POLITECNICO DI TORINO

Corso di Laurea
in Ingegneria Informatica

Tesi di Laurea

Trasformare l'Inventario delle Cabine Elettriche con la Computer Vision nel Settore Energetico, grazie a tecnologie di Robotica e Computer Vision

Relatore
prof. Luciano Lavagno
firma del relatore

. . . . . .

Candidato
Gabriele Lucca
firma del candidato

. . . . . .

Anno Accademico 2023-2024

Ai miei genitori

Sommario

Questa tesi si concentra sull'applicazione si tecnologie di Robotica e Computer Vision per ottimizzare la gestione delle cabine elettriche nel settore energetico. Il problema principale affrontato riguarda l'ottimizzazione del processo di inventario e manutenzione delle infrastrutture elettriche. Tradizionalmente, questo compito richiede che gli operai leggano manualmente le targhette sulle apparecchiature per raccogliere informazioni, un'attività ripetitiva e soggetta a errori. La tesi propone di sostituire questo metodo con un sistema che, attraverso una semplice foto, permette di acquisire e interpretare automaticamente i dati, riducendo così il rischio di errori e migliorando l'efficienza operativa. Utilizzando strumenti avanzati offerti da Amazon Web Services (AWS), come Textract per il riconoscimento ottico dei caratteri (OCR) e Bedrock per l'implementazione di modelli generativi, il lavoro propone un sistema automatizzato che integra la visione artificiale con processi robotici per migliorare la raccolta e l'interpretazione dei dati tecnici.

Il progetto si articola in diverse fasi, iniziando con l'analisi e la valutazione di modelli OCR per l'estrazione dei dati da targhette elettriche, seguita dallo sviluppo di modelli personalizzati tramite AWS Rekognition e Textract. Successivamente, è stato sviluppato un modello generativo che sfrutta embedding per interpretare e contestualizzare le informazioni raccolte. Attraverso una serie di esperimenti e confronti tra diverse soluzioni tecniche, la tesi evidenzia come l'integrazione tra modelli OCR e generativi non solo aumenti la precisione dell'inventario, ma riduca in modo significativo i rischi operativi e i tempi di intervento nelle attività di manutenzione. I risultati ottenuti indicano che queste tecnologie, se opportunamente adattate, potrebbero essere applicate con successo anche in altri settori industriali per affrontare problemi analoghi.

Ringraziamenti

Arrivare a questo traguardo è stato un viaggio impegnativo e pieno di sfide, e non avrei potuto farcela senza il supporto di persone a me molto care.

Prima di tutto, voglio esprimere la mia più profonda gratitudine ai miei genitori e mio fratello, che mi hanno sempre sostenuto con amore incondizionato e che mi hanno supportato (e sopportato) anche nei momenti più difficili. Non ci sono parole sufficienti per ringraziarli. Vi amo.

Un sentito grazie va anche al professor Lavagno, il mio relatore, per la sua competenza, disponibilità che mi ha fornito durante la stesura di questa tesi.

Vorrei ringraziare Oscar e Mattia, che mi hanno accompagnato lungo il percorso lavorativo della tesi. La vostra collaborazione, il vostro sostegno e la vostra amicizia hanno reso questo viaggio molto più ricco e significativo. Grazie per aver condiviso con me le sfide e le soddisfazioni di questa esperienza.

Ma questo viaggio è iniziato molto tempo fa, quando un piccolo ragazzino che non conosceva nessuno si è trasferito nella grande città di Torino.

Un ringraziamneto a Maria Antonietta, Nataly e Lucrezia che ci sono state con me dal primo giorno e non mi hanno mai fatto sentito solo.

A Letizia che mi ha spronato ad essere una persona migliore e non riesco a immaginare quanto brutto sarebbe stato il lockdown senza di te (probablimente sarei ancora a dare APA)

A Matteo, a cui posso confidare qualsiasi cosa, sapendo che capirà sempre, ad Alekos, che è la (grande) spalla su cui posso sempre contare e ad Andrea che mi fa sentire a casa ovunque c'è lui. Grazie per essere stati non solo dei compagni di viaggio, ma dei veri amici. Ognuno di voi ha avuto un ruolo fondamentale in questo percorso, e senza il vostro supporto, le vostre risate e la vostra compagnia, tutto sarebbe stato diverso.

Abbiamo condiviso momenti di fatica, notti insonni, ma anche tante soddisfazioni e risate che porterò sempre nel cuore. Insieme abbiamo affrontato sfide, ci siamo incoraggiati a vicenda e, soprattutto, ci siamo divertiti lungo il cammino.

Spero che la nostra amicizia continui a crescere, anche dopo che questo capitolo si sarà chiuso. Con tutto il cuore, grazie per aver reso questo viaggio speciale.

A Gaia, Simone, Lorenza e Beatrice che nonstante la lontananza sono sempre stati vicini a me.

Infine a mia nonna, perchè non mi sono mai dimenticato della promessa che ti ho fatto. E finalmente l'ho mantenuta.

A tutti voi, il mio più sincero grazie.

Indice

Elenco delle tabelle

Elenco delle figure

"Il futuro inizia oggi, non domani."
[Papa Giovanni Paolo II]

Parte I

Introduzione

Capitolo 1

Introduzione generale

Negli ultimi decenni, l'intelligenza artificiale (AI) ha rivoluzionato il modo in cui lavoriamo, influenzando una vasta gamma di settori e professioni. Dall'assistenza sanitaria alla finanza, dall'industria manifatturiera all'educazione, l'AI è diventata uno strumento indispensabile, capace di migliorare l'efficienza, ridurre i costi e aprire nuove opportunità. La capacità dell'AI di analizzare grandi quantità di dati, apprendere da essi e fare previsioni accurate ha trasformato compiti complessi in operazioni più gestibili e precise. Questo progresso tecnologico non solo ha aumentato la produttività, ma ha anche permesso ai lavoratori di concentrarsi su attività di maggiore valore aggiunto, lasciando alle macchine i compiti più ripetitivi e dispendiosi in termini di tempo.

