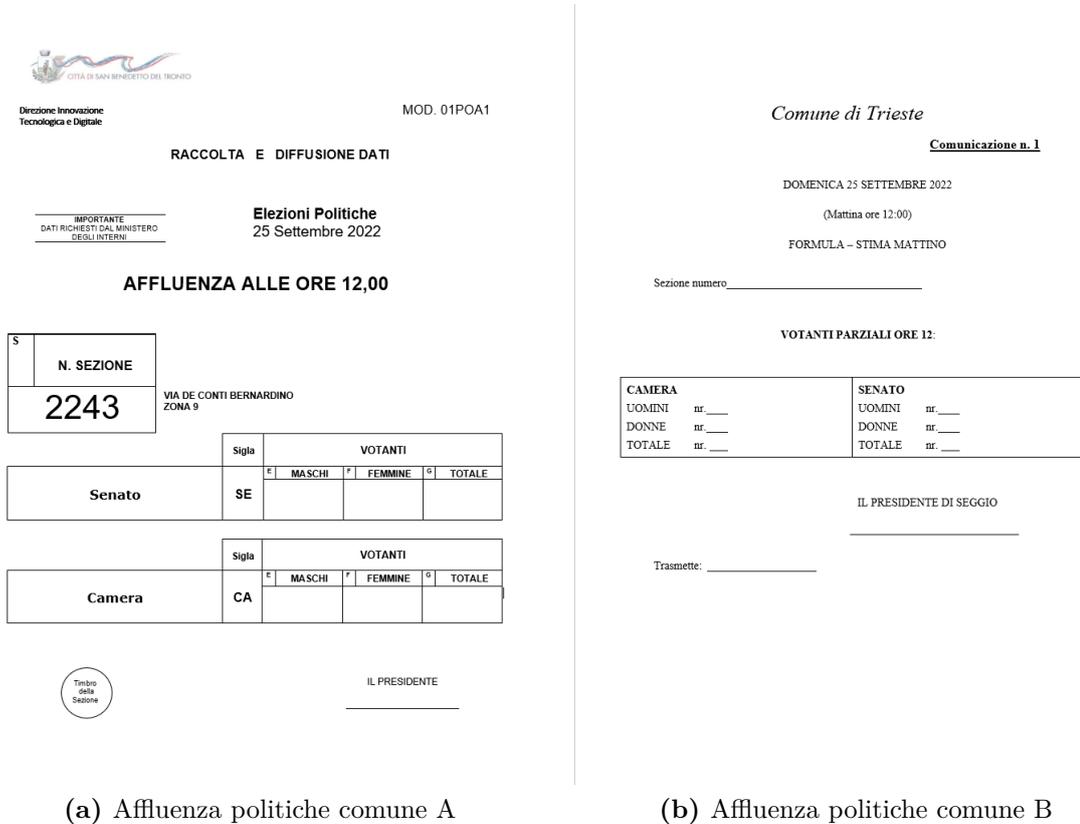


Figura 3.1: Confronto affluenza politiche tra due comuni

Mettiamo a confronto queste due immagini che si riferiscono allo stesso sub-evento, ovvero affluenze alle politiche di due differenti comuni. Notiamo che non vi sono molti aspetti in comune, per il comune A le affluenze al Senato sono posizionate sopra quelle per la Camera, mentre per il comune B sono una di fianco all'altra. Inoltre cambia anche come vengono indicati i votanti maschi e femmine, in quanto nel foglio sulla destra sono chiamati uomini e donne, in quello di sinistra maschi e femmine. Maschi femmine e totale sono posizionati in orizzontale per il comune A e in verticale per il comune B. Non vi è dunque almeno per questo caso un pattern comune, analizzando altri foglietti ci si è resi conto che i pattern sono 4/5 e variano da comune a comune ma sono troppi per poter fare un'analisi statistica sui dati riportati dall'OCR per capire dove sono posizionati i dati in automatico.

Ora focalizziamoci su questo secondo esempio: La prima cosa che notiamo in figura 3.2 è che nel foglio A vi sono sia le liste che i raggruppamenti, mentre nel foglio B solo le liste, mentre un altro foglietto è usato per i raggruppamenti, notiamo

RACCOLTA E DIFFUSIONE DATI 25 SETTEMBRE 2022 Mod.08SUSP-03

IMPORTANTE
DATI RICHIESTI DAL MINISTERO DEGLI INTERNI

CITTA' DI SAN BENEDETTO DEL TRONTO
Direzione Innovazione
Tecnologica e Digitale

RISULTATI
Senato della Repubblica
Collegio n° 3

S I SEZIONE
2243

F SCHEDE CONTESTATE D SCHEDE BIANCHE E SCHEDE NULLE B SCHEDE VALIDE CON VOTO AL SOLO CANDIDATO (B) G SCHEDE SCRUTINATE (F+D+E+B+C)

CANDIDATO UNINOMINALE	1 VOTI ATTRIBUITI (CANDIDATO+LISTE COLLEGATE)	2 VOTI ATTRIBUITI (AL SOLO CANDIDATO)	LISTE COLLEGATE	3 VOTI ATTRIBUITI (ALLA SOLA LISTA)
PANNELLI FILIPPO			ITALIA SOVRANA E POPOLARE	
			NOI MODERATI/LUPI - TOTI - BRUGNARO - UDC	
LEONARDI ELENA			FORZA ITALIA	
			FRATELLI D'ITALIA CON GIORGIA MELONI	
			LEGA PER SALVINI PREMIER	
SPALLETTI ANDREA			ITALEXIT PER L'ITALIA	
CATALEDI ROBERTO			MOVIMENTO 5 STELLE	
COMPAGNUCCI ELENA			UNIONE POPOLARE CON DE MAGISTRIS	
BERARDI DANILLO			PARTITO COMUNISTA ITALIANO	
VENTURINI GIULIANA			VITA	
GATTARI MIRELLA			+EUROPA	
			ALLEANZA VERDI E SINISTRA	
			IMPEGNO CIVICO LUIGI DI MAIO - CENTRO DEMOCRATICO	
			PARTITO DEMOCRATICO - ITALIA DEMOCRATICA E PROGRESSISTA	
URBINATI FABIO			AZIONE - ITALIA VIVA - CALENDIA	
DI STEFANO CRISTIANA			ALTERNATIVA PER L'ITALIA - NO GREEN PASS	
	A TOTALE (B+C)	B TOTALE		C TOTALE

IL PRESIDENTE

Timbro della Sezione

(a) Liste politiche comune A

(b) Liste politiche comune B

Figura 3.2: Confronto foglietto liste tra due comuni

Comune di Trieste
Comunicazione n. 7
FORMULA SCRUTINIO - LISTE
SENATO

Sezione numero _____

LISTE	VOTI VALIDI
1 IL POPOLO DELLA FAMIGLIA	
2 LEGA	
3 FORZA ITALIA	
4 FRATELLI D'ITALIA	
5 PARTITO DEMOCRATICO	
6 EUROPA VERDE RINASCI SAN BENEDETTO	
7 IL POPOLO DELLA FAMIGLIA	
8 PARTITO DEMOCRATICO	
9 EUROPA VERDE RINASCI SAN BENEDETTO	
10 MOVIMENTO CINQUE STELLE 2020	
TOTALE VOTI DI LISTA	

(H)

IL PRESIDENTE DI SEGGIO

Trasmette: _____

Il totale (B) deve corrispondere al totale (C) + (H)

che hanno in comune il fatto di avere i dati in verticale. Nonostante ciò la struttura dei due fogli è troppo diversa per poter pensare di trovare uno schema comune a entrambi.

Bastano semplicemente questi due esempi per capire che i fonogrammi non hanno pattern comuni e non vi è uno standard a livello nazionale. Andare a cercare dei pattern "simili" e non uguali sarebbe stata una cosa possibile ma volevamo un sistema utilizzabile da ogni comune, che durasse negli anni anche a fronte di cambiamenti nella struttura di foglietti. La seconda soluzione che si è pensata è stata quella di ideare dei foglietti, creati dall'azienda da distribuire ai comuni per le elezioni, il problema che si è presentato è che alcuni comuni potrebbero non gradire cambiare fonogrammi che usano da anni, inoltre ogni tipologia di elezione dovrebbe avere un modello e alcuni eventi elettorali, come ad esempio le Regionali e le Comunali non sono uguali in tutta Italia dunque andrebbero creati dei foglietti supplementari che variavano da comune a comune.

Per fornire massima elasticità e accuratezza nel riconoscimento della posizione

dei dati sul foglio la soluzione che si è pensata è dunque quella del **training**.

3.2 Training

Il training è la parte principale del modulo OCR, senza il quale non sarebbe possibile riconoscere quali tra i dati sul foglietto devono essere inseriti, e dove, nelle maschere di inserimento nella sezione raccolta dati. L'idea è nata perché le altre due strade che sicuramente permettevano di non fare operazioni supplementari prima delle elezioni, non erano percorribili e dunque si è scelto di optare per il training a patto di renderlo il più semplice e veloce possibile. Il training si può riassumere brevemente definendolo come il metodo utilizzato per dire all'OCR quali tra i box che trova, sono quelli relativi ai dati che andranno inseriti nella web application di Eleweb.

L'OCR scansiona il foglio e restituisce un json [6] che contiene dei bounding box, uno per parola. Un box non è altro che il più piccolo contenitore rettangolare che contiene una parola. All'interno del json ci sono le coordinate di tutti i box. La teoria è che disegnando dei rettangoli sulla rappresentazione digitale del foglio in corrispondenza dei dati che andranno inseriti nelle maschere e salvando le coordinate sul database, sarà possibile confrontare quali box matchano i rettangoli disegnati in fase di training, così da prendere i dati necessari per riempire i campi di inserimento dati all'interno della web app.

Il training deve essere ovviamente fatto dopo che i foglietti per le elezioni sono stati ideati e stampati in versione definitiva. Può essere fatto o direttamente dai comuni o dal team di Eleweb, a seconda degli accordi presi con le singole amministrazioni. A prescindere da chi farà il training ciò che è necessario è che sia semplice, intuitivo e veloce.

Eleweb è un applicazione già esistente dunque andremo a focalizzarci maggiormente sull'UX più che sull'UI perché l'interfaccia deve avere delle similarità con le altre schermate all'interno del sistema, per non sembrare completamente slegata dal resto dell'applicazione. Per quanto riguarda l'UX l'utente deve sempre essere in grado di sapere per quale sub-evento sta facendo il training e di capire quale rettangolo si riferisce a quale dato inoltre bisogna cercare di prevenire che faccia qualche errore guidandolo durante questa fase. In caso di errore l'utente deve poter ridimensionare spostare e cancellare il rettangolo e in caso di errori multipli annullare il training.

3.3 OCR

Come già riportato nell'introduzione di questa tesi è importante che l'OCR sia veloce e che abbia un accuracy elevata. Per accuracy intendiamo la probabilità che la stringa che l'OCR ci restituisce sia uguale a ciò che è scritto sul foglio. Preferibilmente dovrebbe essere lato client, questo per non andare a sovraccaricare i server aziendali durante l'evento elettorale. Il problema riscontrato già nella prima fase di ricerca è che il tool più affidabile che si è trovato lato client è Tesseract.js che come riportato nel sito ufficiale [7] è molto affidabile per riconoscere testo stampato mentre per il testo scritto a mano la quantità di caratteri riconosciuti correttamente diminuisce notevolmente. Si è dunque optato per una soluzione lato server di cui parleremo nel capitolo dedicato.

Dividendo il training dalla scansione vera e propria i presidenti di seggio o i messi comunali si occuperanno solo di scattare la foto e attendere il risultato che era quello richiesto dai comuni. In questa fase è molto importante che vengano matchati i box che effettivamente corrispondono ai rettangoli disegnati nella fase di training, è questo il vero punto critico del modulo, che analizzeremo a breve.

3.4 Backend

Il back-end è fatto tramite PostgreSQL detto comunemente Postgres che è un database relazionale open source che utilizza il modello ACID [8] ovvero atomicità, consistenza, isolamento, atomicità.

La scelta dell'uso di Postgres è stata obbligatoria, considerando che il database di Eleweb è già esistente, sono state inserite due tabelle, relazionate tra di loro che permettono di salvare le coordinate dei rettangoli del training una volta premuto il pulsante di salvataggio.

La prima tabella si chiama `ewocr_campi` e contiene solamente due colonne, una chiamata `id` e una nome del campo. E' utilizzata per essere referenziata dalla seconda tabella in modo da comprendere per ogni record a che campo si riferiscono le coordinate salvate. La seconda tabella, `ewocr_ele` è invece la tabella principale dell'applicativo che contiene più colonne, alcune per identificare l'elezione, poi vi è l'`idcampo` della precedente tabella e `x,y,w,h` riferite a `x` iniziale, `y` iniziale larghezza e altezza. Infine vi è una colonna valore utilizzata solo per alcuni tipi di campi.

La figura 3.3 mostra la relazione tra le due tabelle, ovviamente il database di eleweb è molto più grande e contiene alcune tabelle che contengono come chiave primaria `id_ente` e `cd_ele` che vengono referenziati come chiave esterna nella tabella `ewocr_ele`

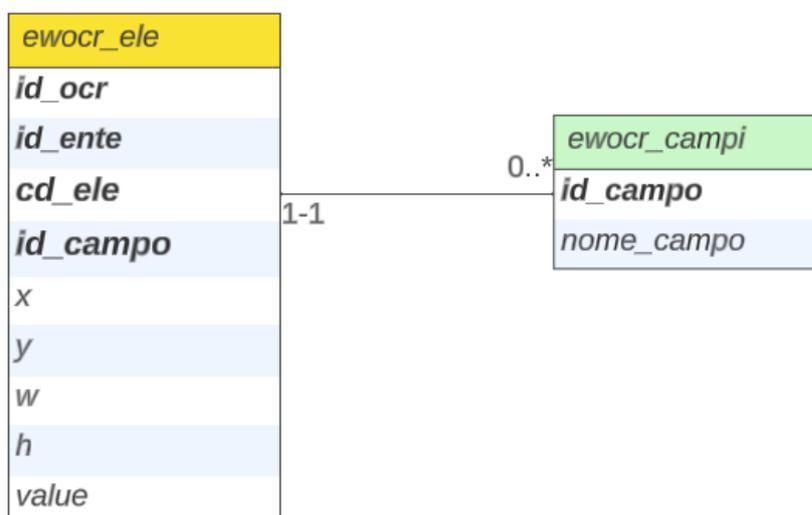


Figura 3.3: Diagramma e-r backend

Capitolo 4

Progettazione Training

In questo capitolo analizzeremo la tecnologia utilizzata nel training, la gestione del backend, le scelte legate all'UX/UI e soprattutto la logica con cui è stato ideato e realizzato.

4.1 Tecnologie utilizzate

Per realizzare l'interfaccia utente e la pagina web in generale sono stati utilizzati HTML e CSS. Per lo sviluppo del codice lato client è stato utilizzato javascript mentre lato server PHP. La scelta di utilizzare queste tecnologie è stata obbligatoria, dato che l'applicativo Eleweb è scritto utilizzando questi linguaggi e si è deciso di utilizzarli anche per questo modulo OCR.

4.1.1 PHP

PHP è un linguaggio di scripting, dunque lato server, che permette di creare pagine web dinamiche interpretando le istruzioni del client e restituendogli ciò che è stato elaborato. E' molto efficace per l'integrazione con i database e con l'HTML dunque è facile inserire codice di questo tipo all'interno di una pagina PHP ed è open source e multiplatforma.

4.1.2 Canvas

Il canvas non è una tecnologia, ma è un elemento di HTML che permette di avere una superficie disegnabile con javascript. E' possibile ottenere il contesto del canvas che rappresenta l'area dell'elemento e fornisce i metodi con cui disegnare o scrivere su di esso.

Attraverso l'utilizzo del metodo `strokeRect(x,y,height,width)`[9] è possibile disegnare un rettangolo ed è questo che verrà utilizzato all'interno dell'applicazione per creare i rettangoli da confrontare con i bounding box.

4.2 Logica del training

Il training va fatto prima dell'elezione una volta che i foglietti sono definitivi. L'idea è che va fatto un training per ogni tipologia di sub-evento, essendo che i fonogrammi cambiano ovviamente a seconda del sub-evento ma sono uguali per tutte le sezioni. La funzione di training è diversa a seconda del sub-evento elettorale, ma la logica di base è sempre la stessa. L'utente seleziona il sub-evento attraverso dei bottoni, una volta selezionato verrà visualizzato un file picker per scegliere il foglietto da visualizzare in un canvas, una volta che il file è selezionato si può scegliere di iniziare il training.

4.2.1 Funzione di training

Il training è una funzione, che cambia a seconda del sub-evento, che richiama a sua volta una funzione al suo interno che si auto-invoca ogni 50 millisecondi. All'utente devono essere date delle istruzioni. Ogni istruzione è relativa a un rettangolo che l'utente deve disegnare per mappare le coordinate del foglio in cui è presente il dato che l'OCR deve leggere e inserire all'interno di una maschera.

Ogni sub-evento può avere diverse tipologie di dati, a seconda dell'elezione. In Eleweb le informazioni riguardanti i dati presenti nel sub-evento sono salvate in alcuni flag nel database. I flag sono dei bit che sono messi a 1 o a 0 a seconda della presenza o meno del dato elettorale. Vengono verificati tutti i flag relativi all'evento per capire quali istruzioni dare all'utente, ogni volta che viene trovato un flag che indica che il dato è presente all'interno del sub-evento viene aggiunta un'istruzione al vettore chiamato **istruzioni**.

Oltre al vettore di istruzioni viene anche caricata una struttura ausiliaria chiamata **aus** che serve per memorizzare i rettangoli disegnati durante il training e poi viene riportata, una volta terminata la funzione, nella struttura **Trainingdata** che contiene tutti i dati relativi ai training effettuati in modo da poterli inviare al server una volta che si preme sul pulsante salva training.

Una volta caricato il vettore istruzioni viene chiamata la funzione che effettivamente controlla il training. Questa funzione richiama se stessa ogni 50 millisecondi e termina se avviene uno di questi tre eventi:

- si completa il training
- si clicca sul pulsante annulla training

- si clicca sul menù laterale di Eleweb cambiando scheda

Prima di cominciare viene caricata una variabile chiamata **prevEmptyField** che indica quale era il campo all'interno dell'oggetto `aus` che era vuoto prima che iniziasse l'attuale iterazione. A ogni iterazione si verifica se la lunghezza attuale del vettore **rect** che è un vettore che contiene le coordinate dei rettangoli disegnati e viene azzerato ogni volta che inizia il training. Se la lunghezza del vettore è minore del numero di rettangoli che ci si aspetta abbia alla fine del training allora la funzione continuerà a richiamare se stessa. Se si verifica che il vettore `rect` ha cambiato dimensione rispetto all'iterazione precedente allora ci sono due possibilità:

- E' stato aggiunto un rettangolo
- E' stato eliminato un rettangolo

Per fare un esempio, le affluenze finali hanno sempre 3 dati: maschi, femmine e totale, dunque la dimensione finale di `rect` dovrà essere 3, se è inferiore a 3 allora la funzione continua a iterare. Se ho riquadrato maschi e femmine, allora il contatore sarà a due. Se alla futura iterazione la dimensione di `rect` sarà uno, allora un rettangolo è stato eliminato.

In caso di inserimento, si va a inserire in `aus[prevEmptyField]` le coordinate del rettangolo disegnato. In caso di cancellazione invece si va a iterare nei rettangoli contenuti nel vettore `aus` e si vede quale di quelli non è presente all'interno di `rect` e si va a mettere a null quell'oggetto nel vettore `aus`.

In caso di inserimento o eliminazione la cosa comune è che deve cambiare l'istruzione fornita all'utente, per capire quale istruzione mostrare, si scorre il vettore `aus` e si vede qual'è il primo elemento a null, si controlla e nel caso il nome dell'elemento sia diverso da `PrevEmptyField`, si aggiorna il valore della variabile con il nuovo campo vuoto e si richiede l'istruzione nel vettore `istruzioni` che ha la stessa posizione del campo trovato vuoto in `aus`. Questo fa capire come la costruzione dei due vettori non è casuale ma deve essere fatta una mappatura ordinata 1:1 tra l'istruzione e il campo che conterrà il rettangolo.

In caso di cancellazione, usando questo stesso ragionamento si torna a chiedere l'istruzione per creare il rettangolo cancellato, una volta che questo è stato fatto si riprende dall'ultima istruzione prima della cancellazione perché sarà il primo campo vuoto che verrà dunque messo in `PrevEmptyField`.

Esempio: devo trainare le affluenze maschi femmine totale, ho disegnato i rettangoli di maschi e femmine dunque mi viene chiesto di disegnare il rettangolo del totale. Se cancello i maschi l'istruzione diventa di disegnare i maschi, una volta fatto torna a richiedermi il totale.

Una volta terminato il training, i valori dei rettangoli nel vettore `aus` vengono riportati in una struttura di raccolta dati chiamata `trainingData` che contiene tutti

i dati relativi ai training effettuati in modo da poterli inviare al server una volta che si preme sul pulsante salva training.

Questa struttura è un vettore di oggetti, ogni oggetto mappa un sub-evento, abbiamo dunque affluenze parziali, finali, raggruppamenti, liste, preferenze. Ogni sub-evento contiene tutti i possibili campi che andranno mappati con le coordinate dei rettangoli disegnati. Ad ogni rettangolo contenuto nel vettore `aus` è assegnato un valore che indica a che dato il rettangolo si riferisce e verrà mostrato in un tooltip presente quando si andrà con il mouse sopra il rettangolo.

Se vogliamo accedere alle coordinate dei votanti totali delle affluenze parziali accederemo in questa maniera:

`trainingData.affluParziali.totali`

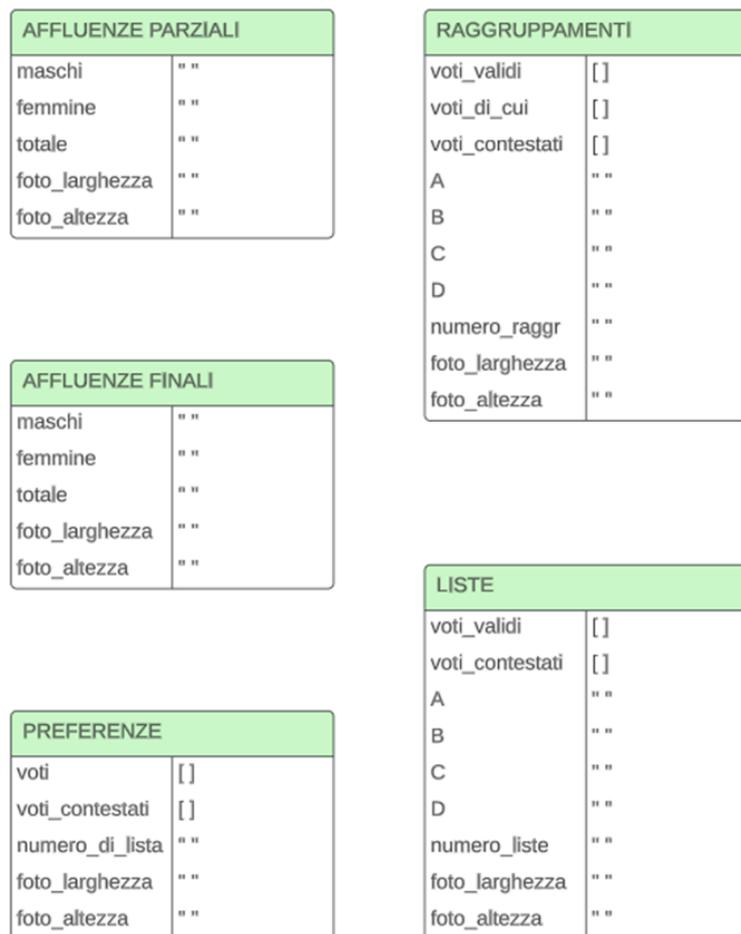


Figura 4.1: Strutture all'interno di `trainingData` con inizializzazione dei campi

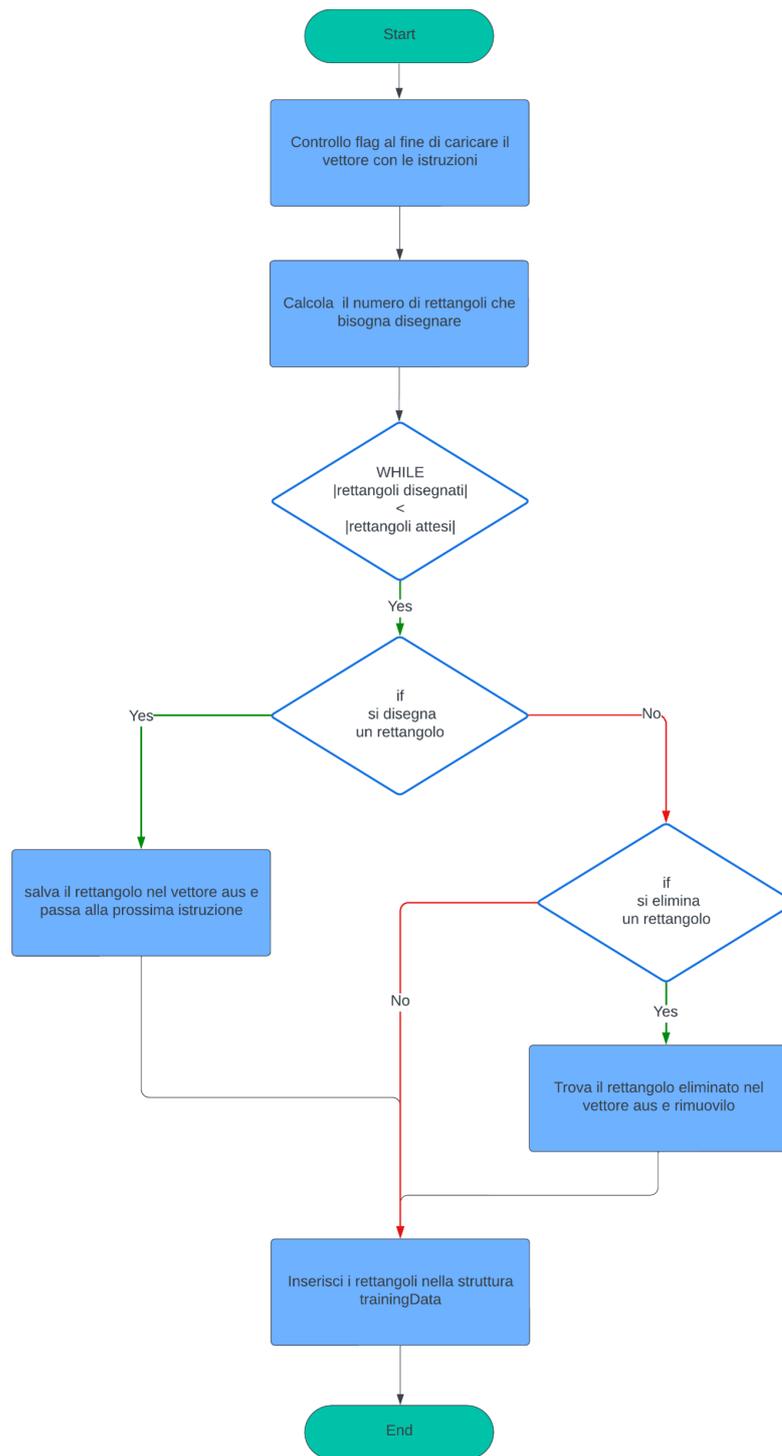


Figura 4.2: Flow chart funzione training

4.3 La costruzione dei rettangoli

Si è già spiegato che i rettangoli sono fondamentali perché l'OCR restituisce dei bounding box rettangolari che rappresentano la posizione delle parole nello spazio del foglio. A seconda della tipologia di OCR utilizzato ci sono vari modi di fornire le coordinate dei box:

- X,Y dell'angolo del rettangolo in alto a sinistra e in basso a destra
- X,Y dell'angolo del rettangolo in alto a sinistra, larghezza e altezza

Questi valori possono essere espressi in base alla dimensione reale dell'immagine e partire quindi da (0,0) nell'angolo in alto a sinistra e (larghezza immagine,altezza immagine) nel punto in basso a destra, tenendo dunque conto che l'asse delle ascisse cresce da sinistra a destra e quello delle ordinate dall'alto al basso.



Figura 4.3: Coordinate non normalizzate

Altri OCR invece rappresentano i box attraverso delle coordinate che vanno da (0,0) a (1,1) mappando gli estremi del foglio in quei punti per avere così delle coordinate normalizzate.

L'OCR scelto, che verrà analizzato nel capitolo 5 fornisce le coordinate in questo secondo modo, dunque ogni volta che viene tracciato un rettangolo sul canvas, avrà delle coordinate espresse in base alla dimensione del foglio, queste devono essere mappate da globali a normalizzate tramite un'operazione detta di normalizzazione, ovvero semplicemente ogni punto viene mappato dividendo la sua coordinata per la larghezza e l'altezza del foglio. Si analizzano x,larghezza,y e altezza del rettangolo

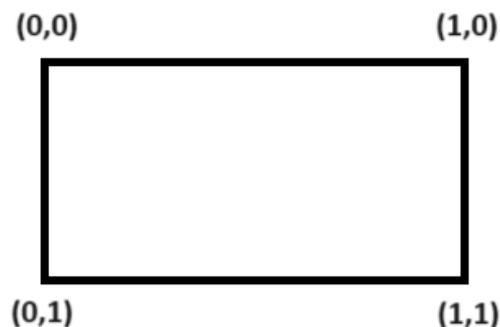


Figura 4.4: Coordinate normalizzate

tracciato e si dividono, le prime due per la larghezza del canvas, le ultime due per la sua altezza, in questa maniera tutte le grandezze saranno mappate tra 0 e 1. Questo ha influito sulla scelta dell'OCR in quando queste coordinate permettono di essere indipendenti dall'effettiva risoluzione e dimensione dell'immagine e poter lavorare in maniera globale su ogni fonogramma usato.

Durante la funzione di training, è possibile eseguire alcune operazioni come la cancellazione, la modifica della dimensione (resizing) e lo spostamento dei rettangoli.

4.3.1 Le funzioni mousemove, mouseup e mousedown

Le funzioni mouseup, mousedown, e mousemove sono eventi del mouse in JavaScript che vengono utilizzati per intercettare l'interazione dell'utente con gli elementi di una pagina web. Ognuno di questi eventi è relazionata a un'azione specifica che l'utente esegue tramite il mouse. MouseDown viene scatenato quando l'utente fa clic sul tasto sinistro del mouse e lo tiene premuto, viene spesso utilizzato per catturare l'inizio di un trascinamento. MouseUp viene scatenato quando si rilascia il tasto sinistro del mouse, spesso viene utilizzato per catturare la fine di un trascinamento. MouseMove invece viene scatenato quando il mouse viene mosso, spesso si mettono delle condizioni if, per catturare il movimento su un determinato oggetto HTML.

Nel modulo OCR queste tre funzioni vengono implementate per gestire creazione, movimento, resizing dei rettangoli e i tooltip su di essi.

La funzione mouseDown viene utilizzata per verificare se si sta per andare a disegnare, muovere o modificare la dimensione del rettangolo. Quando viene

scatenata si esegue un ciclo for che scorre il vettore `rects`. Se le coordinate del click sono contenute all'interno di uno degli oggetti del vettore allora si controlla se il click è avvenuto all'interno del quadratino per il ridimensionamento, in questo caso si predispongono le variabili per iniziare il dragging, altrimenti ci si predispone per muovere il rettangolo. In caso il click sia avvenuto al di fuori dei rettangoli costruiti, se si sta eseguendo il training allora vengono settate le variabili per costruire un rettangolo. Nel caso in cui non si verificasse nessuna di queste casistiche, non viene modificata alcuna variabile e non accade nulla.

La funzione `mouseMove` invece legge semplicemente i valori delle variabili settati dalla funzione `mouseDown`, in caso ci si trovi nella condizione in cui bisogna muovere un rettangolo esistente allora farà sì che le coordinate del rettangolo cambino mentre l'utente muove il mouse, in caso l'utente debba disegnare allora mostrerà a video il rettangolo mentre viene costruito, in caso sia stato premuto uno dei quadratini per il ridimensionamento l'utente potrà muovere il mouse per rendere il rettangolo più grande o più piccolo. Nel caso nessuna operazione sia concessa all'utente, quando il mouse verrà mosso sopra un rettangolo comparirà un tooltip contenente il dato della raccolta dati a cui fa riferimento

La funzione `mouseUp` invece è la meno complessa, semplicemente resetta le variabili che erano state impostate per consentire le operazioni all'utente.

Purtroppo non è stato trovato un modo intuitivo per concedere all'utente di eliminare un rettangolo tracciato male. Quello che viene fatto, è l'utilizzo dell'evento **`dblclick`** del canvas. Quando il mouse viene premuto due volte, se ci si trova in fase di esecuzione del training, se il clic avviene all'interno di un rettangolo, esso viene eliminato e viene richiesto all'utente di disegnarlo nuovamente. La cancellazione del singolo rettangolo viene concessa solo in fase di esecuzione del training, non al termine o dopo aver premuto il pulsante Carica Training. Questo perché se si permettesse di cancellare il singolo rettangolo anche una volta terminato il training, andrebbe eseguita in maniera perpetua una funzione ricorsiva che controlli sempre il numero dei rettangoli e quale viene cancellato, e questo comporterebbe un grande utilizzo di risorse. Per questa prima versione si è preferito dunque non concedere la cancellazione ma semplicemente il movimento e il ridimensionamento dei rettangoli dopo la conclusione del training e sul caricamento training, in caso di sviluppi futuri dell'applicativo verrà pensata una soluzione per concedere questa possibilità

4.4 Operazioni concesse

In questa sezione andremo a vedere quali sono le operazioni che l'utente può eseguire in fase di training. Sostanzialmente sono:

- Eseguire il training

- Caricare il training
- Annullare il training

4.4.1 Esecuzione training

Il training ovviamente si differenzia a seconda del sub-evento che stiamo trattando, perché i fonogrammi sono diversi proprio dal punto di vista strutturale. Fondamentalmente la funzione segue sempre la stessa struttura ma varia leggermente per alcuni casi particolari. Per questo motivo andiamo a vedere ad uno ad uno tutti i sub-eventi.

Training affluenze

Per quanto riguarda le affluenze, il modello utilizzato per tutte le parziali è sempre lo stesso, cambia solo l'orario scritto ma le coordinate dove si trovano i dati sono sempre le stesse. Le affluenze finali sono sempre divise in maschi, femmine, totale, se anche le parziali hanno questa divisione allora basterebbe un solo training, ma può capitare che per le parziali venga segnata solo l'affluenza totale e dunque vanno fatti due training differenti per i due modelli. Si è comunque pensato di creare due sezioni differenti che saranno presenti per tutte le tipologie di elezione esistenti.

Affluenze parziali

Per le affluenze parziali viene fatto un unico check, viene controllato se sono solo totali, e nel caso vi è un unico elemento nel vettore istruzioni che è relativo ai votanti totali, o se i votanti sono divisi in maschi femmine e totali, e in questo caso il vettore ha 3 elementi.

Viene richiesto di disegnare singolarmente i rettangoli, nel primo caso solo uno, nel secondo 3, e poi il training termina.

Affluenze finali

Le affluenze finali sono ancora più semplici delle parziali, perché il dato è sempre diviso in maschi, femmine e totali, dunque il vettore istruzioni viene precaricato senza effettuare nessun controllo, poi come nel caso prima descritto viene richiesto di disegnare i 3 rettangoli e il training termina.

Training Liste

Per quanto riguarda le liste il ragionamento di base è come quello fatto precedentemente. Vengono controllati i flag per capire quali campi sono presenti sul fonogramma per capire quali istruzioni dare all'utente. Un caso particolare è la

gestione dei voti alla lista e dei voti contestati. Si è visto che mediamente nelle elezioni vi è un alto numero di liste, dunque andare a far riquadrare i voti validi o contestati lista per lista sarebbe un'operazione onerosa in termini di tempo. Considerando di avere un numero ragionevole e non troppo elevato di liste ad esempio 10, e avendo sia la colonna voti validi che quella voti contestati avremo da fare 20 rettangoli, considerando un tempo medio di 5 secondi a rettangolo avremo $5s \cdot 20 = 100s$ cioè poco meno di 2 minuti, il che andrebbe in contrasto con il nostro obiettivo di rendere il training veloce e rapido.

Si è osservato che nonostante i fonogrammi siano diversi per ogni comune, tutti mettono le liste in verticale, in una colonna, tranne rari casi come ad esempio i referendum. Dunque si è deciso di mettere un box di input che viene precaricato con il numero di liste attese ma che l'utente può modificare e di richiedere all'utente di selezionare l'intera colonna riferita ai voti validi/contestati. Una volta tracciato il rettangolo l'algoritmo crea in base al valore nel box di input e della grandezza dell'area tracciata n rettangoli della stessa dimensione, in modo da dividere il rettangolo tracciato in tanti rettangoli più piccoli quanti indicati nel box di input. Così si risparmia tempo sul training, lo svantaggio è che bisogna essere molto precisi perché un errore porta a errori su tutti i rettangoli creati automaticamente. Proprio per questo problema si è deciso che per questi casi se viene eliminato uno dei rettangolini allora vengono eliminati tutti e bisogna rieseguire il comando. Invece per quanto riguarda resizing e spostamento viene applicato ai singoli rettangoli.