Nel settore energetico, l'integrazione delle tecnologie di robotica e computer vision sta trasformando profondamente la gestione e l'inventario delle cabine elettriche. La computer vision, una branca dell'AI, consente ai sistemi di riconoscere e interpretare le informazioni visive dalle immagini digitali, facilitando l'automazione e la precisione nel monitoraggio delle infrastrutture elettriche. La combinazione di robotica e computer vision permette la realizzazione di ispezioni più sicure ed efficienti, riducendo il rischio per i lavoratori e migliorando la tempestività degli interventi di manutenzione.

Questa tesi, intitolata "Trasformare l'Inventario delle Cabine Elettriche con la Computer Vision nel Settore Energetico, grazie a tecnologie di Robotica e Computer Vision", esplorerà come l'applicazione di queste tecnologie avanzate possa ottimizzare il processo di gestione delle cabine elettriche. Attraverso l'analisi di casi studio e l'implementazione di prototipi, verranno evidenziati i benefici in termini di efficienza operativa, precisione dei dati e sicurezza sul lavoro . Inoltre, verranno discussi i potenziali ostacoli e le soluzioni per superare le sfide tecniche e operative associate a questa trasformazione.(Siciliano and Khatib [2016], Szeliski [2010])

Capitolo 2

Nozioni generali

In questo capitolo verranno presentate le principali nozioni e tecnologie che costituiscono la base del lavoro di ricerca svolto durante la tesi. Questo lavoro è stato sviluppato in collaborazione con Reply, una delle più grandi aziende di consulenza informatica a livello nazionale. In particolare, ho avuto l'opportunità di lavorare con Sprint Reply, un'unità specializzata nell'offrire soluzioni agili e personalizzate per i clienti, in collaborazione con una delle principali aziende italiane operanti nel mondo dell'energia.

Oltre alla descrizione del contesto aziendale, in questo capitolo verranno approfondite le tecnologie chiave utilizzate nel progetto: l'OCR (Optical Character Recognition), i modelli generativi di intelligenza artificiale e la piattaforma cloud AWS (Amazon Web Services). Queste tecnologie sono centrali per lo sviluppo e la realizzazione dei prototipi discussi nella tesi.

2.1 Reply

Reply è un network di aziende che offre servizi di consulenza, integrazione di sistemi e digitalizzazione dei processi aziendali. Con sede principale in Italia e una forte presenza internazionale, Reply si distingue per la sua capacità di combinare competenze tecnologiche avanzate con un approccio innovativo per risolvere le sfide aziendali dei suoi clienti.(Reply [a])

2.2 Sprint Reply

Reply Sprint, una delle unità operative di Reply, è specializzata nell'adozione e implementazione di metodologie agili per accelerare il time-to-market dei progetti. Sprint si concentra sulla gestione efficiente e rapida di progetti complessi, facilitando la collaborazione tra i team e l'adozione di pratiche moderne per garantire risultati di alta qualità in tempi ridotti.(Reply [b])

2.3 OCR

L'OCR (Optical Character Recognition) è una tecnologia che permette di convertire immagini di testo scritto a mano o stampato in testo digitale. Questa tecnologia è ampiamente utilizzata per digitalizzare documenti cartacei, rendendoli modificabili e ricercabili. (Smith [2007]) Consente di convertire diversi tipi di documenti, come documenti cartacei, file PDF o immagini catturate da una fotocamera digitale, in dati modificabili e ricercabili.

2.4 Modello generativo

I modelli di intelligenza artificiale generativa sono un tipo di modello di machine learning progettato per generare nuovi dati che assomigliano a quelli su cui sono stati addestrati. Questi modelli possono creare immagini, testi, musica, e altri tipi di media. Gli esempi più noti al momento sono GPT-3 e Claude.(Goodfellow et al. [2014], Russell and Norvig [2021])

2.5 AWS

Amazon Web Services (AWS) è una piattaforma di servizi cloud offerta da Amazon, che fornisce una vasta gamma di servizi di calcolo, archiviazione, database, machine learning e altre funzionalità a livello globale. AWS consente alle aziende di scalare rapidamente le loro operazioni senza dover investire in infrastrutture fisiche costose. (Barr [2013])

Capitolo 3

Obiettivi della tesi

L'idea centrale della mia tesi è quella di sviluppare una soluzione basata su Amazon Web Services (AWS) che possa gestire e ottimizzare la lettura automatica delle targhette elettriche presenti su vari apparecchi elettrici. Questo progetto prevede la presenza di un operatore che scatti una foto verso la targhetta in questione e che questo venga passato all'interno di un sistema creato appositamente per questo progetto. L'idea è quella di adattare un OCR (Optical Character Recognition) avanzato o, in alternativa, un modello generativo in grado di interpretare e decifrare le informazioni riportate su queste targhette.

Le targhette elettriche contengono dati cruciali riguardanti la sicurezza, le specifiche tecniche e le caratteristiche degli apparecchi. Tuttavia, la loro lettura manuale può essere un compito ripetitivo e soggetto a errori, specialmente quando si tratta di grandi quantità di dispositivi.

Per affrontare questa sfida, il progetto mira a utilizzare le potenzialità dei servizi cloud di AWS per costruire un sistema robusto e scalabile.

In particolare, il sistema proposto sarà in grado di:

1. **Sviluppare i possibili modelli**
   (a) **Riconoscimento Ottico dei Caratteri (OCR):** Implementare un algoritmo di OCR che utilizzi tecnologie come Amazon Textract e/o AWS Rekognition per estrarre testo dalle immagini delle targhette elettriche. Questo processo include la pre-elaborazione delle immagini per migliorare la qualità e l'accuratezza del riconoscimento.
   (b) **Modello Generativo:** Sviluppare un modello generativo che, tramite tecniche di apprendimento automatico, possa non solo leggere ma anche interpretare e contestualizzare le informazioni delle targhette. Questo modello potrebbe includere reti neurali convoluzionali (CNN) e modelli di trasformatore per migliorare la precisione nella lettura e nella comprensione dei dati.
2. **Sviluppare un'architettura di sostegno**
   Progettare un'architettura che sfrutti i servizi AWS per garantire una gestione efficiente delle risorse e una scalabilità dinamica. Questo includerà l'uso di AWS Lambda per l'elaborazione serverless, Amazon S3 per l'archiviazione delle immagini e dei

risultati e Amazon RDS per avere un database capace di salvare dati e interagire con il resto dell'architettura.