Training raggruppamenti

I raggruppamenti seguono la logica delle liste, anche questo sub-evento prevede che vi siano voti validi, voti al singolo raggruppamento e infine i voti contestati. Anche qui vi è un input box precaricato ma modificabile e viene richiesto di riquadrare l'intera colonna e in caso di cancellazione non viene cancellato il singolo rettangolo ma tutto l'insieme. Come per le liste resizing e drag sono relativi ai singoli rettangoli. Per ciò che riguarda gli altri tipi di dati all'interno dell'elezione viene data un'istruzione come nel caso delle affluenze parziali di riquadrare il singolo spazio del fonogramma dove sarà presente il dato.

Training preferenze

Come per raggruppamenti e liste anche le preferenze hanno una colonna dove devono essere inseriti i voti relativi ai singoli candidati nelle liste. Qui abbiamo un input box per inserire il numero della lista e un altro box per inserire il numero di candidati, questo perché il training va fatto per ogni lista e per ognuno dei suoi candidati.

Anche qui durante il training per quello che riguarda i voti ai membri della lista si richiede di tracciare un rettangolo che comprenda tutti gli elementi che poi verrà diviso in rettangoli più piccoli, tanti quanti specificati nell'input box.

4.4.2 Caricamento training

Il training può venire salvato, in caso di salvataggio le informazioni relative a dimensione, posizione, nome del dato a cui si riferiscono, dei rettangoli vengono immagazzinate nel database. Il training salvato può voler essere semplicemente visionato oppure modificato, senza doverlo rieseguire dall'inizio. In questo caso esiste il pulsante carica training che carica dal database tutti i rettangoli relativi a un singolo training. Questi non possono essere eliminati, ma possono essere spostati e ridimensionati. Una volta modificato il training può essere salvato nuovamente, andando a sovrascrivere il precedente.

4.4.3 Annullamento training

Il training può essere annullato sia mentre lo si sta facendo che dopo che è stato completato. Se si annulla mentre è in corso la procedura guidata, essa viene fermata e i rettangoli che erano già stati disegnati vengono eliminati. In caso il training si annulli dopo che è stato già completato, ovvero la procedura guidata è terminata, i rettangoli vengono eliminati e non sarà possibile ripristinarli a meno che non siano stati precedentemente salvati tramite il bottone salva training.

Capitolo 5

Progettazione OCR

In questo capitolo tratteremo come è stato pensato e progettato il modulo OCR all'interno della web app, partendo dalla scelta della tecnologia per poi passare al funzionamento logico del modulo.

5.1 Tecnologie utilizzate

La web app è realizzata tramite l'utilizzo di PHP, javascript e HTML esattamente come il sito di Eleweb. Per questo prototipo non viene integrata alcuna base dati in quanto i risultati del confronto tra ocr e training vengono solamente inseriti all'interno di alcune maschere di inserimento sull'applicativo e non salvati in maniera permanente sul database. L'invio e il salvataggio dei dati non viene modificato, dunque non si crea nessuna tabella aggiuntiva all'interno del database.

5.1.1 Jscanify

Jscanify [10] è una libreria open-source che utilizza openCV, la libreria di computer vision di Python e permette di scannerizzare un foglio a partire da una foto tramite javascript. Spiegheremo nel dettaglio successivamente come verrà utilizzata all'interno dell'applicativo. Il suo funzionamento è quello di prendere in input una foto contenente un foglio e l'ambiente esterno (ad esempio il tavolo su cui è poggiata), prende una serie di punti che sono i quattro angoli della foto e li utilizzerà per l'analisi prospettica successivamente. Dopo aver ottenuto i quattro punti trasforma l'immagine in una scala di grigi e applica un algoritmo di sfumatura gaussiana per ridurre il rumore poi binarizza l'immagine con l'algoritmo di Otsu, infine analizza tutti i contorni ottenuti e prende quello più grande che sarà il nostro foglio, esegue la trasformazione prospettica usando i punti salvati in precedenza e ottiene il foglio raddrizzato adattandolo a delle dimensioni ricevute come input.

L'algoritmo di Otsu presuppone che nell'immagine ci siano soltanto due elementi cromatici, il bianco e il nero e calcola la soglia migliore per separarli. Considerando che trasformiamo l'immagine in una scala di grigi l'algoritmo trova la soglia migliore per capire quale è il bianco.

Per un utilizzo ottimale di questa libreria l'immagine e lo sfondo devono avere un elevato contrasto e il foglio deve trovarsi su una superficie piatta

5.1.2 Scelta della tecnologia

I requisiti dell'applicativo erano chiari, ovvero andava trovato un tool in grado di fare da OCR che fosse rapido e affidabile. Purtroppo non si è riusciti a trovare nessuna libreria open source in grado di garantire entrambi questi aspetti. Per non andare a sovraccaricare il server durante le elezioni, dato che già le richieste che deve gestire sono numerose, si era pensato di utilizzare una soluzione che girasse lato client. Si è dunque trovato Tesseract.js [11] che è una libreria open source che nasce come Tesseract ed è stata sviluppata anche per javascript con questo nome. Questa libreria aveva il vantaggio di poter essere utilizzata direttamente tramite javascript e di non andare dunque a sovraccaricare il server. Dopo aver fatto numerose prove e osservato scrupolosamente i risultati ci si è accorti che questa libreria funzionava molto bene con il testo stampato, ma per ciò che riguardava il testo scritto a mano l'affidabilità diminuiva drasticamente fornendo risultati spesso completamente errati. Andare a lavorare su quella libreria e provare a fare training per far sì che riuscisse a comprendere bene il testo scritto a mano poteva essere una soluzione ma troppo onerosa in termini di tempo e risorse.

Per questo motivo ci si è trovati a dover cercare altre soluzioni che sono state trovate grazie alla libreria Doctr [12].

5.1.3 Doctr

Doctr è una libreria open source creata dall'azienda Mindee che permette di fare da OCR tramite due passaggi:

- Text detection
- Text recognition

E' una libreria Python che può sfruttare sia Tensorflow che con PyTorch che sono due librerie di machine learning e deep learning.

La text detection e la text recognition sono due operazioni che hanno fondamentale importanza all'interno di un sistema OCR e dunque è importante scegliere delle buone tecnologie che facciano questo lavoro e capire cosa sono e cosa cambia tra le due.

La text detection è un'operazione di computer vision che ha come obiettivo riconoscere la presenza del testo all'interno di un immagine. E' la prima operazione che viene fatta ed è qui che vengono evidenziati i bounding box, ovvero i rettangoli che contengono le parole all'interno del documento.

La text recognition invece basandosi sul dove si trova il testo va a capire cosa c'è scritto. E' dunque una fase che avviene successivamente rispetto a quella di detection.

Doctr permette di utilizzare varie tecnologie per queste due operazioni. Dopo più di 500 esperimenti, fornendo in input al sistema i fonogrammi e analizzando i risultati ottenuti si è optato per l'utilizzo di:

- DbNet: usato per la text detection
- CRNN per la text recognition

DbNet [13] è uno strumento che viene utilizzato per lo più per la detection real time ma che si è visto funziona benissimo anche su singoli frame e dunque immagini.

CRNN [14] viene usato per fare una end-to-end recognition di sequenze basate su immagini ed è trainabile, dunque per sviluppi futuri potrebbe essere modificata direttamente in azienda per aumentare l'efficacia nel riconoscimento parole.

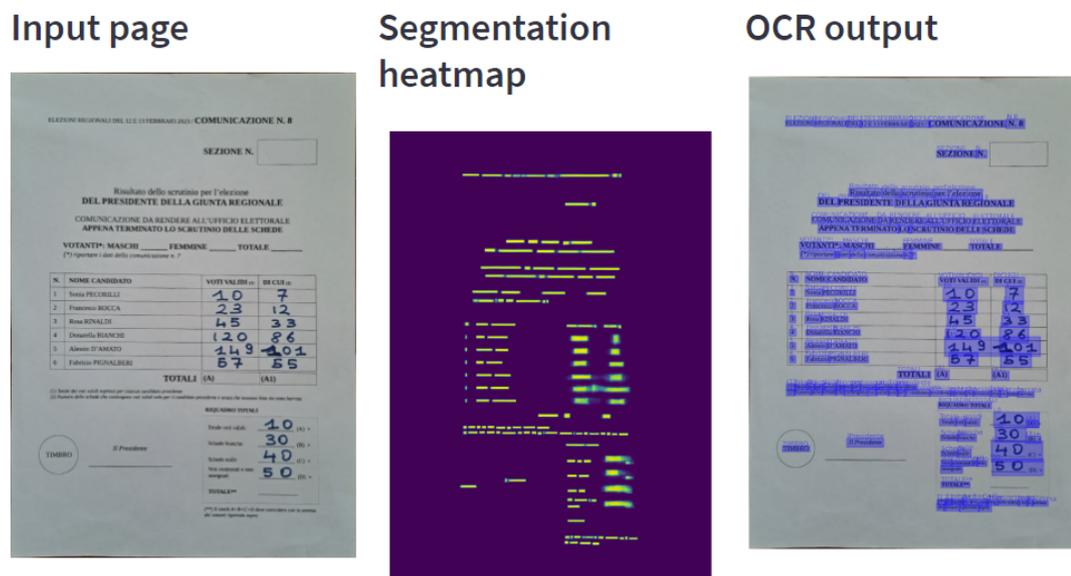


Figura 5.1: Esempio fasi elaborazione OCR

Nella figura 5.1 possiamo notare nella pratica che viene presa in input un'immagine, un fonogramma relativo ai candidati sindaco, viene fatta la segmentation heatmap tramite DbNet, sulla base dei box, che vengono fatti notare anche nell'immagine a fianco, CRNN riconosce il testo. Ognuno dei box colorati di blu nell'OCR output è descritto all'interno di un JSON tramite le sue coordinate e il valore della parola al suo interno. Il json è ciò che l'OCR restituisce.

5.2 Funzionamento OCR

Il primo problema che si è riscontrato nella progettazione del sistema, una volta definito che tipo di libreria utilizzare, è stato capire come e quando fare avvenire la scansione. Nel dettaglio la massima comodità si sarebbe ottenuta se l'operatore incaricato di scattare le foto avesse potuto inserire tutte le foto insieme da un unico punto e una volta inserite automaticamente si fosse rilevato sezione, tipo di sub-evento e dati. Il problema è che questo si è deciso di non farlo per motivi dovuti all'affidabilità del rilevamento. Sebbene l'OCR utilizzato sia affidabile, comunque non ha una percentuale di riconoscimento corretto del 100% e dunque la lettura errata di un dato sensibile come la sezione elettorale porta a salvare erroneamente dei dati per la sezione sbagliata, il che potrebbe provocare errori a catena. Si è dunque deciso che va inserito un foglio per volta, dopo che si è già selezionato il sub-evento e la sezione relative al fonogramma. Questo indubbiamente rallenta il processo, però fa sì che sia più difficile sbagliare a inserire dati, l'unico errore può essere fatto dall'operatore umano che sbaglia a selezionare la sezione.

5.2.1 Il problema delle dimensioni

Il nostro training è fatto su una scansione di un fonogramma e sul database vengono salvate le coordinate dei rettangoli disegnati sul foglio. Quando l'utente effettua la foto è molto improbabile che vada a farla in modo che inquadri tutto il foglio senza tagliarne neanche una parte e non inquadrando l'ambiente che lo circonda. Probabilmente o verrà fatta una foto che riprende solo parzialmente il foglio o una che riprende anche l'ambiente che lo circonda, facendo sì che il foglio non sia l'interezza della fotografia.

Entrambi questi casi conducono a un errore dell'applicativo e spieghiamo perché. Se la foto non inquadra solamente il foglio ma anche l'ambiente che lo circonda, allora le coordinate in cui si trovano i rettangoli del training potrebbero essere fuori del foglio come nella figura 5.2. Inoltre a seconda della vicinanza con cui la foto viene scattata il foglio risulterà più o meno grande, dunque non è possibile utilizzare questo tipo di fotografia per eseguire l'OCR.

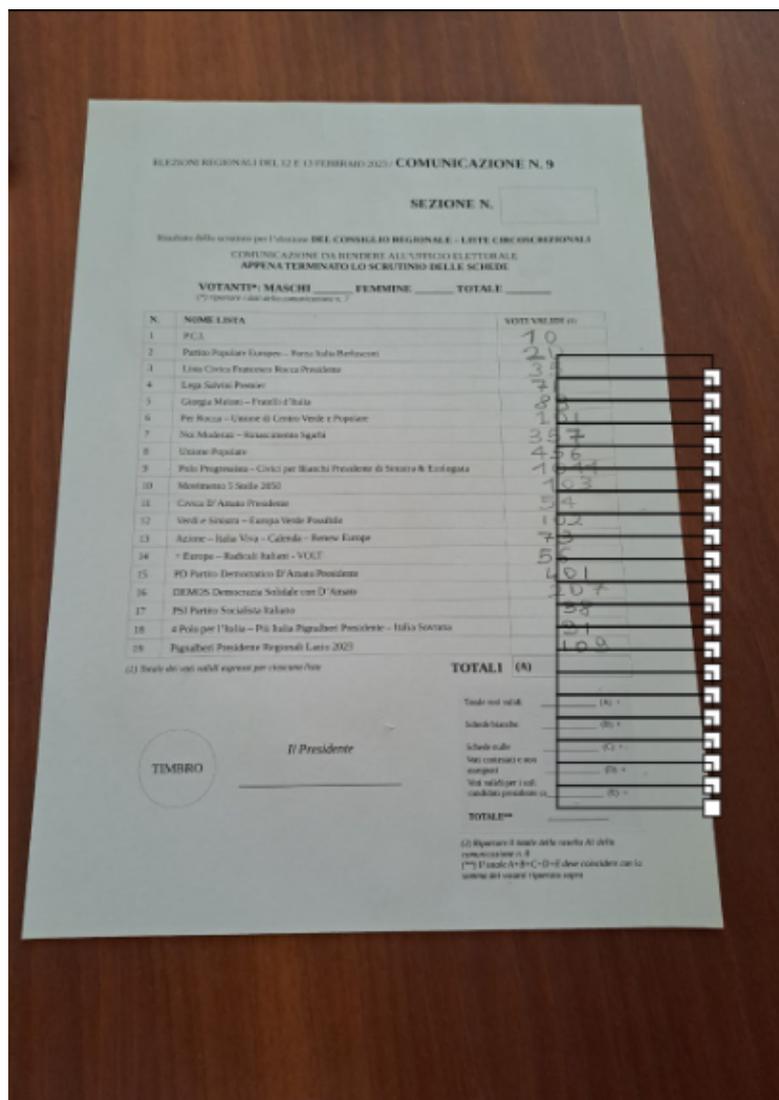


Figura 5.2: Confronto training con fonogramma più ambiente esterno

Se invece la foto riprende solo ed esclusivamente il fonogramma ma non viene inquadrato tutto e dunque una parte è esclusa dalla foto, avremo un problema relativo alla posizione dei rettangoli del training, che sono stati tracciati su un fonogramma intero, in questa maniera i rettangoli si troveranno spostati a seconda di quale parte del fotogramma è stato tagliato nella foto. Nella figura 5.3 si può osservare che il taglio si trova nella parte bassa del foglio, infatti le ultime righe non sono visibili, e dunque i rettangoli del training si trovano troppo in alto rispetto a dove dovrebbero essere.

ELEZIONI REGIONALI DEL 12 E 13 FEBBRAIO 2023 / COMUNICAZIONE N. 9

SEZIONE N. _____

Risultato dello scrutinio per l'elezione DEL CONSIGLIO REGIONALE - LISTE CIRCOSCRIZIONALI
 COMUNICAZIONE DA RENDERE ALL'UFFICIO ELETTORALE
 APPENA TERMINATO LO SCRUTINIO DELLE SCHEDE

VOTANTI: MASCHI _____ FEMMINE _____ TOTALE _____
 (*) riportare i dati della comunicazione n. 7

N.	NOME LISTA	VOTI VALIDI (1)
1	P.C.I.	10
2	Partito Popolare Europeo - Forza Italia Berlusconi	20
3	Lista Civica Francesco Rocca Presidente	35
4	Lega Salvini Premier	71
5	Giorgia Meloni - Fratelli d'Italia	89
6	Per Rocca - Unione di Centro Verde e Popolare	101
7	Noi Moderati - Rinascimento Sgarbi	357
8	Unione Popolare	456
9	Polo Progressista - Civici per Bianchi Presidente di Sinistra & Ecologista	1044
10	Movimento 5 Stelle 2050	103
11	Civica D'Amato Presidente	54
12	Verdi e Sinistra - Europa Verde Possibile	102
13	Azione - Italia Viva - Calenda - Renew Europe	73
14	* Europa - Radicali Italiani - VOLT	56
15	PD Partito Democratici D'Amato Presidente	401
16	DEMOS Democrazia Solidale con D'Amato	207
17	PSI Partito Socialista Italiano	38
18	4 Polo per l'Italia - Più Italia Pignatelli Presidente - Italia Sovrana	31
19	Pignatelli Presidente Regionali Lazio 2023	109
TOTALI (A)		
Totale voti validi (A) *		
Schede bianche (B) *		
Schede nulle (C) *		
Voti censurati e non assegnati (D) *		
Voti validi per i soli candidati presidente (E) *		
TOTALE**		