3. **Valutazione e Ottimizzazione**
   Creare un sistema di valutazione per monitorare l'accuratezza del riconoscimento e del modello generativo. Implementare meccanismi di feedback per affinare continuamente le prestazioni del sistema e adattarsi a diverse tipologie di targhette e condizioni ambientali.

Il risultato atteso è un sistema automatizzato e altamente efficiente che riduce il margine di errore umano nella lettura delle targhette elettriche e facilita la gestione dei dati tecnici e di sicurezza degli apparecchi elettrici. Questo progetto non solo contribuirà a migliorare la precisione e la velocità di raccolta delle informazioni, ma potrà anche essere adattato e scalato per applicazioni future in altri ambiti di riconoscimento automatico.

Capitolo 4

Struttura della tesi

Questa tesi è suddivisa in cinque sezioni principali, considerando l'introduzione, ciascuna delle quali tratta un aspetto specifico del progetto svolto. Di seguito viene fornita una panoramica della struttura della tesi e dei contenuti di ciascun capitolo.

Parte II: Background

La seconda parte della tesi si concentra sul background tecnologico, con un'attenzione particolare ai servizi AWS utilizzati nel progetto. Include un unico capitolo:

- **Capitolo 5: Servizi AWS usati**: Questo capitolo descrive i vari servizi AWS utilizzati durante il progetto, come Rekognition, Textract, Bedrock, Lambda, Gateway API, RDS, e CloudWatch. Ogni sezione esplora il ruolo specifico di ciascun servizio nel contesto dell'implementazione del sistema.

Parte III: Sviluppo e Implementazione

La terza parte della tesi riguarda lo sviluppo e l'implementazione del sistema proposto. Comprende tre capitoli:

- **Capitolo 6: Primo approccio**: Questo capitolo illustra il primo approccio adottato per affrontare il problema, discutendo le scelte iniziali e le eventuali difficoltà incontrate.
- **Capitolo 7: Sviluppo modelli Rekognition e Textract**: Qui vengono presentati i dettagli dello sviluppo dei modelli basati su Rekognition e Textract. Il capitolo è suddiviso in sezioni che analizzano singolarmente l'implementazione di ciascun modello.
- **Capitolo 8: Implementazione modello generativo**: Questo capitolo si concentra sull'implementazione del modello generativo, suddividendosi in sezioni che trattano la generazione di testo, l'uso di immagini con prompt e l'integrazione degli embeddings.

- **Capitolo 9: Implementazione Lambda**: Viene descritto il processo di implementazione delle funzioni Lambda, con un'attenzione particolare alla creazione dell'API Lambda e alla costruzione della pipeline di Lambda.

Parte IV: Analisi e Risultati

La quarta parte della tesi si occupa dell'analisi dei risultati ottenuti e della scelta del miglior modello. Include due capitoli:

- **Capitolo 10: Scelta miglior modello Textract: Questo capitolo discute i criteri di selezione del miglior modello Yextract, presentando i risultati delle valutazioni eseguite.**
- **Capitolo 11: Scelta miglior modello Bedrock**: Questo capitolo discute i criteri di selezione del miglior modello Bedrock, presentando i risultati delle valutazioni eseguite.
- **Capitolo 12: Claude-3: Descrizione e Confronto tra le Varianti Haiku e Sonnet**: Questo capitolo offre una panoramica dettagliata del modello Claude-3, analizzando le varianti Haiku e Sonnet e confrontando le loro caratteristiche e prestazioni.
- **Capitolo 13: Scelta finale miglior modello**: Qui viene descritta la selezione finale del miglior modello, basata sui risultati ottenuti nelle analisi precedenti.

Parte V: Conclusione

La quinta e ultima parte della tesi consiste in un solo capitolo:

- **Capitolo 14: Conclusione**: Questo capitolo riassume i risultati raggiunti nel corso del progetto, riflettendo sull'efficacia delle soluzioni proposte e suggerendo possibili sviluppi futuri.

Parte II

Background

Capitolo 5

Servizi AWS usati

In questo capitolo verranno descritti i principali servizi AWS impiegati nel corso del progetto. Verranno analizzate le funzionalità offerte da ciascun servizio, il loro ruolo specifico all'interno dell'architettura complessiva, e le motivazioni che hanno guidato la loro scelta. L'obiettivo è fornire una panoramica dettagliata delle risorse cloud utilizzate, evidenziando come ciascuna di esse abbia contribuito al raggiungimento degli obiettivi prefissati e alla risoluzione delle problematiche tecniche incontrate. (Barr [2013]) (Russell and Norvig [2021])

5.1 Rekognition

Amazon Rekognition semplifica l'aggiunta di analisi di immagini e video alle applicazioni utilizzando una tecnologia di deep learning collaudata e altamente scalabile. Con Amazon Rekognition, puoi identificare oggetti, persone, testo, scene e attività in immagini e video, oltre a rilevare eventuali contenuti inappropriati. Amazon Rekognition offre anche funzionalità di analisi e ricerca facciale estremamente accurate che puoi utilizzare per rilevare, analizzare e confrontare i volti per un'ampia varietà di casi d'uso in materia di verifica degli utenti, conteggio delle persone e sicurezza pubblica. Con Amazon Rekognition Custom Labels, puoi identificare gli oggetti e le scene nelle immagini che sono specifiche per le tue esigenze aziendali. Ad esempio, puoi creare un modello per classificare parti specifiche della macchina sulla linea di assemblaggio o per rilevare piante non sane. Amazon Rekognition Custom Labels si occupa dello sviluppo di modelli al posto tuo, quindi non è richiesta alcuna esperienza di machine learning. Devi semplicemente fornire immagini di oggetti o scene che desideri identificare e il servizio si occuperà del resto.