(1) Totale dei voti validi espressi per ciascuna lista

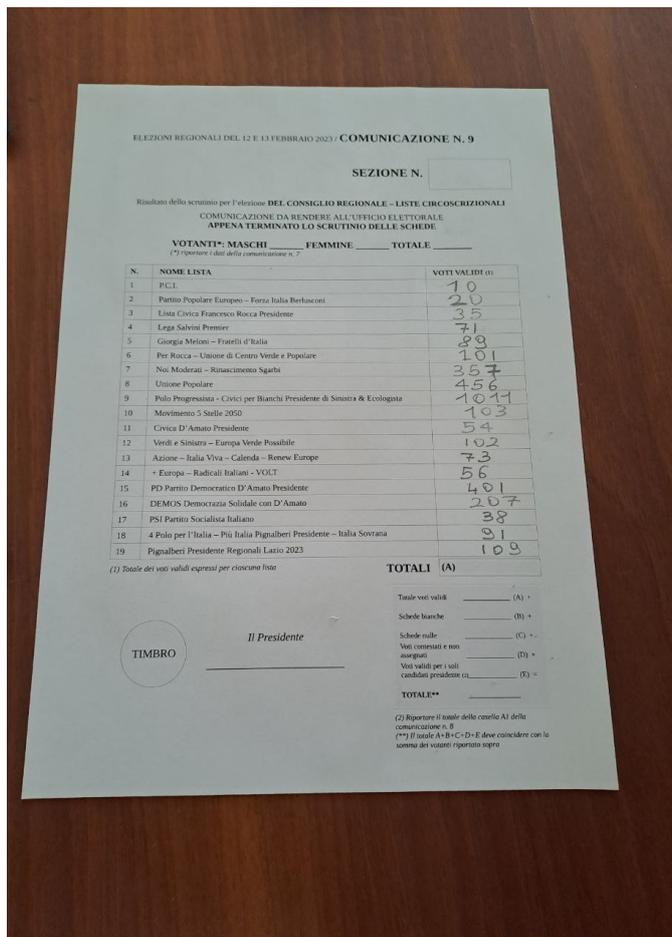
TIMBRO _____ Il Presidente _____

(2) Ripetere il totale della casella A1 della comunicazione n. 8

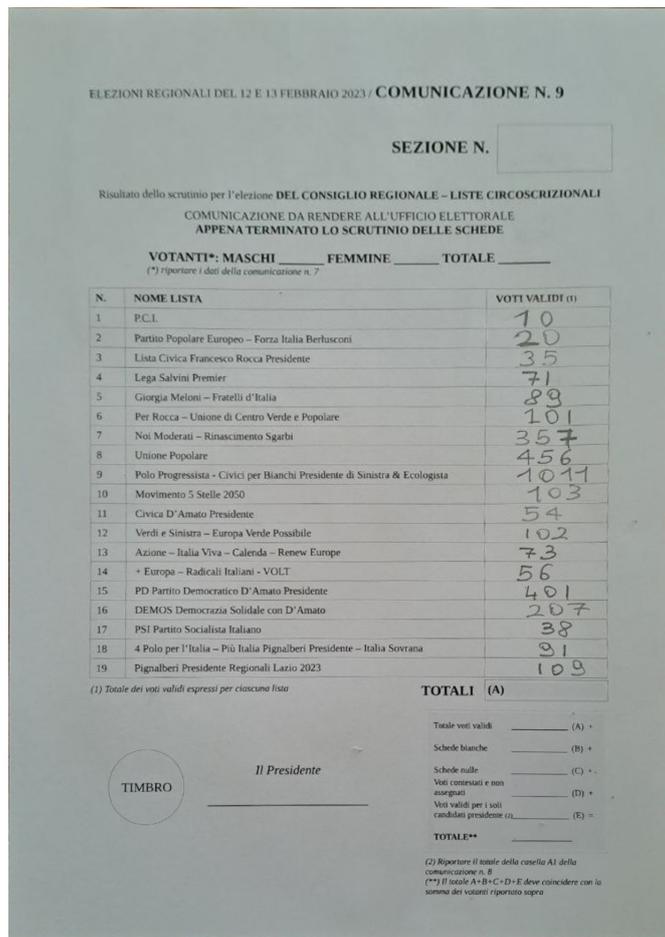
Figura 5.3: Confronto training con fonogramma tagliato

Per evitare questo problema si è deciso di utilizzare uno scanner integrato nell'applicativo che riceve in input una foto contenente il fonogramma e l'ambiente esterno come in figura 5.2 la taglia in automatico lungo i bordi del foglio.

Questo porta comunque a un problema, perché il foglio non viene sempre fotografato tenendo il cellulare perfettamente parallelo ad esso, ma si crea un angolo tra l'asse del foglio e quello del telefono. Più quest'angolo è grande, maggiore è l'inclinazione con cui il foglio viene fotografato. Lo scanner ci restituisce un foglio perfettamente dritto grazie alla trasformazione prospettica, e per fare ciò deve andare a cambiare l'asse, perdendo parte delle informazioni relative alla vera distanza tra le parole e alla dimensione del foglio. Questa sembrava comunque la migliore soluzione possibile e dunque si è deciso di percorrerla, tenendo conto di questo problema nell'algoritmo che valuta la sovrapposizione tra i bounding box e i rettangoli del training .



(a) Foto scattata dall'operatore



(b) Foto dopo l'operazione di scansione

Figura 5.4: Confronto tra la foto originale e quella scannerizzata

Dopo aver ragionato su tutte queste accortezze per rendere l'immagine il più possibile simile a quella inserita in fase di training è importante che prendiamo la foto scannerizzata e la elaboriamo in modo che abbia le stesse dimensioni dell'immagine di training. Per far comprendere questo facciamo l'esempio di un quadrato 10x10 e un rettangolo 10x8. Il primo è l'immagine usata per il training, il secondo è il risultato della scansione. Nel primo il punto a metà del foglio si trova in (0.5,0.5). Sul secondo (0.5,0.5) non si trova nello stesso punto in cui si trovava sul quadrato ma sarà un po' più in alto. Dunque se lasciassimo le dimensioni della foto scansionata invariate, il training sarebbe sfalsato. Per risolvere questo problema allora l'immagine scansionata viene fatta della stessa dimensione di quella del training. Basterebbe che il rapporto tra altezza e larghezza delle due immagini fosse uguale, ma per comodità poniamo uguali le dimensioni. Vi è un ultimo

problema che dipende da come i comuni strutturano i fonogrammi. Studiando i foglietti ed eseguendo l'algoritmo di text detection e text recognition ci si è accorti che se la dimensione delle celle all'interno delle quali vanno inseriti a mano i dati è troppo piccola, l'OCR fa fatica a riconoscere i caratteri. Come già detto il software che taglia il foglio può traslare la posizione delle celle nelle 4 direzioni. Se la dimensione di queste è limitata, uno spostamento di pochi pixel potrebbe comportare una lettura errata di tutto il foglio, non solo andando a leggere male i caratteri, ma anche andando a "ingannare" l'algoritmo di IoU.

5.2.2 L'algoritmo di confronto

La parte fondamentale di tutto il modulo OCR si trova in questo algoritmo. Come riportato precedentemente, per colpa della diversa risoluzione dell'immagine, del taglio applicato dallo scanner e dell'algoritmo per annullare l'inclinazione del foglio, il fonogramma che verrà usato per l'OCR non sarà perfettamente uguale a quello utilizzato per il training. E' dunque fondamentale studiare un algoritmo che permetta di intersecare i rettangoli sul database con i corrispondenti bounding box dati in output da Doctr.

L'algoritmo utilizzato è una rivisitazione di un algoritmo famoso chiamato **Intersection Over Union** (IoU) che calcola il grado di sovrapposizione di due box sulla base del rapporto tra l'area dell'unione e quella dell'intersezione tra i due box.

$$\text{IoU}(A, B) = \frac{\text{Area}(A \cup B)}{\text{Area}(A \cap B)}$$

L'algoritmo utilizzato per questo applicativo si basa su due vettori, il primo chiamato **rectsByDb** che contiene le coordinate relative ai rettangoli definiti in fase di training, e **rectsByOcr** che invece contiene tutte le informazioni sui box trovati da Doctr. Si utilizzano due cicli for. Per ogni elemento del primo vettore, si verifica quale tra gli elementi del secondo, ovvero dei box dell'OCR, ha il suo punto medio all'interno del rettangolo del ciclo interno. Abbiamo scelto questa strategia perché come si vede dalla figura 5.5 i rettangoli del training sono traslati leggermente rispetto al foglio e non rispettano le celle reali stampate sul foglio. Sperimentalmente si è visto che il punto medio è il modo migliore per fare la corretta valutazione del box che matcha correttamente il corrispettivo rettangolo tracciato in fase di training.

Se nessun box dell'ocr ha il punto medio contenuto all'interno di uno dei rettangoli del training allora si utilizza l'IoU in modo da prendere il box che meglio interseca il rettangolo e che quindi con maggiore probabilità corrisponde ad esso. Ovviamente erano molti gli algoritmi che si potevano utilizzare in questa frase, ma

tramite un elevato numero di esperimenti si è compreso che questa metodologia utilizzata è la migliore.

Per essere certi che un box non venga preso due volte, una volta che un box è stato utilizzato per un determinato dato, esso viene marcato con un flag, in modo da riconoscere che è già stato utilizzato e non includerlo più tra i possibili box che meglio intersecano un rettangolo.

ELEZIONI REGIONALI DEL 12 E 13 FEBBRAIO 2023 / COMUNICAZIONE N. 9

SEZIONE N.

Risultato dello scrutinio per l'elezione DEL CONSIGLIO REGIONALE - LISTE CIRCOSCRIZIONALI
 COMUNICAZIONE DA RENDERE ALL'UFFICIO ELETTORALE
 APPENA TERMINATO LO SCRUTINIO DELLE SCHEDE

VOTANTI*: MASCHI _____ FEMMINE _____ TOTALE _____
 (*) riportare i dati della comunicazione n. 7

N.	NOME LISTA	VOTI VALIDI (1)
1	P.C.I.	10
2	Partito Popolare Europeo - Forza Italia Berlusconi	20
3	Lista Civica Francesco Rocca Presidente	35
4	Lega Salvini Premier	71
5	Giorgia Meloni - Fratelli d'Italia	89
6	Per Rocca - Unione di Centro Verde e Popolare	101
7	Noi Moderati - Rinascimento Sgarbi	357
8	Unione Popolare	456
9	Polo Progressista - Civici per Bianchi Presidente di Sinistra & Ecologista	1044
10	Movimento 5 Stelle 2050	103
11	Civica D'Amato Presidente	54
12	Verdi e Sinistra - Europa Verde Possibile	102
13	Azione - Italia Viva - Calenda - Renew Europe	73
14	+ Europa - Radicali Italiani - VOLT	56
15	PD Partito Democratico D'Amato Presidente	401
16	DEMOS Democrazia Solidale con D'Amato	207
17	PSI Partito Socialista Italiano	38
18	4 Polo per l'Italia - Più Italia Pignalberi Presidente - Italia Sovrana	31
19	Pignalberi Presidente Regionali Lazio 2023	109
TOTALI (A)		

(1) Totale dei voti validi espressi per ciascuna lista

Totale voti validi _____ (A) +
 Schede bianche _____ (B) +
 Schede nulle _____ (C) +
 Voti censurati e non assegnati _____ (D) +
 Voti validi per i soli candidati presidente (1) _____ (E) =
TOTALE** _____

(2) Riportare il totale della casella A1 della comunicazione n. 8
 (***) Il totale A+B+C+D+E deve coincidere con la somma dei votanti riportata sopra

TIMBRO Il Presidente _____

Figura 5.5: Esempio sfalsatura training rispetto a immagine ocr

5.2.3 Il rilevamento degli errori

L'OCR non è infallibile, spesso può errare nella lettura di un singolo carattere o addirittura di un'intera parola. Alla fine dell'iterazione su tutti gli elementi contenuti all'interno del vettore `rectsByOcr` ci troveremo un insieme di elementi all'interno di un altro vettore, detto **rects** che contiene dati in questo formato: `{x, y, w, h, value, id_campo, confidence}` dove `x,y,w,h` sono le coordinate del rettangolo, `value` è il valore letto dall'OCR, `id_campo` si riferisce al valore `id_campo` memorizzato nel database che indica il nome del dato a cui l'elemento si riferisce e infine `confidence` che è la percentuale di affidabilità con cui l'OCR ha riconosciuto la parola. Sappiamo che tutti i dati che dovremo leggere sono numeri, è però molto probabile che anche per colpa di una calligrafia non ottimale da parte dell'operatore, l'OCR faccia confusione tra numeri e tra numeri e lettere. Sulla prima tipologia di errore non possiamo intervenire mentre sulla seconda si, riportiamo dunque tramite la tabella 5.1 le correzioni che facciamo in caso venga trovata una lettera al posto di un numero. Queste correzioni sono frutto sia di uno studio teorico sui casi più comuni di errore degli OCR, sia di uno studio sperimentale sulla base degli errori trovati nei numerosi test eseguiti. Ovviamente se viene fatta una correzione la `confidence` della lettura viene messa a 0, perché sicuramente la lettura fatta è stata errata

NUMERO	LETTERE
0	O,U
1	I, l
2	Z
3	e
4	A ,H,L
5	S
6	G,B
7	T
8	B,R,&
9	J

Tabella 5.1: Tabella errori in lettura tra lettera e numero dell'OCR

E' importante capire che l'OCR serve per velocizzare il procedimento di inserimento dati, dunque se vengono inseriti molti dati con una `confidence` troppo bassa probabilmente l'operatore incaricato dell'inserimento dati dovrà impiegare molto tempo a verificare tutti i numeri e cambiarli. Quello che si fa è dunque inserire delle soglie e in caso non si raggiungano non verrà mostrato a video il risultato dell'OCR ma verrà mostrato un messaggio d'errore.

Il primo controllo che si fa una volta che si ha completato la scansione di tutti i rettangoli si guarda il numero di intersezioni trovate, in caso siano 0 si segnala all'utente che non è stato trovato nulla.

Per questa versione sperimentale del programma abbiamo deciso di considerare accettabile un dato se la confidence è maggiore del 60%. Se almeno il 50% degli elementi all'interno del vettore `rects` supera la soglia allora stampiamo a video tutti i valori, compresi quelli che hanno un valore di confidence minore del 60%, altrimenti si mostra un messaggio d'errore.

Ovviamente tutte le soglie sono configurabili a seconda delle esigenze del singolo comune, nella fase pre-elettorale vengono definiti questi valori per venire incontro alle esigenze del cliente.

Capitolo 6

Prototipo Training

Tramite lo studio del dominio si sono acquisite delle conoscenze da utilizzare per la creazione del prototipo. Per quanto riguarda la User Interface che è l'interfaccia grafica del nostro applicativo, si è fatta una scelta consapevole di seguire il layout e i pattern già esistenti all'interno di Eleweb. Questo perché volevo fosse chiaro che quella creata è un'estensione dell'applicativo, non una parte a se stante, era fondamentale che l'utente si ritrovasse in qualcosa che già conosce. Inoltre utilizzando pattern e stili già esistenti si va a semplificare lo sviluppo, si è dovuto pensare solamente a come disporre gli elementi nello spazio, non a come progettarli.

Queste considerazioni, unite al fatto che l'applicativo è sviluppato con HTML e CSS hanno portato ad avere un'interfaccia semplice, mirata più alla user experience piuttosto che alla user interface.

Partiamo dal presupposto che tutti gli esempi grafici riportati in questa sezione sono relativi alle elezioni amministrative del 2023. Abbiamo scelto questo tipo di elezione perché comprende tutti i sub-eventi e dunque può mostrare tutte le casistiche. Per ciò che riguarda altre tipologie di evento elettorale, può variare il nome e il numero dei sub-eventi ma lo schema di base è lo stesso.

Eleweb contiene un menù laterale attraverso il quale si può accedere a tutte le pagine, ovviamente anche quella di training è stata inserita, all'interno del sottomenù "Configurazione Elezione" in quanto è in questa fase che va eseguito.

6.1 Cruscotto dei sub-eventi

Abbiamo detto che il training va fatto per ogni sub-evento e che è possibile sia inserirne uno nuovo che caricarne uno precedentemente salvato. E' dunque fondamentale che l'utente abbia un'idea chiara di quale sub-evento stia trainando e abbia chiaro come selezionarlo. L'applicativo usa spesso degli elementi HTML chiamati **fieldset** che non sono altro che dei contenitori per raggruppare altri

elementi al loro interno e ai quali si può aggiungere un tooltip per far capire all'utente che cosa raggruppano.

Si è scelto di utilizzare questo tipo di elemento per racchiudere tutti i bottoni con i nomi dei sub-eventi.

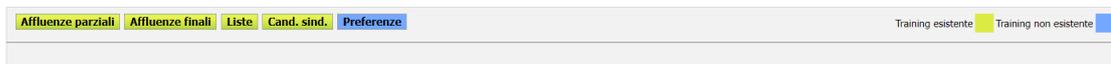


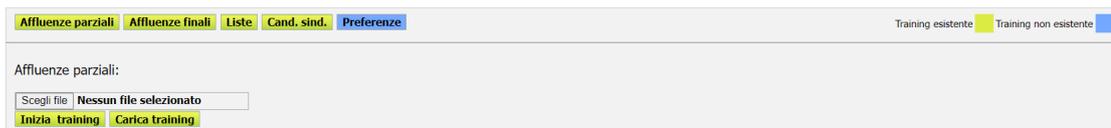
Figura 6.1: Fieldset pulsanti sub-eventi

Come si può notare i pulsanti hanno colori diversi. Il verde è quello di default per i bottoni in Eleweb e si è scelto di utilizzarlo per indicare quei sub-eventi per il quale è stato già eseguito il training. Quelli che ancora non hanno un training salvato sul database invece sono in blu. Inizialmente si era scelto di utilizzare il colore rosso, questa scelta è stata scartata per non dare l'idea che ci fosse qualche errore, o comunque qualcosa di sbagliato, perché spesso associamo il colore verde all'ok e il rosso all'errore. Per far capire meglio all'utente a cosa corrispondono il blu e il verde si è inserita una legenda sulla destra.

Una volta che si seleziona il sub-evento anche il fieldset sottostante ai bottoni si popola. A seconda del pulsante cliccato vengono visualizzati elementi differenti, per questo andremo a vedere uno a uno come vengono visualizzati.

6.1.1 Affluenze

Per le affluenze gli elementi sono gli stessi sia per quelle parziali che per quella finale, cambia solamente il testo della label visualizzata. Abbiamo semplicemente il pulsante inizia training, e nel caso dalla figura 6.3, visto che è già inserito un training, il bottone carica.



I dati qui riportati non sono da considerarsi ufficiali
© 1999 - 2023 - Powered by
Gruppo Pro Logic - info@pro-logic.it

Figura 6.2: Affluenze parziali

Notiamo che abbiamo un selettore di file, per scegliere il file da visualizzare nel canvas sottostante, ovviamente è stato configurato in modo che possano essere

selezionati solo file di tipo immagine e non altre tipologie. La label può sembrare ridondante ma come vediamo in figura 6.2 una volta premuto sul pulsante, se poi si clicca altrove, non abbiamo più modo di sapere quale sub-evento è stato selezionato, dunque è importante scriverlo

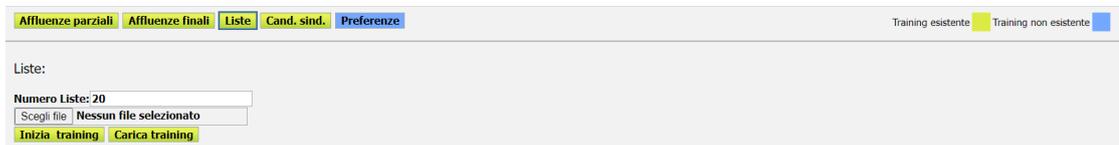


I dati qui riportati non sono da considerarsi ufficiali
© 1929 - 2023 - Powered by
Gruppo Pro Logic - info@pro-logic.it

Figura 6.3: Affluenze finali

6.1.2 Liste

Questo sub-evento si differenzia dalle affluenze perché viene richiesto di inserire il numero di liste, dunque vi è l'aggiunta di un input box che come detto nei capitoli precedenti viene precaricato con il numero di liste atteso, ma che è modificabile. Il resto del menù è uguale al precedente.



I dati qui riportati non sono da considerarsi ufficiali
© 1929 - 2023 - Powered by
Gruppo Pro Logic - info@pro-logic.it

Figura 6.4: Liste

6.1.3 Raggruppamenti

I raggruppamenti, come le liste, prevedono un input box per indicare il numero di candidati sindaco o presidente per l'elezione. In questo caso la label si riferisce al candidato sindaco perché si tratta delle elezioni amministrative. Come si nota in figura 6.5 l'input box si può modificare sia tramite tastiera che attraverso l'utilizzo dei pulsanti sulla destra.



Figura 6.5: Raggruppamenti

6.1.4 Preferenze

Le preferenze sono il sub-evento più complicato da gestire. Per organizzare l'interfaccia grafica si è preso come riferimento un'altra pagina dell'applicativo relativo all'inserimento dati delle preferenze in fase elettorale. Ogni lista ha un numero variabile di candidati, che sono coloro che vengono indicati nelle preferenze. L'obiettivo era far sì che l'utente avesse un modo veloce e intuitivo per tenere traccia delle liste per cui il training delle preferenze è già stato fatto. Si è scelto di inserire due input box: uno relativo al numero della lista, l'altro al numero di candidati, inoltre ogni lista è rappresentata da un bottone con il suo numero, che segue il codice colore dei pulsanti del sub-evento, in questa maniera è possibile tenere traccia dei training eseguiti. Schiacciando sul bottone verranno cambiati i due input box con numero della lista e numero di candidati.



Figura 6.6: Preferenza con training esistente

Nella figura 6.6 notiamo che si è cliccato sul pulsante relativo alla lista numero 3, che ha 25 candidati ed è di colore verde dunque esiste un training infatti è presente il pulsante Carica training.



Figura 6.7: Preferenza senza training esistente

Nella figura 6.7 invece notiamo che il bottone è blu, perché non esiste nessun training sul database e dunque non vediamo il pulsante Carica training.

Il bottone preferenze nel fieldset dei sub-eventi diventerà di colore verde una volta che saranno stati caricati i training relativi alle preferenze di **tutte** le liste

6.2 L'inizio del training

Per cominciare il training dobbiamo scegliere un'immagine tramite il file picker. Come già anticipato è possibile selezionare solo file immagine. In caso si provi a iniziare il training senza aver caricato un'immagine viene mostrato un alert di errore. Per le comunicazioni all'utente si è deciso di utilizzare gli alert perché questo è il metodo utilizzato dalla piattaforma. In figura 6.8 vediamo l'errore che viene mostrato se non si seleziona un'immagine. Una volta caricata, si può cliccare sul bottone Inizia training e da lì inizierà la procedura guidata.

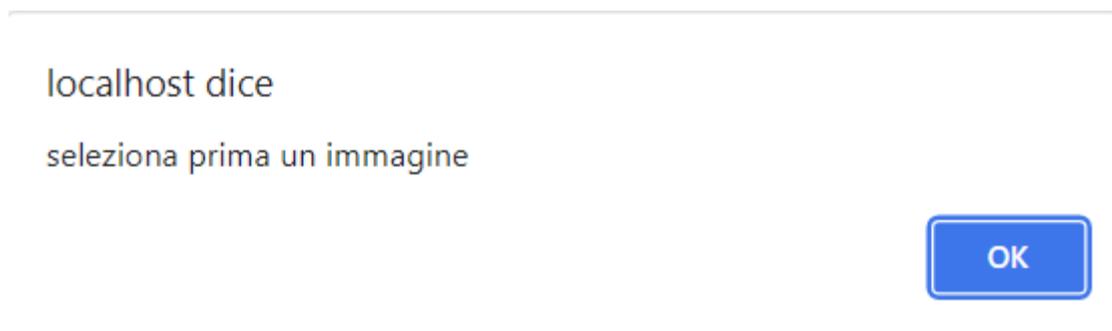
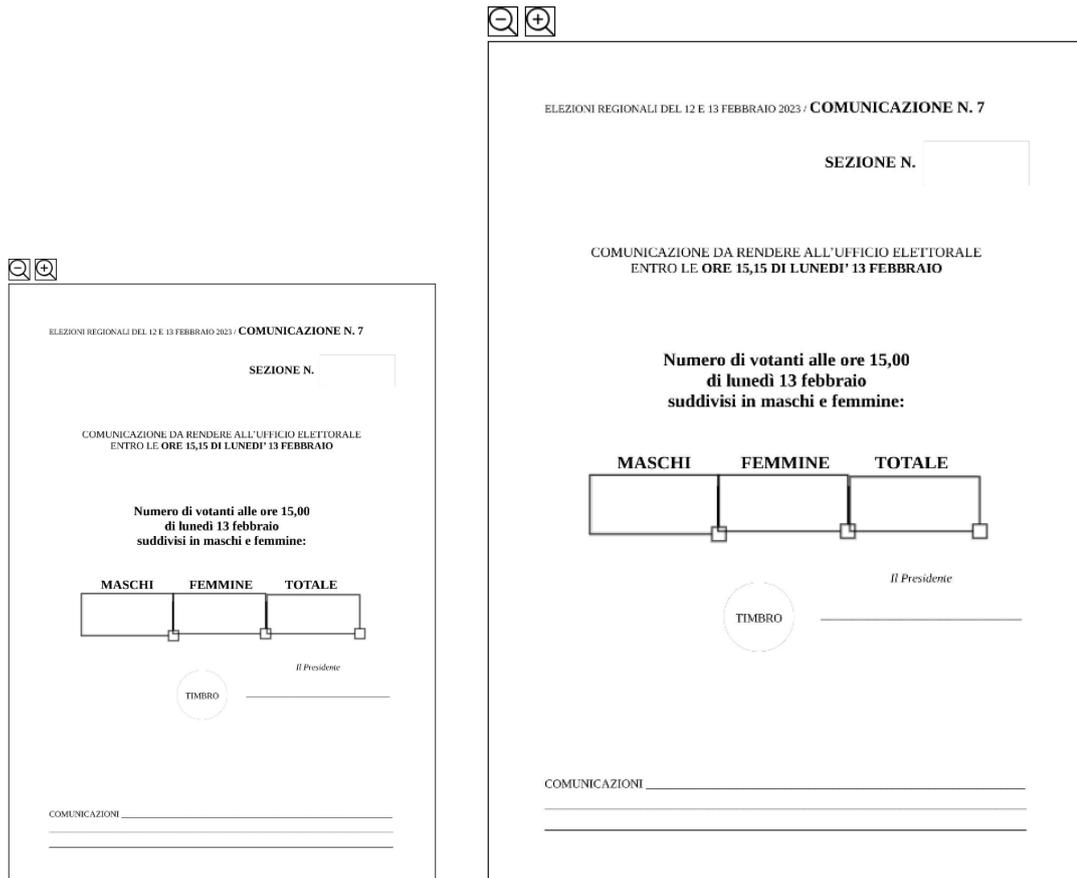


Figura 6.8: Alert errore inserimento immagine

Quando l'immagine viene caricata è inserita all'interno di un canvas, che come detto è un elemento HTML utilizzato per disegnare forme e immagini direttamente all'interno della pagina web, dato che l'immagine avrà probabilmente grosse dimensioni, esse vengono inizialmente dimezzate per non essere troppo grande per

la pagina. Sono comunque stati inseriti due bottoni, che si trovano al di sopra del canvas, raffigurati attraverso due icone, di zoom-in e zoom-out per ingrandire o rimpicciolire l'immagine. Ovviamente viene ridimensionato tutto il contenuto del canvas, compresi i rettangoli disegnati dall'utente in fase di training, per rendere ciò che si è fatto coerente con la nuova dimensione. Ovviamente ogni volta che si preme uno dei due bottoni l'immagine viene ridimensionata, per fare ciò si è scelto come convenzione di variare la dimensione ogni volta del 10%



(a) Canvas dopo zoom-out

(b) Canvas dopo zoom-in

Figura 6.9: Effetto dei bottoni zoom-in e zoom-out sul canvas

Una volta selezionata l'immagine si può iniziare il training tramite il bottone **Inizia training**. Verrà visualizzato un alert con le indicazioni su come eseguire la procedura guidata. Come indicato in figura 6.10 bisogna riquadrare ciò che ci viene detto e una volta disegnato un rettangolo sarà possibile spostarlo, ridimensionarlo ed eliminarlo.

localhost dice

training iniziato, segui i comandi **riquadrandolo** cio' che ti viene detto di selezionare.

Durante il training sarà possibile:

- **spostare**
- **ridimensionare**
- **eliminare**

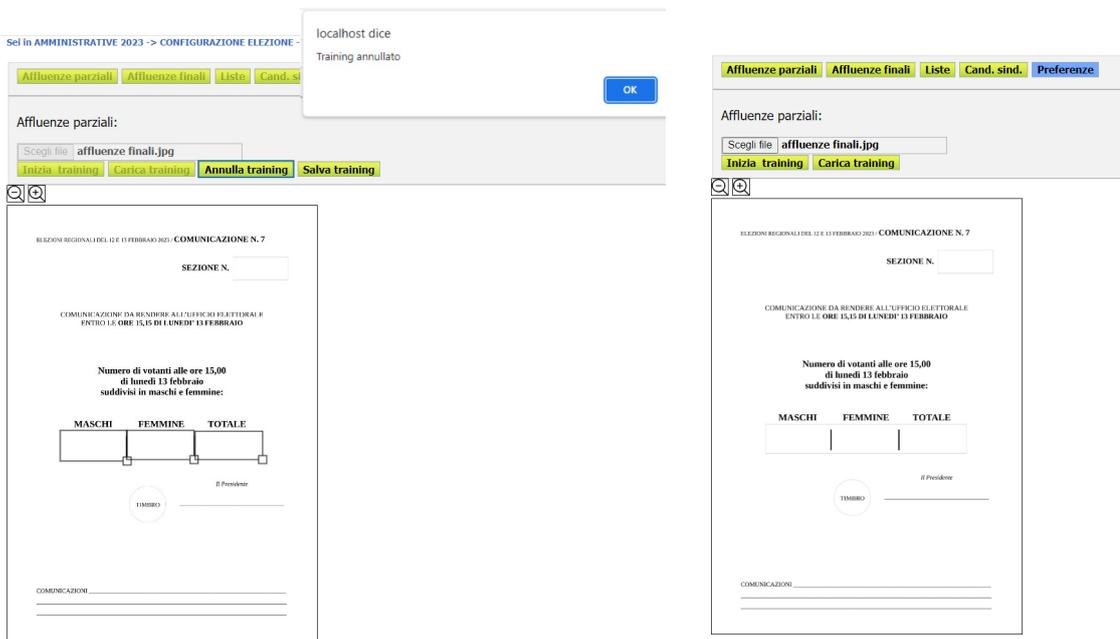
i rettangoli disegnati



Figura 6.10: Alert istruzioni training

Qualora non si voglia iniziare un nuovo training ma caricare quello salvato sul database bisogna premere il bottone Carica training, non verrà visualizzato nessun alert e semplicemente verranno visualizzati sul canvas i rettangoli.

Quando si comincia o viene caricato il training, vengono disattivati tutti i bottoni di selezione sub-evento e il file-picker, però compare a video il bottone annulla training, che elimina tutti i rettangoli creati e mostra a video un alert con la scritta "Training annullato".



(a) Bottone annulla training premuto

(b) Training annullato

Figura 6.11: Esempio di annullamento training

6.3 La visualizzazione delle istruzioni

Dopo che si è premuto il bottone Inizia training, si entra nella funzione principale del programma. In questa fase è importante che l'utente capisca perfettamente quale è il rettangolo che deve andare a tracciare e in caso si distraiga e non ricordi più l'istruzione ricevuta possa recuperarla in qualche modo.

Durante lo sviluppo del modulo OCR sono stati ideati 3 diversi metodi di visualizzazione delle istruzioni, ognuno dei quali ha i suoi pro e i suoi contro, andiamo a vedere brevemente tutti e tre.

6.3.1 Visualizzazione tramite alert

Questo metodo è stato il primo sviluppato e consiste in un messaggio mostrato tramite alert. Ogni istruzione viene notificata in questa maniera, andiamo a vedere i pro e i contro. I principali vantaggi di questa soluzione sono il fatto che l>alert è il metodo standard utilizzato all'interno dell'applicativo per mostrare dei messaggi, dunque l'utente ha familiarità, inoltre se il canvas viene ingrandito troppo può succedere che si sia costretti a scrollare la pagina verticalmente durante il training e quindi alcuni elementi come i fieldset rischiano di venire esclusi dalla porzione

visibile di pagina. L'alert è un elemento che compare sempre in sovrimpressione, a prescindere da quale sia la porzione di pagina visibile e dunque viene sempre notato dall'utente. Gli svantaggi riguardano il fatto che ogni volta che viene mostrata un'istruzione bisogna premere il pulsante "OK" altrimenti si è impossibilitati a compiere altre operazioni all'interno della pagina e questo è molto scomodo soprattutto per training che richiedono di disegnare molti rettangoli. Inoltre una volta che si preme su "OK" l'alert scompare e non è più possibile andare a reperire l'istruzione. Si era pensato di introdurre un ulteriore bottone che quando premuto andava a mostrare nuovamente l'ultimo messaggio visualizzato. Visti i numerosi punti a sfavore di questa soluzione si è deciso di virare su altre.

6.3.2 Visualizzazione tramite label

Successivamente all'idea dell'alert si era pensato un metodo che andava a risolvere i problemi della soluzione precedente. Si sarebbe introdotta una label al di sotto dei bottoni inizia training e carica training, in questa maniera non bisognava premere nessun bottone, semplicemente quando si disegnava o eliminava un rettangolo la label cambiava con l'istruzione successiva. I punti a favore di questo metodo sono dunque sicuramente la velocità di esecuzione di tutto il training e il fatto che la label rimane a video fino al completamento di tutte le istruzioni, dunque se ci si distrae o si dimentica l'istruzione basta guardare cosa vi è scritto per capire che cosa bisogna riquadrare. Lo svantaggio di questa soluzione è che se si scrolla la pagina verso il basso e il fieldset non è più visibile, neanche l'istruzione è visibile e dunque bisogna sempre scrollare verso l'alto e verso il basso per andare a visualizzare l'istruzione successiva e disegnare il rettangolo a lei relativo. Per eliminare questo problema si è pensato di rendere i fieldset fissi sullo schermo, facendo scorrere il canvas in modo che scrollato verso il basso, esso andasse a scomparire sotto i fieldset, ma questa soluzione portava ad altri problemi di sfasamento delle coordinate dei rettangoli, dunque intraprenderla avrebbe portato alla rielaborazione di tutto il training compreso l'algoritmo di costruzione, manipolazione e visualizzazione dei rettangoli, dunque anche questa soluzione è stata scartata.

6.3.3 Visualizzazione tramite elenco laterale

La soluzione che è stata scelta come ottimale e definitiva prevede l'inserimento di un elenco di istruzioni a lato del canvas. Vengono riportate in grigio chiaro tutte le istruzioni che l'utente deve seguire per completare il training. L'istruzione corrente è scritta in nero, dunque l'utente non solo ha sempre chiaro quale istruzione sta eseguendo, ma anche quali sono state le precedenti e quali saranno le successive. In questa maniera qualora l'utente vada a scorrere la pagina l'elenco di istruzioni, essendo collocato a metà del canvas rimane sempre visibile, ed è dunque consultabile

in ogni momento. Anche questa soluzione presenta degli svantaggi, ad esempio si va a introdurre un elemento grafico più ingombrante di una singola label messa in testa al canvas o di un alert. Un esempio di questo problema lo vediamo quando il canvas viene ingrandito troppo tramite il pulsante di zoom, in questo caso c'è il rischio che l'elenco a lato finisca troppo sulla destra e si venga a creare uno scorrimento laterale della pagina. Questo caso comunque è abbastanza remoto perché non ci sarà mai bisogno di lavorare così precisi da andare a zoommare così tanto il canvas, dunque tra tutte le soluzioni pensate e sviluppate è sembrata la migliore.

The image shows a screenshot of a form titled "ELEZIONI REGIONALI DEL 12 E 13 FEBBRAIO 2023 / COMUNICAZIONE N. 7". It includes a field for "SEZIONE N.", a deadline of "ENTRO LE ORE 15,15 DI LUNEDÌ 13 FEBBRAIO", and a section for reporting the number of voters at 15:00 on February 13th, divided by gender. The form has three columns: "MASCHI", "FEMMINE", and "TOTALE". A yellow callout box on the right contains the following instructions:

- seleziona il numero di votanti maschi
- seleziona il numero di votanti femmine
- seleziona il numero di votanti totali

Figura 6.12: Istruzioni affluenze finali

6.4 La forma del cursore

Una volta terminato l'applicativo, ci si è accorti che effettivamente per l'utente era difficile capire quali operazioni effettuare e quando poterle fare. Si è pensato di cambiare la forma del cursore a seconda della situazione per rendere maggiormente

intuitiva l'UX. Per capire quale forma dare al cursore in quale situazione ci si è ispirati ad alcuni software esistenti quali ad esempio Paint, noto programma di grafica.

All'interno del nostro applicativo servono dei cursori per le seguenti casistiche:

- indicare che è impossibile eseguire operazioni sul canvas
- disegnare i rettangoli
- ridimensionare i singoli rettangoli
- spostare i singoli rettangoli

Durante l'utilizzo dell'applicativo ci si trova in una situazione in cui viene caricata la foto, che appare dunque a schermo, ma ancora non si è premuto nessun bottone per iniziare o caricare il training. In questo caso non si possono eseguire operazioni sul canvas, per queste situazioni si è scelto di utilizzare un tipo di cursore della categoria drag & drop chiamato *not-allowed*.

Quando incomincia il training, possiamo avere diverse forme di cursore. Se siamo in una zona del canvas dove non vi è tracciato alcun rettangolo abbiamo una forma chiamata *crosshair*. Se invece siamo all'interno del rettangolo abbiamo un tooltip in cui vi è scritto per che dato è stato disegnato e una forma del cursore chiamata *move*.