Figura 5.1. Icona di AWS Rekognition

5.2 Bedrock

Amazon Bedrock è un servizio completamente gestito che offre una scelta di modelli di fondazione (FM) ad alte prestazioni delle principali aziende di AI, come AI21 Labs, Anthropic, Cohere, Meta, Mistral AI, Stability AI e Amazon, tramite un'unica API, insieme ad un'ampia gamma di funzionalità necessarie per creare applicazioni di AI generativa, utilizzando l'AI in modo sicuro, responsabile e nel rispetto della privacy. Utilizzando Amazon Bedrock, è possibile sperimentare e valutare facilmente i migliori FM per il proprio caso d'uso, personalizzarli privatamente con i propri dati utilizzando tecniche come l'ottimizzazione e la generazione aumentata dei recuperi (Retrieval Augmented Generation, RAG), e creare agenti che eseguono attività utilizzando i sistemi e le origini dati aziendali. Poiché Amazon Bedrock è serverless, non è necessario gestire alcuna infrastruttura, mentre è possibile integrare e implementare in modo sicuro funzionalità di AI generativa nelle applicazioni utilizzando i servizi AWS già noti.

Figura 5.2. Icona di AWS Bedrock

5.3 Textract

Amazon Textract è un servizio che estrae automaticamente testo e dati dai documenti scansionati. Amazon Textract va ben oltre il semplice riconoscimento OCR per identificare anche il contenuto dei campi nei moduli e le informazioni memorizzate nelle tabelle. Oggi, molte aziende estraggono manualmente i dati da documenti scansionati come PDF, immagini, tabelle e moduli, oppure tramite un semplice software OCR che richiede una configurazione manuale (che spesso deve essere aggiornato quando il modulo cambia). Per superare questi processi manuali e costosi, Amazon Textract utilizza il machine learning per leggere ed elaborare qualsiasi tipo di documento, estraendo accuratamente testo, grafia, tabelle e altri dati senza alcuno sforzo manuale. Amazon Textract ti offre la flessibilità di specificare i dati che devi estrarre dai documenti utilizzando le query. Puoi specificare le informazioni di cui hai bisogno sotto forma di domande in linguaggio naturale (come «Qual è il nome del cliente»). Non è necessario conoscere la struttura dei dati nel documento (tabella, modulo, campo implicito, dati annidati) né preoccuparsi delle variazioni tra le Amazon Textract. Le query di Amazon Textract sono preformate su un'ampia gamma di documenti, tra cui buste paga, estratti conto bancari, documenti W-2, moduli di richiesta di prestito, note ipotecarie, documenti relativi ai reclami e tessere assicurative. Con Amazon Textract, puoi automatizzare rapidamente l'elaborazione dei documenti e agire in base alle informazioni estratte, sia che tu stia automatizzando l'elaborazione dei prestiti o estraendo informazioni da fatture e ricevute. Amazon Textract può estrarre i dati in pochi minuti anziché in ore o giorni.

Figura 5.3. Icona di AWS Textract

5.4 Lambda

AWS Lambda è un servizio di elaborazione che esegue il codice in risposta agli eventi e gestisce automaticamente le risorse di elaborazione, rendendolo il modo più veloce per trasformare un'idea in moderne applicazioni di produzione serverless. AWS Lambda consente di eseguire il codice senza effettuare il provisioning dei server o senza gestirli. Con Lambda, puoi eseguire codice praticamente per qualsiasi tipo di applicazione o servizio di backend, il tutto senza alcuna amministrazione. Basta caricare il codice e Lambda si occuperà di tutto il necessario per eseguire e scalare il codice con un'elevata disponibilità.

Figura 5.4. Icona di AWS Lambda

5.5 Gateway API

Amazon API Gateway è un servizio completamente gestito che semplifica la creazione, la pubblicazione, la manutenzione, il monitoraggio e la protezione delle API su qualsiasi scala. Con AWS Management Console, puoi creare un'API che funga da «porta d'ingresso» per consentire alle applicazioni di accedere ai dati, alla logica di business o alle funzionalità dei tuoi servizi di back-end, come il codice in esecuzione su AWS Lambda o qualsiasi applicazione web. Amazon API Gateway gestisce tutte le attività legate all'accettazione e all'elaborazione di centinaia di migliaia di chiamate API simultanee, tra cui la gestione del traffico, il controllo delle autorizzazioni e degli accessi, il monitoraggio e la gestione delle versioni delle API

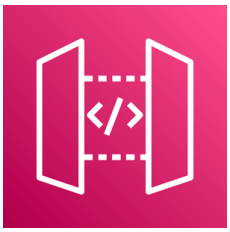

Figura 5.5. Icona di API Gateway

5.6 RDS

Amazon Relational Database Service (Amazon RDS) è un servizio di database relazionale facile da gestire ottimizzato per il costo totale di proprietà. È semplice da configurare, utilizzare e dimensionare in base alla domanda. Amazon RDS automatizza le attività indifferenziate di gestione dei database, come il provisioning, la configurazione, i backup e l'applicazione di patch. Amazon RDS consente ai clienti di creare un nuovo database in pochi minuti e offre la flessibilità di personalizzare i database in base alle proprie esigenze. Ti consente di implementare rapidamente esempi di codice, componenti e applicazioni complete per casi d'uso comuni come backend web e mobili, elaborazione di eventi e dati, registrazione, monitoraggio, Internet of Things (IoT) e altro ancora.

Figura 5.6. Icona di AWS RDS

5.7 CloudWatch

Amazon CloudWatch è un servizio di monitoraggio e gestione creato per sviluppatori, operatori di sistema, ingegneri dell'affidabilità del sito (SRE) e responsabili IT. CloudWatch ti fornisce dati e approfondimenti utilizzabili per monitorare le tue applicazioni, comprendere e rispondere ai cambiamenti delle prestazioni a livello di sistema, ottimizzare l'utilizzo delle risorse e ottenere una visione unificata dello stato operativo. CloudWatch raccoglie dati operativi e di monitoraggio sotto forma di log, metriche ed eventi, fornendo una visione unificata di AWS risorse, applicazioni e servizi eseguiti su server e locali. Puoi utilizzarli per impostare allarmi ad alta risoluzione, visualizzare log e metriche fianco a fianco, intraprendere azioni automatizzate, risolvere problemi e scoprire approfondimenti per ottimizzare le applicazioni e garantire che funzionino senza intoppi.

Figura 5.7. Icona di AWS CloudWatch

Parte III

Sviluppo e Implementazione

Capitolo 6

Primo approccio

L'obiettivo principale dell'estrazione automatica di informazioni visive da una targhetta elettrica è quello di facilitare e automatizzare il lavoro dell'operaio. L'idea generale del progetto consiste nel permettere all'operaio di scattare semplicemente una foto della targhetta elettrica, dopodiché il sistema analizzerà l'immagine e restituirà i dati in maniera ordinata e sistemata.

Per realizzare questo sistema, ho valutato diverse soluzioni offerte da Amazon Web Services (AWS), concentrandomi in particolare su due strumenti avanzati di analisi delle immagini e dei testi: Rekognition e Textract. Di seguito, presento un confronto tra i punti di forza e debolezza di ciascuno di questi servizi, al fine di identificare quello più idoneo alle specifiche esigenze del progetto.

6.1 AWS Rekognition

6.1.1 Punti di Forza

- **Ampio supporto per il riconoscimento di oggetti**: AWS Rekognition è estremamente efficace nel rilevare oggetti, scene, e attività all'interno delle immagini. Questa capacità potrebbe risultare utile per identificare automaticamente elementi visivi presenti sulla targhetta elettrica, come loghi o simboli di avvertenza.
- **Facilità d'integrazione**: Essendo parte dell'ecosistema AWS, Rekognition si integra facilmente con altri servizi AWS, facilitando lo sviluppo di un flusso di lavoro automatizzato e scalabile.

6.1.2 Punti di Debolezza

- **Limitato nel riconoscimento del testo**: Sebbene Rekognition possa rilevare testo all'interno delle immagini, la sua capacità di interpretare e strutturare il testo è limitata rispetto a soluzioni più specifiche come Textract.
- **Dipendenza dalla qualità dell'immagine**: Le prestazioni di Rekognition dipendono fortemente dalla qualità dell'immagine fornita. Immagini sfocate o a bassa

risoluzione possono portare a risultati non accurati, specialmente quando si tratta di testi piccoli o complessi.

6.2 AWS Textract

6.2.1 Punti di Forza

- **Estratto di testo strutturato**: Textract è progettato specificamente per l'estrazione di testo e dati strutturati da documenti. Questo servizio è altamente efficiente nell'identificare blocchi di testo, tabelle, e moduli, garantendo una maggiore accuratezza e precisione nella lettura di targhette elettriche.
- **Capacità di comprensione del layout**: Textract non si limita a leggere il testo, ma comprende anche la struttura del documento, il che è fondamentale per interpretare correttamente le informazioni visive su una targhetta che potrebbe avere una disposizione complessa.
- **Integrazione con altri servizi AWS**: Come Rekognition, Textract si integra facilmente con l'ecosistema AWS, permettendo di costruire pipeline automatizzate per l'elaborazione e l'analisi dei dati.

6.2.2 Punti di Debolezza

- **Meno efficace nel riconoscimento di oggetti visivi**: Textract è principalmente orientato all'analisi del testo e non offre le stesse capacità avanzate di Rekognition nel riconoscimento di oggetti e scene visive.
- **Prestazioni variabili con immagini non standard**: Mentre Textract eccelle nell'estrazione di testo da documenti ben strutturati, potrebbe non essere altrettanto efficace con immagini di qualità inferiore o con testi disposti in modo non convenzionale, come quelli su targhette usurate o con sfondi complessi.

6.3 Considerazioni Finali

Dal confronto emerge che la scelta degli OCR, AWS Rekognition e AWS Textract dipende strettamente dalle specifiche esigenze del progetto. L'obiettivo principale è l'estrazione accurata e strutturata di testo, quindi Textract, in questo caso, si presenta come la soluzione più adatta, grazie alle sue capacità di comprensione del layout e all'accuratezza nell'interpretazione del testo.

Capitolo 7

Sviluppo modelli Rekognition e Textract

Dal capitolo precedente emerge chiaramente che AWS Rekognition non sia l'OCR ideale che stiamo cercando. Tuttavia, approfondendo le funzionalità del servizio e riguardando i punti di forza, si scopre che Rekognition offre molto di più rispetto alla semplice capacità di riconoscere il testo. Una delle caratteristiche più potenti è la Custom Labels, che consente di addestrare modelli personalizzati per rilevare oggetti o caratteristiche specifiche all'interno delle immagini.

Rekognition Custom Labels permette di creare modelli di machine learning personalizzati per individuare oggetti, scene, o dettagli che sono rilevanti per uno specifico caso d'uso. Questo strumento può essere particolarmente utile per il nostro progetto, poiché possiamo addestrare un modello che riconosca con precisione le targhette all'interno delle foto scattate. Una volta individuata la targhetta, Rekognition può isolare l'area di interesse, eliminando il rumore e le informazioni irrilevanti attorno ad essa.

Successivamente, questa immagine pulita e focalizzata può essere passata ad AWS Textract, ottimizzando così il processo di OCR e garantendo risultati più accurati ed efficienti. Textract, lavorando su un'immagine priva di elementi di disturbo, è in grado di estrarre il testo con la massima precisione possibile.