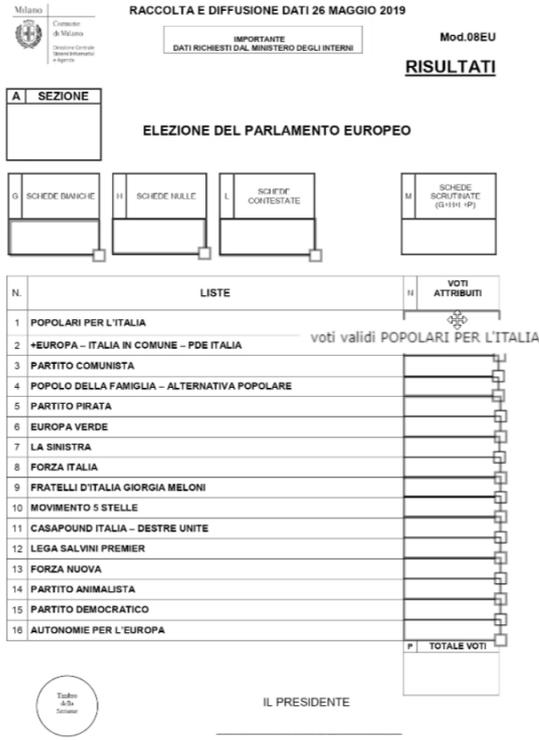
Se ci troviamo nel quadratino in basso a destra del rettangolo, chiamato in gergo tecnico maniglia di dimensionamento o maniglia di controllo, che è stato disegnato per permettere il resizing, allora il cursore assume la forma *nwse-resize*. Una volta terminato il resizing, nel caso il rettangolo sia stato dimensionato in maniera che la maniglia di controllo che prima si trovava in basso a destra si trova in un altro angolo della forma, allora viene riposizionata nella posizione originale, questo per comodità nella gestione dell'applicativo. Si è scelto di mettere una singolo maniglia per ridimensionare e non una per angolo per avere un'interfaccia più pulita, troppe avrebbero fatto sì che specialmente per liste e preferenze, che hanno molti rettangoli, si ottenesse un'area troppo affollata di elementi, creando difficoltà nel muovere o ridimensionare il rettangolo quello desiderato.

Quando invece terminiamo il training o si è preme caricamento training, se ci troviamo sul canvas, al di fuori dei rettangoli, il puntatore è quello di default.

Notiamo che per quanto riguarda l'operazione di cancellazione del singolo rettangolo non esiste nessun tipo di cursore, questo perché come specificato nel capitolo 4 non è stato trovato un metodo intuitivo per questa operazione. Quello che viene richiesto è di fare doppio click sopra il rettangolo che si vuole ridisegnare, a cui il sistema risponderà con l'eliminazione dell'elemento ed un alert che specifica che è stato eliminato.

crosshair	
resize	
not-allowed	
move	

Tabella 6.1: Tabella forme cursore



RACCOLTA E DIFFUSIONE DATI 26 MAGGIO 2019
Mod.08EU
RISULTATI

ELEZIONE DEL PARLAMENTO EUROPEO

N.	LISTE	N	VOTI ATTRIBUITI
1	POPOLARI PER L'ITALIA		
2	*EUROPA - ITALIA IN COMUNE - PDE ITALIA		
3	PARTITO COMUNISTA		
4	POPOLO DELLA FAMIGLIA - ALTERNATIVA POPOLARE		
5	PARTITO PIRATA		
6	EUROPA VERDE		
7	LA SINISTRA		
8	FORZA ITALIA		
9	FRATELLI D'ITALIA GIORGIA MELONI		
10	MOVIMENTO 5 STELLE		
11	CASAPOUND ITALIA - DESTRE UNITE		
12	LEGA SALVINI PREMIER		
13	FORZA NUOVA		
14	PARTITO ANIMALISTA		
15	PARTITO DEMOCRATICO		
16	AUTONOMIE PER L'EUROPA		
		P	TOTALE VOTI

IL PRESIDENTE _____



ELEZIONI REGIONALI DEL 12 E 13 FEBBRAIO 2023 / COMUNICAZIONE N. 7

SEZIONE N. _____

COMUNICAZIONE DA RENDERE ALL'UFFICIO ELETTORALE
 ENTRO LE ORE 15,15 DI LUNEDI' 13 FEBBRAIO

**Numero di votanti alle ore 15,00
 di lunedì 13 febbraio
 suddivisi in maschi e femmine:**

MASCHI	FEMMINE	TOTALE
		

Il Presidente _____

TIMBRO _____

COMUNICAZIONI _____

(a) Tooltip liste

(b) Tooltip affluenze

Figura 6.13: Esempio di tooltip

ELEZIONI REGIONALI DEL 12 E 13 FEBBRAIO 2023 / COMUNICAZIONE N. 7

SEZIONE N.

COMUNICAZIONE DA RENDERE ALL'UFFICIO ELETTORALE
ENTRO LE ORE 15,15 DI LUNEDÌ 13 FEBBRAIO

Numero di votanti alle ore 15,00
di lunedì 13 febbraio
suddivisi in maschi e femmine:

MASCHI	FEMMINE	TOTALE
<input type="text" value="maschi"/>	<input type="text"/>	<input type="text"/>

Il Presidente

TIMBRO

COMUNICAZIONI _____

ELEZIONI REGIONALI DEL 12 E 13 FEBBRAIO 2023 / COMUNICAZIONE N. 7

SEZIONE N.

COMUNICAZIONE DA RENDERE ALL'UFFICIO ELETTORALE
ENTRO LE ORE 15,15 DI LUNEDÌ 13 FEBBRAIO

Numero di votanti alle ore 15,00
di lunedì 13 febbraio
suddivisi in maschi e femmine:

MASCHI	FEMMINE	TOTALE
<input type="text" value=""/>	<input type="text"/>	<input type="text"/>

Il Presidente

TIMBRO

COMUNICAZIONI _____

ELEZIONI REGIONALI DEL 12 E 13 FEBBRAIO 2023 / COMUNICAZIONE N. 7

SEZIONE N.

COMUNICAZIONE DA RENDERE ALL'UFFICIO ELETTORALE
ENTRO LE ORE 15,15 DI LUNEDÌ 13 FEBBRAIO

Numero di votanti alle ore 15,00
di lunedì 13 febbraio
suddivisi in maschi e femmine:

MASCHI	FEMMINE	TOTALE
<input type="text"/>	<input type="text"/>	<input type="text"/>

Il Presidente

TIMBRO

COMUNICAZIONI _____

ELEZIONI REGIONALI DEL 12 E 13 FEBBRAIO 2023 / COMUNICAZIONE N. 7

SEZIONE N.

COMUNICAZIONE DA RENDERE ALL'UFFICIO ELETTORALE
ENTRO LE ORE 15,15 DI LUNEDÌ 13 FEBBRAIO

Numero di votanti alle ore 15,00
di lunedì 13 febbraio
suddivisi in maschi e femmine:

MASCHI	FEMMINE	TOTALE
<input type="text"/>	<input type="text"/>	<input type="text"/>

Il Presidente

TIMBRO

COMUNICAZIONI _____

Figura 6.14: I puntatori nell'applicativo

Capitolo 7

Prototipo OCR

Come per il prototipo del training, anche quello dell'OCR deve seguire dei pattern ben definiti perché deve essere integrato in un applicativo già esistente. Ovviamente il lavoro relativo all'UI/UX è stato minore rispetto a quello fatto per il training, perché in questa fase quello che è fondamentale è l'algoritmo di riconoscimento e di corretta collocazione dei valori letti all'interno delle maschere di inserimento dati.

Dato che l'applicativo è realizzato per una web application, tutte le figure mostrate in questo capitolo saranno relative alla visualizzazione tramite tablet o smartphone. Per lo sviluppo dell'applicativo sono stati forniti foglietti relativi a un comune, mentre all'interno della web app è attiva una elezione relativa a un'altra città, per questo motivo non vi sarà corrispondenza tra i nominativi usati nei foglietti d'esempio e quelli all'interno dell'applicativo. Nonostante ciò quello che conterà saranno i risultati scritti nell'ordine e nel campo di input corretto, dunque questa discrepanza tra nominativi e numero di candidati non andrà a influire sulla correttezza dei risultati ottenuti. Visto che i candidati sindaco dell'elezione attiva erano 8 e non 6 come sul nostro fonogramma, gli ultimi due candidati manterranno i voti precedentemente inseriti dall'operatore e non riscontrati tramite OCR.

7.1 Pulsante attivazione OCR

Come specificato nel capitolo 5 l'OCR verrà fatto foglietto per foglietto, dunque abbiamo inserito un file-picker all'interno di ogni singola pagina riferita a un sub-evento elettorale. Si è scelto di utilizzare un'icona al posto del classico selettore di file (come quello usato nel training) perché più intuitivo per l'utente. La pagina si presenta in maniera semplice, innanzitutto all'utente è richiesto di inserire la sezione, una volta scelta, per quanto riguarda le affluenze verrà chiesto di selezionare l'orario dell'affluenza, per quanto riguarda le preferenze la lista a cui si fa riferimento, per gli altri sub-eventi invece verrà direttamente mostrata la maschera di inserimento

dati. L'icona OCR viene visualizzata sempre di fianco alla scelta della sezione, così da averla visibile quando si fa partire la scansione, essendo un dato molto importante.

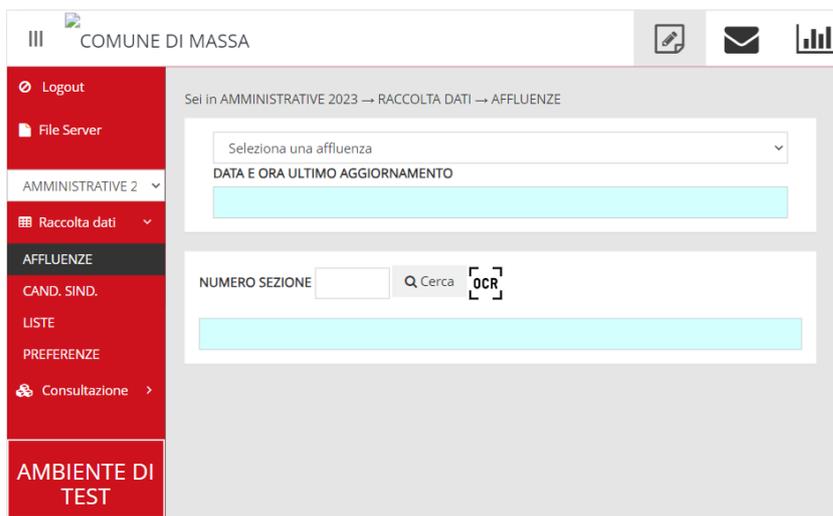


Figura 7.1: Esempio pagina sub-evento web application

In caso venga cliccata l'icona OCR prima che avvenga la scelta relativa alla sezione elettorale viene mostrato un messaggio di errore. Anche in questo caso, come per il training, è utilizzato un alert per mostrare gli errori. Questo però non è quello di default ma è stato realizzato ad hoc per l'applicativo. La scelta di

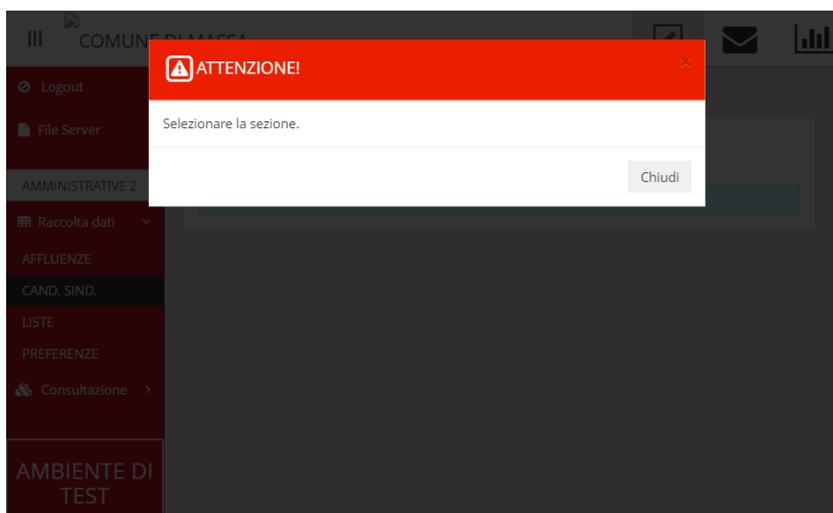


Figura 7.2: Alert sezione non inserita

avere un alert customizzato per la web app e non per eleweb non dipende da noi in quanto sono stati utilizzati in entrambi casi gli alert di default dell'applicativo.

Una volta premuto sull'icona OCR, viene richiesto all'utente se vuole scattare una foto o se selezionarla dalla galleria. La maniera in cui questo viene mostrato è indipendente dalla nostra implementazione e dipende dal modello e dal sistema operativo del device utilizzato.

7.2 Lo scanning dell'immagine

Una volta selezionato il file bisogna mostrare a video in qualche maniera la foto scannerizzata in modo da ottenere semplicemente il foglio e non tutta la foto. Questa operazione verrà chiamata ritaglio. Il ritaglio andrà mostrato in quanto l'operatore che effettua la fotografia deve capire se il è corretto o meno. In un primo momento quello che si voleva fare era una visione live del taglio, ovvero accedere direttamente alla fotocamera, e ogni 10 ms disegnare le linee del ritaglio che verrebbe eseguito se in quell'istante si scattasse la foto. Il problema di questa soluzione è che non essendo un'applicazione nativa non potevamo direttamente agire sulla fotocamera, ma avremmo dovuto simularla attraverso l'utilizzo di un canvas e di un bottone per scattare la foto. Tutto questo perché non è possibile disegnare le linee necessarie direttamente all'interno dell'applicazione fotocamera del telefono senza l'uso di un app nativa.

Qualora avessimo deciso che questo non era un problema e potevamo simulare una fotocamera, accedere direttamente allo stream video dell'app camera dell'iphone è molto complesso, soprattutto per versioni ormai superate di iOS e dunque si sarebbe perso troppo tempo per risolvere questa problematica. Nel caso in cui il cliente al quale il prodotto sarà venduto abbia la necessità di vedere il live stream del taglio verrà implementata questa funzionalità.

Si è optato per mostrare il risultato del ritaglio dopo che la foto è stata scattata. La prima soluzione ideata era quella di aprire un popup e mostrare la fotografia al suo interno però questa soluzione sarebbe stata adatta per un computer, ma per un device mobile non sembrava la soluzione migliore aprire un'altra pagina, dunque si è scelto di simulare un popup tramite l'utilizzo di un overlay.

Un elemento è detto overlay se è usato come elemento visuale sovrapposto ad un altro (nel nostro caso è la pagina questo elemento sottostante). Solitamente è usato per aggiungere un informazione o la modifica dell'aspetto dell'elemento sottostante. Noi lo usiamo per mostrare la foto tagliata e chiedere all'utente se va bene il ritaglio e allora lanciare l'algoritmo di riconoscimento OCR o se il risultato ottenuto non è abbastanza accurato e dunque annullare l'operazione e rieseguire la foto.

Nell'overlay sono presenti due bottoni, uno di conferma e uno di annullamento. Si è optato per l'utilizzo del colore rosso per annullare l'operazione e il colore blu per confermare, si è scelto questo colore per riprendere il pattern dell'applicativo, ma in caso dai test risulti ingannevole e non abbastanza intuitivo si può pensare di cambiare e utilizzare il verde come di consuetudine per i bottoni di conferma.

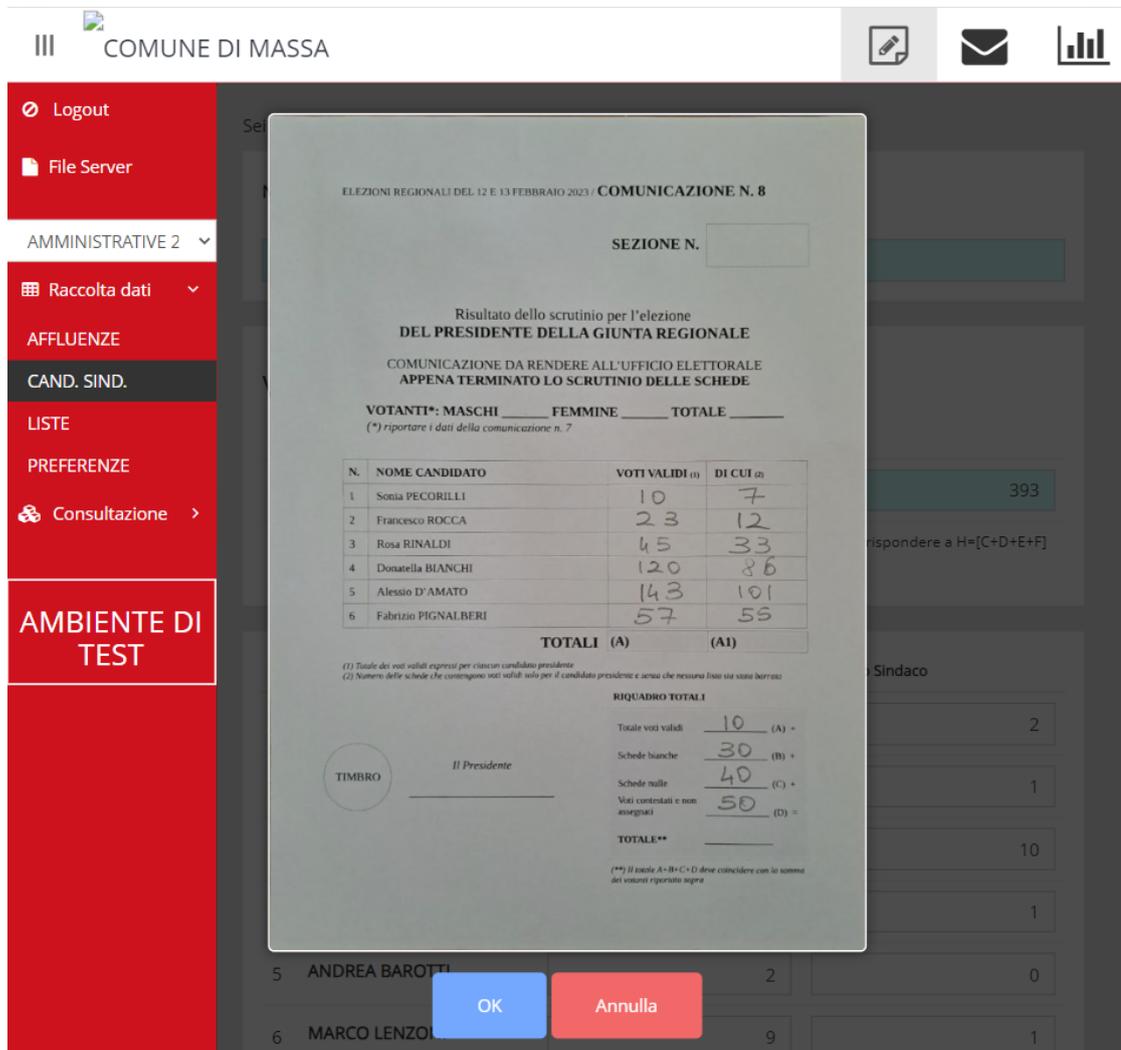


Figura 7.3: Esempio di overlay per il sub evento raggruppamenti

Dopo che si preme il pulsante di conferma, viene lanciato l'algoritmo che effettua l'OCR, confronta i box con i rettangoli del DB e scrive sulla maschera d'inserimento dati. Questa operazione è molto onerosa in termini di time consuming, è dunque necessario che durante questo tempo l'utente non possa eseguire altre operazioni, ma sia a conoscenza che qualcosa in background sta avvenendo e che deve aspettare il suo completamento. Per fare ciò si è utilizzato un elemento già presente all'interno dell'applicativo, che viene usato in genere quando si aspetta una risposta dal server che ci mette più tempo del previsto ad arrivare.