Di seguito, descrivo come ho costruito ciascuno di questi modelli utilizzando i servizi offerti da AWS:

7.1 Creazione di un Modello con AWS Rekognition Custom Labels

Creare un modello personalizzato con AWS Rekognition Custom Labels richiede una serie di passaggi che consentono di addestrare il modello a riconoscere specifici oggetti o caratteristiche nelle immagini. Di seguito vengono descritti i vari step necessari.

7.1.1 Preparazione dei Dati

Il primo passo è raccogliere un set di immagini rappresentative che contengano gli oggetti o le caratteristiche da riconoscere. È importante avere un numero adeguato di immagini per ogni categoria (classe) che si intende addestrare. Successivamente, le immagini devono essere etichettate manualmente per indicare con precisione dove si trovano gli oggetti di interesse, utilizzando strumenti di annotazione che permettono di disegnare riquadri attorno agli oggetti e assegnare loro etichette appropriate.

7.1.2 Creazione di un Progetto, Caricamento e Gestione del Dataset

Per iniziare, accedi alla Console AWS e vai alla sezione *Rekognition Custom Labels*. Qui, puoi creare un nuovo progetto, assegnandogli un nome e una descrizione. Il progetto funge da contenitore logico per tutti i dataset, i modelli e le relative operazioni. Una volta creato il progetto, carica il dataset etichettato nella console di Custom Labels. È consigliabile dividere il dataset in due parti: un set di addestramento (training dataset) e un set di test (test dataset). Il set di addestramento servirà per addestrare il modello, mentre il set di test verrà utilizzato per valutare le sue prestazioni. Verifica che tutte le etichette siano corrette e che i dati siano organizzati in modo coerente.

7.1.3 Addestramento del Modello

Con il dataset pronto, puoi configurare il processo di addestramento. AWS Rekognition si occuperà automaticamente della creazione del modello di machine learning. Durante questa fase, puoi scegliere parametri come il numero di epoche e la suddivisione dei dati tra training e test. Una volta configurato, avvia l'addestramento. AWS Rekognition analizzerà le immagini, identificherà le caratteristiche rilevanti e costruirà un modello capace di riconoscere le etichette nelle nuove immagini.

7.1.4 Valutazione del Modello

Al termine dell'addestramento, AWS fornirà una valutazione delle prestazioni del modello, includendo metriche come accuratezza, precision, recall e matrice di confusione. Queste informazioni ti aiuteranno a capire quanto il modello sia efficace. Se necessario, potrai ottimizzarlo ulteriormente aggiungendo più dati, migliorando le etichette o regolando i parametri di addestramento.

7.1.5 Distribuzione e Utilizzo del Modello

Dopo la valutazione, il modello può essere pubblicato e reso disponibile per l'uso tramite l'API di AWS Rekognition. Potrai integrarlo in applicazioni esistenti o pipeline di elaborazione immagini, ad esempio utilizzandolo in un codice Python attraverso AWS Lambda. Se in futuro dovessero emergere nuove esigenze, come il riconoscimento di nuovi oggetti o

condizioni di illuminazione diverse, sarà possibile aggiornare il modello con nuovi dati e riaddestrarlo.

Questo processo ti permette di creare un modello su misura per le esigenze specifiche del tuo progetto, capace di riconoscere oggetti o caratteristiche particolari nelle immagini.

7.2 Creazione di un Modello con AWS Textract

AWS Textract è un potente servizio che consente di estrarre testo e dati strutturati, come tabelle e moduli, da documenti scansionati. A differenza di un semplice OCR, Textract comprende la struttura dei documenti e organizza le informazioni in modo coerente. Di seguito è descritto il processo per utilizzare Textract al meglio nel nostro progetto.

7.2.1 Preparazione dei Dati

Il primo passo consiste nel raccogliere i documenti da analizzare, che possono essere immagini o PDF contenenti testo e dati strutturati come tabelle o moduli. È essenziale verificare che i documenti siano di buona qualità, con testo leggibile e ben contrastato, poiché documenti di scarsa qualità possono compromettere l'accuratezza dell'estrazione.

7.2.2 Caricamento dei Documenti su S3

Prima di utilizzare Textract, i documenti devono essere caricati su Amazon S3, il servizio di storage di AWS. Textract opera direttamente sui file presenti in S3, quindi è importante che i documenti siano accessibili in un bucket. Organizzare i documenti in cartelle può facilitare la gestione, soprattutto quando si lavora con grandi volumi di dati.

7.2.3 Configurazione dell'Estrazione

Textract offre diverse opzioni per l'estrazione di dati. È possibile scegliere tra:

- **Estratto di Testo Semplice (Plain Text)**: L'opzione di base che estrae tutto il testo trovato nel documento, senza tener conto della struttura. Ho scartato a priori questa opzione, perchè molto simile a quello che ritorna lOCR di Rrkognition
- **Estrazione di Tabelle**: Se il documento contiene tabelle, Textract può identificare e organizzare i dati tabulari in un formato strutturato. Questa opzione è utile per estrarre dati che sono naturalmente organizzati in righe e colonne.
- **Estrazione di Moduli e Campi**: Textract può anche riconoscere campi specifici all'interno di moduli, permettendo di estrarre coppie chiave-valore che rappresentano informazioni strutturate come nomi, indirizzi, date, ecc.
- **Custom Queries**: Textract permette di creare Custom Queries per estrarre informazioni altamente specifiche da documenti complessi. Ad esempio, è possibile definire una query personalizzata che cerca una frase specifica o un concetto particolare all'interno del documento. Queste query possono essere configurate per rispondere

a domande come "Qual è l'importo totale della fattura?" o "Chi è il destinatario principale del documento?".

Nella configurazione, è importante selezionare le opzioni di estrazione che meglio si adattano alle esigenze del progetto.