E' un alert di attesa, il problema di questo elemento è che non è ben chiaro quanto manca al termine dell'operazione in background. Si è comunque scelto di utilizzarlo per coerenza con il resto dell'applicativo. Vi è una scritta e una barra a strisce grige e nere, quella è semplicemente una gif che viene mostrata, non rappresenta l'effettivo completamento dell'esecuzione del task.

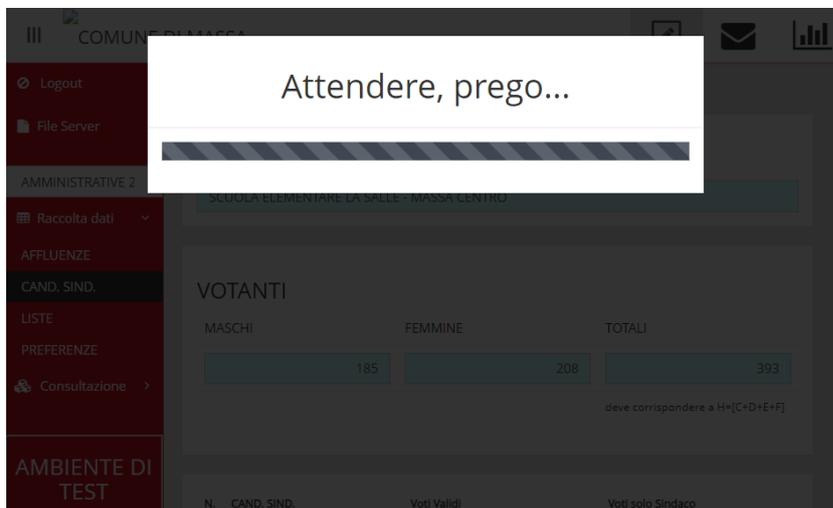


Figura 7.4: Alert di attesa

7.3 L'inserimento automatico dei dati

Come ampiamente riportato nei capitoli precedenti l'OCR non garantisce una capacità di riconoscimento del 100%, dunque gli operatori dovranno controllare tutti i valori che vengono inseriti in modo automatico nella maschera d'inserimento. Per velocizzare questa operazione, sulla base di alcune soglie decise con il singolo comune viene definito un codice colore per disegnare i bordi delle celle di inserimento dati, in modo che sulla base del colore l'operatore sappia quali dati controllare maggiormente. Per questo lavoro di tesi si sono scelte 3 soglie: per i valori con una confidence maggiore dell'80% il risultato è considerato affidabile, il bordo di queste

celle non viene colorato ma lasciato quello di default, questo perché anche una confidence del 99% non garantisce che il dato letto sia effettivamente quello scritto, dunque non si vuole ingannare l'utente utilizzando un bordo verde, perché potrebbe essere indotto a pensare che non è necessario verificare che il valore riportato sia quello corretto. Tra il 60% e l'80% il valore riportato potrebbe essere giusto ma vi è una buona probabilità che il valore non sia stato riconosciuto correttamente, la cella viene colorata di giallo, a indicare la probabile presenza di un errore. Sotto il 60% il colore scelto è il rosso, la probabilità di errore è elevata e dunque è necessario che l'utente sappia che è necessaria la verifica del dato.

N.	CAND. SIND.	Voti Validi	Voti solo Sindaco
1	MARCO GUIDI	10	7
2	CESARE MARIA RAGAGLINI	23	12
3	FRANCESCO PERSIANI	45	33
4	ROMOLO ENZO RICCI	120	86
5	ANDREA BAROTTI	143	101
6	MARCO LENZONI	57	55
7	DANIELA BENNATI	35	3
8	GUIDO MUSSI	10	0
TOTALI :		443	297
		[C]	[B]

Schede Bianche [D]	30
Schede nulle [E]	40
Schede Contestate [F]	0
In totale	70

Figura 7.5: Esempio inserimento dati via OCR

Capitolo 8

Valutazione

Per concludere questa tesi si è deciso di eseguire un test di valutazione ed usabilità del prototipo realizzato. Un test di usabilità è un metodo di valutazione di prodotti, tra cui i siti web e le applicazioni software, che ha come obiettivo quello di valutare quanto sia semplice, facile e intuitivo da utilizzare il prodotto a cui il test fa riferimento. Per svolgere questo test è necessaria la presenza di alcuni utenti che siano parte o che abbiano ampie informazioni del dominio del prototipo sviluppato, degli osservatori e dei moderatori che devono monitorare e fornire informazioni all'utente durante il test.

8.1 Pianificazione

Per effettuare il test è stato definito un protocollo o script.

Considerato che il prodotto verrà utilizzato dagli amministratori comunali, si è eseguito il test su questa categoria di utenti. Si è scelto di sottoporre cinque persone al test, il numero di candidati a cui è stato sottoposto il test non è scelto casualmente ma numerosi studi dimostrano che bastano cinque persone per trovare la maggior parte dei problemi all'interno di un prototipo.

I test sono stati svolti su Zoom, questo perché gli amministratori scelti provenivano da diversi comuni italiani.

Durante lo svolgimento del test i partecipanti sono stati affiancati da un facilitatore incaricato di descrivere i task e annotare le loro azioni e i loro ragionamenti (è stato chiesto agli utenti di ragionare ad alta voce). In caso il partecipante si fosse trovato in difficoltà nell'eseguire un task, il facilitatore sarebbe potuto intervenire per aiutarlo.

Lo script del test prevede 7 attività che vengono descritte successivamente e riassunte nella tabella 8.1, a ogni attività è associata una metrica, in modo da poter calcolare in maniera matematica il successo o il fallimento del test.

Al termine del test, è stato sottoposto a ognuno dei partecipanti un questionario SUS(System Usability Scale) [15], infine sono state post alcune domande per trovare i problemi, le funzionalità da aggiungere e le criticità del sistema.

8.2 Attività

Per valutare il prototipo si è utilizzato un approccio basato su diverse attività. Le sette attività previste sono divise in cinque per quanto riguarda il training e due per la scansione OCR all'interno della web application.

Attività 1

Per la prima attività viene richiesto all'utente di iniziare il training per un sub-evento che ancora deve essere trainato. Si predispose dunque l'ambiente in modo che ci siano alcuni sub-eventi trainati e altri no. In questa maniera si valuta se il codice colori e la legenda vengono visti e compresi dall'utente, inoltre se comprende come caricare la foto e capisce che deve cliccare il pulsante "Inizia training" per cominciare. Questa attività viene considerata completata con successo anche in caso l'utente prema il pulsante "Inizia training" prima di aver caricato il fonogramma, ma solo nel caso in cui dopo l>alert di errore riesca a caricare correttamente il foglietto e iniziare il training.

Attività 2

In questa attività viene richiesto all'utente di seguire le istruzioni ed effettuare il training. Inoltre il partecipante è invitato a "giocare" e provare a ingrandire, spostare, eliminare un rettangolo. E' importante tenere traccia innanzitutto del fatto che l'utente comprenda come disegnare i rettangoli, e poi delle azioni che riesce a fare, con un focus sulla cancellazione, che in fase di progettazione è l'operazione che è ritenuta più criptica da comprendere per gli utenti.

Attività 3

Questa attività è molto semplice, bisogna solo valutare se l'utente comprende come è possibile salvare il training una volta finito. Dopo aver terminato il training il partecipante potrebbe voler ancora modificare i rettangoli, ma non gli viene richiesto per forza di farlo, dunque non verrà valutato questo task.

Attività 4

Questa attività è simile alla prima, viene richiesto però di scegliere un sub-evento già trainato e di caricare il training esistente, dunque ovviamente serve a capire se

il partecipante si accorge che l'interfaccia cambia rispetto a un sub-evento privo di training e capisce che il pulsante "carica training" fa esattamente quello. Inoltre viene richiesto anche in questo caso di modificare, spostare, eliminare i rettangoli caricati.

Attività 5

E' l'ultima attività per quanto riguarda il training. Viene richiesto al partecipante di eliminare le modifiche al training. Una volta terminata l'attività la valutazione relativa al training è finita e si passerà alla valutazione della web app.

Attività 6

Prima attività all'interno della web application, viene richiesto al partecipante di caricare una foto e avviare l'OCR per un sub-evento. Bisogna verificare che comprenda che deve inserire la sezione prima di avviare l'OCR. Si considera superata l'attività anche se l'utente prova ad avviare l'OCR senza selezionare la sezione, nel caso in cui dopo l'alert di errore riesca a eseguire correttamente l'operazione.

Attività 7

Per questa ultima attività viene richiesto di verificare che i dati inseriti siano corretti. E' importante che il partecipante capisca che l'OCR non ha un'affidabilità del 100% e che dunque deve controllare i valori inseriti nella maschera, inoltre dovrebbe capire che il codice colori è usato per identificare i dati con diverse probabilità di errore.

8.3 Svolgimento e Risultati

Gli strumenti utilizzati per lo svolgimento del test sono stati: un computer sul quale era presente il prototipo funzionante, un computer in remoto, del partecipante, e infine il mezzo utilizzato è stato Zoom. Il test non è stato registrato ma i comportamenti e le riflessioni ad alta voce dei partecipanti sono state annotate tramite carta e penna. I partecipanti sono stati selezionati da 5 differenti comuni, per motivi di privacy non verranno rivelati i nomi e i comuni di appartenenza, dunque per distinguerli ci si riferirà a loro con identificativi da P01 a P05.

Tutti i partecipanti hanno una buona esperienza del web e sono a conoscenza della problematica che il prototipo mira a risolvere.

Le attività scelte per questo test sono mirate a valutare tutte le funzionalità dei due moduli, quello di training e dello di scansione OCR, è stato chiesto ai partecipanti di ragionare ad alta voce in modo da poter conoscere e annotare il perché delle decisioni degli utenti.

Risultati

I test hanno avuto esito nel complesso un esito positivo. I partecipanti P01 e P02 hanno riscontrato un problema iniziale nell'eseguire l'attività 6 ovvero avviare la procedura OCR, hanno avuto bisogno di più tempo del normale per cliccare sul bottone OCR e far partire la procedura, questo perché la forma del cursore, essendo in realtà quell'icona un file picker e non un bottone, non cambia e dunque non pensavano di poterci interagire. Nessuno dei partecipanti è riuscito al primo tentativo a eliminare un rettangolo, ad eccezione di P05. Il partecipante P03 nell'attività 4, dopo non esserci riuscito nella 2, è riuscito a provare ad eliminarne uno, per poi accorgersi che non era più consentito in quella fase del training, a seguito della comparsa dell>alert. I partecipanti P03 e P04 hanno selezionato l'immagine per iniziare il training solamente dopo la comparsa dell>alert, perché hanno subito cliccato sul bottone inizia training. P04 e P05 hanno premuto l'icona OCR prima della selezione della sessione. In generale i partecipanti hanno capito che il colore giallo e rosso indicavano una probabilità di errore nel riconoscimento OCR, il partecipante P02 però ha pensato fosse dovuto a valori non accettabili perché troppo grandi o troppo piccoli, dunque non ha compreso il vero significato.

8.3.1 Possibili modifiche

Sulla base dei risultati ottenuti dai test e dai commenti dei partecipanti, emerge che l'applicativo nel complesso è abbastanza facile e intuitivo da usare, detto ciò comunque delle modifiche potrebbero essere fatte.

Due partecipanti hanno selezionato comincia training prima di selezionare l'immagine, il che non è un errore ma comunque fa perdere tempo agli operatori, si potrebbe considerare dunque di mostrare i bottoni per iniziare e caricare il training, solo a seguito della selezione dell'immagine. Idem per ciò che riguarda il training qualcuno dei partecipanti ha selezionato il pulsante OCR prima di inserire la sezione, si potrebbe anche qui far comparire il bottone solo una volta dopo aver inserito la sezione.

Un cambiamento che sicuramente va apportato è trasformare il puntatore quando ci si posiziona sopra l'icona OCR, utilizzando lo stile "pointer" ovvero il classico puntatore a forma di mano utilizzato per quando ci si trova sopra un bottone.

Va modificato anche il modo in cui si eliminano i rettangoli. Solamente uno dei cinque partecipanti ha trovato subito il modo di eliminare un rettangolo ovvero il P05. Si è notato che molti partecipanti hanno provato a usare il tasto destro del mouse per farlo. Esiste un evento che si può aggiungere al canvas chiamato *contextmenu*, che sarebbe l'evento relativo a ciò che accade quando viene premuto il tasto destro del mouse. Se si disattiva il metodo di default e si riscrive la funzione relativa all'evento è possibile chiedere all'utente se è sicuro di voler eliminare il rettangolo se ci clicca all'interno.

Titolo	Descrizione	Criteri di successo	Metrica
1 Iniziare un training	Al partecipante viene presentata la pagina Training OCR e cerca di capire che colore hanno i sub-eventi non trainati ed inizia un training per uno di loro.	Il partecipante riesce a capire il colore, seleziona il sub-evento, carica un fonogramma tramite il file picker e preme il bottone "Inizia training".	Successo (1) o fallimento (0)
2 Eseguire un training	Il partecipante segue le istruzioni e disegna tutti i rettangoli necessari nella posizione giusta, gli è richiesto anche di provare a ridimensionarli, spostarli ed eliminarli.	Il partecipante riesce a comprendere come sono mostrate le istruzioni, disegna i rettangoli e li modifica.	1 punto per il successo più da 0 a 3 punti, uno per ogni operazione di modifica
3 Salvare il training	Finita l'attività 2 il partecipante deve salvare il training.	Il partecipante riesce correttamente a salvare un training e verificare che effettivamente il colore del bottone è cambiato, dunque il salvataggio è avvenuto.	Successo (1) o fallimento (0)
4 Caricare un training	Il partecipante deve caricare un training selezionando un sub-evento già trainato e notando che è presente un ulteriore bottone.	Il partecipante riesce correttamente a caricare il training seguendo passi simili a quelli dell'attività 1. Provando a eliminare un rettangolo dovrebbe accorgersi che non si può fare.	1 punto per il successo più da 0 a 3 punti, uno per ogni operazione di modifica
5 Annullare il training	Il partecipante deve annullare le modifiche effettuate.	Il partecipante riesce correttamente ad annullare le modifiche tramite il bottone "Annulla training".	Successo (1) o fallimento (0)
6 Iniziare una scansione OCR	Il partecipante deve selezionare un sub-evento e iniziare la scansione di un foglio fotografato. 64	Il partecipante trova il bottone per iniziare l'iter di riconoscimento OCR, carica il foglio, conferma il ritaglio e fa partire la procedura di scansione.	Successo (1) o fallimento (0)
7 Verifica inserimento dati	Il partecipante deve controllare che i dati inseriti siano corretti.	Il partecipante controlla i dati riconoscendo che il codice colori è usato a seconda dell'affidabilità di un dato.	Successo (1) o fallimento (0)

Tabella 8.1: Attività per la valutazione

Partecipante	Attività 1	Attività 2	Attività 3	Attività 4	Attività 5	Tasso di successo medio per partecipante
P01	1	3/4	1	3/4	1	60%
P02	1	3/4	1	3/4	1	60%
P03	1	3/4	1	4/4	1	80%
P04	1	3/4	1	3/4	1	60%
P05	1	4/4	1	4/4	1	100%
Tasso di successo medio per task	100%	100%	100%	40%	100%	

Tabella 8.2: Tabella dei risultati: esiti delle attività del training

Partecipante	Attività 6	Attività 7	Tasso di successo medio per partecipante
P01	1	1	100%
P02	1	0	50%
P03	1	1	100%
P04	1	1	100%
P05	1	1	100%
Tasso di successo medio per task	100%	80%	

Tabella 8.3: Tabella dei risultati: esiti delle attività dell'OCR

Capitolo 9

Conclusioni

L'obiettivo del tirocinio e del progetto di tesi era la creazione di un'interfaccia flessibile e adattabile per integrare un modulo di riconoscimento OCR all'interno dell'applicativo Eleweb.

Si è partiti dallo studio del dominio, con l'obiettivo di comprendere al meglio il contesto e i bisogni delle amministrazioni comunali. Successivamente sulla base delle informazioni acquisite si è scomposto il problema in modo da poter sviluppare correttamente i singoli moduli al fine di renderli facili, veloci e utili da usare. Si è ottenuto un prodotto in grado di riconoscere i dati all'interno del fonogramma e riportarli nei corretti campi di input della maschera di inserimento dati grazie all'operazione di training fatta in periodo pre-elezione.

Per quanto riguarda l'interfaccia grafica sono stati eseguiti dei test di usabilità sia all'interno di Eleweb che della web app a cinque differenti operatori comunali. Ai partecipanti è stato richiesto di utilizzare la piattaforma e fornire delle valutazioni sulle problematiche e i punti di forza dell'applicativo, al fine di poter andare a intervenire laddove fossero evidenziate delle criticità da parte di uno o più partecipanti. Nel complesso le interviste hanno avuto un esito positivo, ovviamente a seconda delle esigenze specifiche del singolo comune l'applicativo verrà modificato ed eventualmente integrerà ulteriori funzionalità. Si è inoltre somministrato un questionario SUS a tutti i partecipanti ed ha avuto una valutazione media di 90, considerato un risultato positivo.

Possiamo concludere che è stato sviluppato un applicativo che va a rendere più facile e veloce uno dei punti critici all'interno di un evento elettorale, ovvero il riportare i dati dei fonogrammi sulla piattaforma Eleweb.