7.2.4 Esecuzione del Processo di Estrazione

Una volta configurate le opzioni, è possibile avviare il processo di estrazione tramite l'API di Textract o utilizzando la console AWS. Textract analizzerà i documenti e restituirà i dati estratti nel formato scelto, come JSON. Questi risultati possono essere gestiti direttamente o integrati con altri servizi AWS per ulteriori elaborazioni.

7.2.5 Analisi e Utilizzo dei Risultati

Dopo l'estrazione, è importante rivedere e validare i dati per assicurarsi che Textract abbia identificato correttamente tutte le informazioni necessarie. I dati estratti possono essere integrati in applicazioni esistenti o utilizzati per ulteriori analisi, come l'inserimento di dati tabulari in database o fogli di calcolo per un'analisi più approfondita.

Se necessario, il processo può essere automatizzato, configurando Textract per analizzare nuovi documenti caricati su S3 e processarli automaticamente, o tramite script. Questo approccio permetterà anche di combinare le funzionalità di Rekognition con Textract. In caso di necessità, è possibile aggiornare e ottimizzare il processo di estrazione aggiungendo nuove regole o migliorando le opzioni di estrazione per ottenere risultati più accurati.

Questo processo consente di configurare AWS Textract per l'estrazione precisa e strutturata dei dati dai documenti, adattandosi alle esigenze specifiche del progetto e sfruttando le Custom Queries per ottenere informazioni mirate e specifiche.

Capitolo 8

Implementazione modello generativo

Dopo aver discusso nei capitoli precedenti l'utilizzo di sistemi di riconoscimento ottico dei caratteri (OCR) tramite AWS per estrarre dati dai documenti, in questo capitolo verrà esplorato l'uso di modelli generativi come ulteriore strumento per l'analisi e l'interpretazione dei dati. Per raggiungere questo obiettivo, abbiamo scelto di utilizzare AWS Bedrock, una piattaforma che facilita l'accesso a modelli generativi avanzati.

AWS Bedrock permette l'integrazione di vari modelli di intelligenza artificiale, inclusi modelli linguistici, di immagini e di *embeddings*, offrendo un'infrastruttura potente e flessibile per creare applicazioni basate su AI. In questo capitolo verranno illustrate tre modalità di utilizzo dei modelli generativi di AWS Bedrock: attraverso prompt di solo testo, combinando testo e immagini, e utilizzando *embeddings*.

8.1 Utilizzo del Modello con Prompt di Solo Testo

La prima modalità di utilizzo di AWS Bedrock riguarda l'uso di prompt di solo testo. In questo scenario, abbiamo sfruttato modelli linguistici avanzati per generare contenuti basati sui dati estratti tramite OCR.

8.1.1 Procedura

1. **Preparazione del Prompt**: I dati testuali estratti dai documenti tramite OCR sono stati utilizzati come input per il modello generativo. Il prompt è stato strutturato per chiedere al modello di elaborare, riassumere o ampliare il testo.

2. **Configurazione del Modello**: Utilizzando l'API di AWS Bedrock, è stato scelto un modello linguistico pre-addestrato, configurando i parametri del prompt per ottenere risultati ottimali.

3. **Generazione del Contenuto**: Il modello ha generato contenuti basati sul prompt fornito. Questi contenuti sono stati utilizzati per migliorare l'interpretazione dei dati

OCR, ad esempio fornendo riassunti automatici dei documenti o suggerimenti per ulteriori analisi.

8.2 Utilizzo del Modello con Prompt di Testo e Immagini

A differenza dei metodi tradizionali che partono dall'OCR per estrarre testo da immagini, La seconda modalità di utilizzo di AWS Bedrock prevede utilizza direttamente il prompt testuale e le immagini fornite dall'utente.

8.2.1 Procedura

1. **Preparazione del Prompt Multimodale**: I dati testuali estratti tramite OCR sono stati combinati con immagini rilevanti, come diagrammi, tabelle o fotografie presenti nei documenti originali.

2. **Configurazione del Modello**: Su AWS Bedrock, è stato selezionato un modello capace di elaborare input multimodali. Il prompt è stato strutturato per integrare sia il testo che le immagini, chiedendo al modello di generare output che considerasse entrambi i tipi di input.

3. **Generazione del Contenuto**: Il modello ha analizzato i dati combinati, generando interpretazioni più ricche e contestuali rispetto all'uso del solo testo. Ad esempio, il modello potrebbe descrivere l'immagine nel contesto del testo o generare conclusioni che collegano visivamente i dati testuali.

8.3 Utilizzo del Modello con Embeddings

In questa sezione, descriviamo l'uso di modelli generativi che sfruttano gli *embeddings* per analizzare dati complessi. Il modello analizza questi input confrontandoli con un database di *embeddings* pre-esistenti, migliorando l'accuratezza e la rilevanza dell'analisi.

8.3.1 Procedura

1. **Analisi del Prompt e delle Immagini**:

   - Il processo inizia con l'invio di un prompt testuale e di un'immagine al modello. Questi dati vengono elaborati dal modello generativo che, utilizzando tecniche avanzate di machine learning, crea *embeddings* per il testo e l'immagine.
   - Gli *embeddings* rappresentano le caratteristiche semantiche e visive del prompt e dell'immagine in uno spazio vettoriale multidimensionale.

2. **Confronto con il Database di Embeddings**:

- Il modello confronta gli *embeddings* generati dal prompt e dall'immagine con quelli presenti in un database preesistente. Questo database contiene *embeddings* di un'ampia varietà di testi e immagini già processati.
- Il confronto si basa sulla somiglianza tra i vettori, identificando l'*embedding* più simile a quello generato dal prompt e dall'immagine forniti dall'utente. Questo permette al modello di trovare riferimenti simili e di contestualizzare meglio l'input ricevuto.