9.1 Sviluppi futuri

Rispetto alla terminazione del lavoro di tesi, il progetto ha già subito degli sviluppi. In seguito alla conclusione del modulo OCR, il prodotto è stato proposto al Comune di Milano per l'acquisizione. Sono state richieste delle modifiche sulla base delle loro esigenze, sono state sviluppate e il comune ha deciso di comprare il prodotto. E' stata creata da zero una nuova web application che verrà utilizzata dai membri della polizia municipale i quali scatteranno una foto. Una volta scattata verrà tagliata dallo scanner, inoltre è stato aggiunto un modulo di QRCode scanner che leggerà un QRCode inserito dal comune sui foglietti ed estrapolerà informazioni riguardanti elezione, sezione, tipo di sub-evento elettorale e infine se presente, un informazione aggiuntiva che per l'affluenza è l'orario cui si riferisce e per le preferenze la lista associata ai candidati.

Queste informazioni verranno inserite all'interno di alcuni elementi HTML e saranno modificabili dal vigile che potrà inoltre verificare la foto originale e come è stata ritagliata e confermare o annullare l'operazione. In caso di conferma la foto originale e quella tagliata verranno salvate su un server insieme alle informazioni aggiuntive, in caso di annullamento il vigile potrà riscattare la foto.

Il riconoscimento OCR non viene più fatto partire una volta scattata una foto, ma è stato creato uno script Python che viene eseguito periodicamente che processa tutte le foto presenti sul server che ancora non sono state processate e va a salvare i valori su alcune tabelle ausiliarie che sono state aggiunte sul database. Anche l'inserimento dei dati all'interno del db è stato un aggiornamento dell'applicativo perché per questo lavoro di tesi non era previsto che i dati ottenuti dall'OCR venissero salvati sul database.

La web application non viene più utilizzata, almeno da questo comune, il codice relativo al coloramento dei bordi della maschera d'inserimento è stato comunque "riciclato" per essere utilizzato direttamente all'interno dell'applicativo Eleweb che è quello utilizzato dal centro di raccolta dati del comune di Milano per l'inserimento dei dati riportati all'interno dei fonogrammi elettorali. Quando la fotografia scattata dall'operatore della polizia municipale è stata processata e i dati sono stati inseriti all'interno del database, arriva una notifica al centro raccolta dati relativa al sub-evento e alla sezione del foglio che è stato scansionato. L'operatore all'interno della schermata di inserimento dati vedrà due pulsanti. Il primo permette di mostrare in un popup la foto originale scattata, così da poter visualizzare il fonogramma e i suoi dati, il secondo caricherà i dati all'interno della maschera. In questa maniera l'operatore può utilizzare la fotografia per popolare la maschera, oppure può caricarla con il bottone di caricamento e controllare che i valori riportati siano corretti. Per quanto riguarda la fase di training invece non vi è stata alcuna modifica, il comune l'ha ritenuta intuitiva e veloce e non aveva particolari esigenze di cambiamento.

9.2 Best practices

Durante i vari test eseguiti per valutare il corretto funzionamento dell'applicativo si è notato che ci sono degli accorgimenti che andrebbero presi per sfruttare al meglio l'applicativo. Questi accorgimenti verranno comunicati ai comuni ai quali verrà venduto l'applicativo in modo da poterne sfruttare al meglio le prestazioni. Innanzitutto si è visto che per fare funzionare al meglio lo scanner, la foto al fonogramma va scattata tenendo il telefono parallelo al foglio, a circa 40/50 cm di distanza. In caso sia presente una luce che va a incidere di fianco ai lati del foglio, conviene effettuare la fotografia con il flash, per evitare che la luce, essendo di colore bianco, vada a confondere l'algoritmo di ritaglio. Avere una superficie nera al di sotto del foglio è la soluzione migliore per un corretto ritaglio, perché vi è il massimo contrasto con il bianco, si sconsigliano ovviamente superfici chiare. In caso si avessero tavoli di colori chiari è consigliabile mettere un cartoncino scuro sotto il foglio, anche di pochi centimetri più grande, in questa maniera lo scanner funzionerà correttamente.

Per quanto riguarda i fonogrammi sarebbe opportuno che le varie celle dove devono essere inseriti i dati abbiano uno spessore minimo, in modo da non scrivere le cifre con un carattere troppo piccolo che l'OCR potrebbe non riconoscere correttamente. Tramite alcuni test si è riscontrato che con celle inferiori agli 11 pt l'OCR non riesce a riconoscere in maniera affidabili le cifre, dunque si consigliano celle di dimensioni maggiori, sopra i 15 pt sarebbe la soluzione ottimale sulla base dei test effettuati.

Appendice A

Script per la valutazione del prototipo

A.1 Presentazione

Ciao! Grazie mille per aver deciso di partecipare a questo studio. Permettimi di descriverti brevemente in che contesto andrà utilizzato questo prototipo e come funziona. Il progetto che ho sviluppato propone una soluzione innovativa e alternativa per l'inserimento dati da parte delle amministrazioni comunali durante l'evento elettorale. E' stato sviluppato un modulo OCR che permette di inserire i dati all'interno dell'applicativo semplicemente scattando una foto ad un fonogramma. Per far si che il modulo funzioni correttamente, prima dell'evento elettorale occorre eseguire un'operazione di training. Per training si intende fare si che quando l'OCR va a processare il fonogramma, riesca a capire in che posizione sul foglio sono presenti i dati rilevanti che andranno inseriti nella maschera di inserimento dati presente nella web app di Eleweb. Il training dunque va fatto per ogni sub-evento elettorale e consiste nel dover caricare il fonogramma e disegnare dei rettangoli tramite il mouse negli spazi del foglio in cui saranno presenti i dati. Una volta completati i task relativi al training invece passeremo alla web application, ovvero all'effettiva scansione OCR dei fonogrammi e ti chiederò di completare nuovi task, questa seconda parte sarà più breve. All'interno della web application sarà possibile fare partire la scansione OCR per ogni sub-evento. Durante il test riceverai delle istruzioni relative ai task che dovrai andare a svolgere, ma potrai tranquillamente interagire con l'applicativo come vorrai, senza ricevere delle forzature da parte dell'intervistatore. Sarà presente anche un facilitatore in caso tu abbia delle difficoltà a svolgere il task che ti è stato richiesto di svolgere, in modo da poter chiedere un aiuto.

A.2 Istruzioni

Considerato che quello con cui andrai a interagire è solo un prototipo presente semplicemente in locale sul mio dispositivo, sfrutteremo il controllo remoto offerto da Zoom così che tu possa interagire col prototipo. Interverrò solo per inserire al momento opportuno le foto dei fonogrammi, perché sono presenti all'interno del mio PC e non ti sono state fornite.

L'applicativo è ancora un prototipo ma tutte le funzionalità sono state sviluppate e dovrebbero funzionare correttamente. Qualora dunque dovessi riscontrare degli errori o necessitare di ulteriori funzionalità ti chiedo di comunicarmelo in modo che io possa prenderne nota. Ti troverai direttamente all'interno della pagina **Training OCR** presente nell'applicativo perché la posizione della voce training OCR sul menù laterale non è dipendente dallo sviluppo del sistema ma una scelta dell'azienda, dunque non è parte del lavoro svolto. E' stato scelto come evento elettorale le amministrative del 2018, che prevedono la presenza di 4 sub-eventi elettorali in modo da poter interagire con tutti.

Finita questa parte, invece aprirò la web application in modalità visualizzazione smartphone così da poter simulare di essere su un device mobile, ti troverai già loggato all'interno dell'applicativo e sarà utilizzato come esempio l'evento elettorale precedente ovvero amministrative 2018, questa volta però nella main page della web app.

Una volta terminato il test ti chiederò di rispondere a un questionario chiamato SUS (System Usability Scale) che è uno strumento molto utilizzato per valutare l'usabilità di un sistema che consiste in dieci domande a risposta chiusa a cui ti chiedo di rispondere con la massima sincerità.

Condividere la pagina e attivare il controllo remoto, infine verificare che in effetti il partecipante riesca a vedere e interagire con la pagina

A.3 Attività

A.3.1 Attività 1: Iniziare un training

Per questa prima attività, vorrei che interagissi con i bottoni che vedi e iniziassi un training per il primo sub-evento che ancora non è stato trainato. Pensa a voce alta, commenta i passi che effettuerai.

Non hai limite di tempo per terminare questa attività.

Al termine, passare alla prossima attività.

A.3.2 Attività 2: Eseguire un training

Per la seconda attività devi seguire le istruzioni che ti verranno mostrate e provare a eseguire il training. Ti chiedo anche di provare a spostare, ridimensionare e eliminare i rettangoli che andrai a disegnare.

Nuovamente non hai limite di tempo per l'esecuzione di questa task.

Vedere se il partecipante ha qualche problema nel capire cosa deve fare, si trova in una situazione di stallo o difficoltà. Al termine, passare alla prossima attività.

A.3.3 Attività 3: Salvare il training

Una volta terminata l'attività due ti chiedo di provare a salvare il training e vedere successivamente se è stato salvato correttamente

Segui tutti i passi che ti sembrano necessari. Vorrei che spiegassi ad alta voce cosa stai facendo e cosa stai pensando.

Come prima, non hai limiti di tempo per concludere questa attività.

Al termine, passare alla prossima attività.

A.3.4 Attività 4: Caricare un training

Per questa attività, vorrei che come per l'attività 1 selezionassi un sub-evento ma questa volta già trainato, può essere il sub-evento che hai trainato precedentemente o uno già trainato prima del nostro test. Prova anche in questo caso a interagire con i rettangoli che trovi. Scegli il percorso che preferisci, commenta a voce alta quello che stai facendo e cosa stai pensando.

Al termine, passare alla prossima attività.

A.3.5 Attività 5: Annullare il training

Una volta caricato il training e interagito con i rettangoli, ti chiedo di annullare tutte le operazioni che hai eseguito.

Al termine, passare alla prossima attività.

Ora apri la web application e concedi nuovamente il comando all'utente. Verifica che tutto funzioni correttamente

A.3.6 Attività 6: Iniziare una scansione OCR

Per questa prima attività sulla web applicaton, vorrei che facessi partire l'OCR per un sub-evento a tua scelta, una volta selezionato il sub-evento ti verrà fornito un fonogramma precompilato relativo ad esso Pensa a voce alta, commenta i passi che effettuerai.

Non hai limite di tempo per terminare questa attività.

Attendere che l'utente trovi il modo per far partire il processo OCR, in caso non riesca forniscigli aiuto. Una volta trovato il modo fornisci il fonogramma corretto. Al termine, attendere la scansione sia completa e passare alla prossima attività.

A.3.7 Attività 7: Verifica inserimento dati

Per la seconda attività verifica che i dati siano stati inseriti in maniera corretta, descrivi cosa noti a voce alta.

Nuovamente non hai limite di tempo per l'esecuzione di questa task.

Al termine, passare alla prossima attività.

A.4 Debriefing

Grazie per aver eseguito tutti i task, non vi è altro da eseguire per questo test, ti chiedo soltanto la cortesia di compilare questo questionario SUS e infine se hai voglia di rispondere a qualche domanda. *Passa il questionario SUS al partecipante e al termine se ha risposto in modo affermativo al ricevere ulteriori domande, effettuale*

1. Ti è piaciuto questo test?
2. Reputi che andrebbero fatti dei cambiamenti alle attuali funzionalità del sistema?
3. Ci sono delle funzionalità che secondo te dovrebbero essere aggiunte?
4. Trovi corretta la maniera in cui puoi inserire i fonogrammi per far partire la scansione OCR o preferiresti una pagina a parte, separata dalla pagina del sub-evento elettorale?
5. Hai domande sul questionario?

Grazie mille per la tua partecipazione, i tuoi commenti e il tuo test saranno di grande aiuto per lo sviluppo del prodotto finale.

Appendice B

Questionari SUS relativi al training

In questa appendice sono riportati i questionari SUS compilati dai cinque partecipanti al test di usabilità relativo al training. Per ognuno di essi è stato calcolato il relativo punteggio finale.

Domande		Voto del partecipante	Punteggio risultante
1.	Penso che mi piacerebbe utilizzare questo sito frequentemente	5	4
2.	Ho trovato il sito inutilmente complesso	1	4
3.	Ho trovato il sito molto semplice da usare	5	4
4.	Penso che avrei bisogno del supporto di una persona già in grado di utilizzare il sito	1	4
5.	Ho trovato le varie funzionalità del sito bene integrate	4	3
6.	Ho trovato incoerenze tra le varie funzionalità del sito	1	4
7.	Penso che la maggior parte delle persone possano imparare ad utilizzare il sito facilmente	4	3
8.	Ho trovato il sito molto difficile da utilizzare	1	4
9.	Mi sono sentito a mio agio nell'utilizzare il sito	5	4
10.	Ho avuto bisogno di imparare molti processi prima di riuscire ad utilizzare al meglio il sito	1	4
Valutazione totale			95

Tabella B.1: Valutazione SUS per partecipante P01

Domande		Voto del partecipante	Punteggio risultante
1.	Penso che mi piacerebbe utilizzare questo sito frequentemente	4	3
2.	Ho trovato il sito inutilmente complesso	1	4
3.	Ho trovato il sito molto semplice da usare	4	3
4.	Penso che avrei bisogno del supporto di una persona già in grado di utilizzare il sito	2	3
5.	Ho trovato le varie funzionalità del sito bene integrate	5	4
6.	Ho trovato incoerenze tra le varie funzionalità del sito	2	3
7.	Penso che la maggior parte delle persone possano imparare ad utilizzare il sito facilmente	5	4
8.	Ho trovato il sito molto difficile da utilizzare	1	4
9.	Mi sono sentito a mio agio nell'utilizzare il sito	5	4
10.	Ho avuto bisogno di imparare molti processi prima di riuscire ad utilizzare al meglio il sito	1	4
Valutazione totale			90

Tabella B.2: Valutazione SUS per partecipante P02

Domande		Voto del partecipante	Punteggio risultante
1.	Penso che mi piacerebbe utilizzare questo sito frequentemente	4	3
2.	Ho trovato il sito inutilmente complesso	2	3
3.	Ho trovato il sito molto semplice da usare	4	3
4.	Penso che avrei bisogno del supporto di una persona già in grado di utilizzare il sito	1	4
5.	Ho trovato le varie funzionalità del sito bene integrate	5	4
6.	Ho trovato incoerenze tra le varie funzionalità del sito	1	4
7.	Penso che la maggior parte delle persone possano imparare ad utilizzare il sito facilmente	5	4
8.	Ho trovato il sito molto difficile da utilizzare	1	4
9.	Mi sono sentito a mio agio nell'utilizzare il sito	4	3
10.	Ho avuto bisogno di imparare molti processi prima di riuscire ad utilizzare al meglio il sito	1	4
Valutazione totale			90

Tabella B.3: Valutazione SUS per partecipante P03

Domande		Voto del partecipante	Punteggio risultante
1.	Penso che mi piacerebbe utilizzare questo sito frequentemente	3	2
2.	Ho trovato il sito inutilmente complesso	1	4
3.	Ho trovato il sito molto semplice da usare	3	2
4.	Penso che avrei bisogno del supporto di una persona già in grado di utilizzare il sito	2	3
5.	Ho trovato le varie funzionalità del sito bene integrate	5	4
6.	Ho trovato incoerenze tra le varie funzionalità del sito	1	4
7.	Penso che la maggior parte delle persone possano imparare ad utilizzare il sito facilmente	3	2
8.	Ho trovato il sito molto difficile da utilizzare	1	4
9.	Mi sono sentito a mio agio nell'utilizzare il sito	4	3
10.	Ho avuto bisogno di imparare molti processi prima di riuscire ad utilizzare al meglio il sito	1	4
Valutazione totale			80

Tabella B.4: Valutazione SUS per partecipante P04

Domande		Voto del partecipante	Punteggio risultante
1.	Penso che mi piacerebbe utilizzare questo sito frequentemente	4	3
2.	Ho trovato il sito inutilmente complesso	1	4
3.	Ho trovato il sito molto semplice da usare	5	4
4.	Penso che avrei bisogno del supporto di una persona già in grado di utilizzare il sito	1	4
5.	Ho trovato le varie funzionalità del sito bene integrate	5	4
6.	Ho trovato incoerenze tra le varie funzionalità del sito	1	4
7.	Penso che la maggior parte delle persone possano imparare ad utilizzare il sito facilmente	4	3
8.	Ho trovato il sito molto difficile da utilizzare	1	4
9.	Mi sono sentito a mio agio nell'utilizzare il sito	5	4
10.	Ho avuto bisogno di imparare molti processi prima di riuscire ad utilizzare al meglio il sito	1	4
Valutazione totale			95

Tabella B.5: Valutazione SUS per partecipante P05

Bibliografia

- [1] *Definizione del termine e-voting secondo il polyas*. URL: <https://www.polyas.it/glossario-elettorale/E-voting> (cit. a p. 1).
- [2] *Sito eleweb*. URL: <https://www.pro-logic.it/eleweb-elezioni/> (cit. a p. 1).
- [3] *Informazioni sulle elezioni europee*. URL: <https://www.europarl.europa.eu/at-your-service/it/be-heard/elections> (cit. a p. 5).
- [4] *Norme generali votazioni europee*. URL: <https://www.europarl.europa.eu/factsheets/it/sheet/21/the-european-parliament-electoral-procedures> (cit. a p. 6).
- [5] *Legge 18 del 1979*. URL: <https://www.parlamento.it/parlam/leggi/790181.html> (cit. a p. 6).
- [6] *JSON*. URL: <https://www.oracle.com/it/database/what-is-json/> (cit. a p. 16).
- [7] *Tesseract js*. URL: <https://tesseract.projectnaptha.com/> (cit. a p. 17).
- [8] *Cos'è il modello ACID*. URL: <https://www.databricks.com/it/glossary/acid-transactions> (cit. a p. 17).
- [9] *Metodo strokeRect()*. URL: https://www.w3schools.com/jsref/canvas_strokerect.asp (cit. a p. 20).
- [10] ColonelParrot. *ColonelParrot: Jscanify*. <https://github.com/ColonelParrot/jscanify> (cit. a p. 30).
- [11] Naptha. *naptha: TesseractJs*. <https://github.com/naptha/tesseract.js>. 2019 (cit. a p. 31).
- [12] Mindee. *docTR: Document Text Recognition*. <https://github.com/mindee/doctr>. 2021 (cit. a p. 31).
- [13] *DbNet*. URL: <https://arxiv.org/pdf/1911.08947.pdf> (cit. a p. 32).
- [14] *CRNN*. URL: <https://arxiv.org/pdf/1507.05717.pdf> (cit. a p. 32).

BIBLIOGRAFIA

- [15] *Questionario SUS*. URL: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> (cit. a p. 61).