3. **Esecuzione dell'Analisi**:

- Una volta identificato l'*embedding* più simile nel database, il modello utilizza queste informazioni per condurre un'analisi approfondita. La conoscenza acquisita dal database di *embeddings* aiuta a fornire un'interpretazione più precisa e contestualizzata dell'immagine e del testo.
- Ad esempio, se il modello riconosce che l'immagine fornita è simile a un diagramma già presente nel database, utilizza le informazioni associate a quell'*embedding* per offrire un'analisi dettagliata o per generare un output rilevante.

Capitolo 9

Implementazione servizi AWS

Nel contesto di questo progetto, è stato fondamentale creare e configurare una serie di servizi che lavorano in maniera strettamente coordinata tra di loro. I servizi sono stati progettati per essere modulare, ciascuno con un ruolo specifico, ma interconnessi in modo tale da poter scambiare informazioni e reagire agli eventi in tempo reale. Questo approccio non solo migliora l'efficienza operativa, ma permette anche una maggiore scalabilità e flessibilità. Di seguito è riportata una descrizione dei principali componenti e servizi utilizzati nel progetto:

- **Lambda API**: Funzione AWS Lambda che è chiamata dalla richiesta API e funge come l'interazione con il database e il flusso di dati. Questa funzione è responsabile della ricezione delle immagini delle targhette, dell'estrazione dei dati e della comunicazione con la pipeline di Computer Vision con altri servizi AWS.

- **Lambda Pipeline**: Funzione AWS Lambda dedicata all'elaborazione delle immagini e all'estrazione delle informazioni chiave dalle targhette. Utilizzano algoritmi e servizi di Computer Vsion per identificare e interpretare i dati rilevanti.

- **Bucket S3**: Amazon Simple Storage Service (S3) utilizzato per l'archiviazione delle immagini delle targhette caricate. Questo bucket S3 funge da deposito centrale per tutte le immagini che devono essere elaborate dal sistema.

- **API Gateway**: Amazon API Gateway che espone endpoint API per consentire il caricamento delle immagini nel bucket S3 e l'avvio del processo di elaborazione. Questo servizio fornisce un'interfaccia sicura e scalabile per le interazioni esterne con il sistema.

- **RDS**: Amazon Relational Database Service (RDS) utilizzato per memorizzare i dati estratti dalle targhette. Questo database relazionale contiene le informazioni chiave estratte, facilitando la loro gestione e consultazione. Usiamo un database Postgres, in possesso al team Sense Reply.

9.1 Analisi del Codice Python

Per ragioni di riservatezza aziendale, non possiamo condividere direttamente il codice Python. Tuttavia, in questa sezione forniamo una descrizione dettagliata della logica del codice attraverso pseudocodice e spiegazioni approfondite, per aiutare il lettore a comprendere il funzionamento del sistema.

9.1.1 Lambda Api - funzione Lambda

Questa funzione Lambda su AWS è progettata per gestire un processo complesso che include la ricezione e l'elaborazione di un'immagine, l'interazione con il servizio S3 per lo storage, il recupero delle credenziali tramite AWS Secrets Manager, la connessione a un database PostgreSQL su RDS, e l'invocazione di un'altra funzione Lambda per ulteriori elaborazioni. Di seguito, analizziamo in dettaglio ciascuna fase del codice.

Inizializzazione

La funzione principale è strutturata per gestire efficacemente gli eventi Lambda. Inizialmente, viene creato un client dedicato per interagire con il servizio di storage cloud, essenziale per caricare e recuperare immagini. Successivamente, utilizzando l'oggetto di contesto fornito dall'ambiente di esecuzione Lambda, viene estratto un identificativo unico per il gruppo di log. Questo identificativo è fondamentale per monitorare e tracciare le attività e gli eventuali errori che si verificano durante l'esecuzione della funzione.

Decodifica e Caricamento dell'Immagine su Storage Cloud

La funzione gestisce poi le immagini ricevute attraverso una serie di passaggi. Per prima cosa, decodifica il file immagine ricevuto nell'evento, assicurandosi che sia in un formato leggibile e utilizzabile dal sistema. Una volta decodificata, viene generato un nome univoco per l'immagine. Questo passaggio è cruciale per prevenire conflitti e garantire che ogni immagine abbia un'identificazione unica all'interno del sistema. Infine, l'immagine viene caricata nel servizio di storage cloud utilizzando il nome univoco appena generato, assicurandone la conservazione per futuri utilizzi.

Connessione al Database

La funzione prosegue con la gestione della connessione al database. Inizialmente, vengono recuperate le credenziali per la connessione al database da un servizio di gestione dei segreti. Questo metodo garantisce che le informazioni sensibili siano protette. Con le credenziali in mano, la funzione costruisce l'URI necessario per stabilire la connessione al database. Dopo aver preparato l'URI, la funzione stabilisce la connessione al database, rendendolo pronto per le operazioni successive.

Inserimento di un Record nel Database

Una volta stabilita la connessione al database, la funzione si occupa di inserire un nuovo record nella tabella del database. Questo record contiene i dettagli relativi all'immagine appena caricata. Dopo l'inserimento, la funzione recupera l'ID del nuovo record. Questo ID è fondamentale per eventuali riferimenti futuri e per possibili aggiornamenti. Infine, la funzione conferma le modifiche nel database, assicurandosi che i dati siano correttamente salvati.

Invocazione di un'Altra Funzione Lambda per Ulteriori Elaborazioni

La funzione Lambda procede invocando un'altra funzione Lambda per eseguire ulteriori elaborazioni. Per fare ciò, viene prima creato un client per gestire l'invocazione della seconda funzione Lambda. Successivamente, la funzione invoca la Lambda secondaria, passando i dettagli dell'immagine e l'ID del task come parametri necessari per le elaborazioni successive. Durante questo processo, viene registrata la durata dell'invocazione, un dato utile per monitorare e ottimizzare le prestazioni del sistema.

Aggiornamento del Database con i Risultati della Pipeline

Dopo l'elaborazione da parte della seconda funzione Lambda, i risultati vengono utilizzati per aggiornare il database. La funzione legge la risposta fornita dalla Lambda invocata e analizza i dettagli necessari per l'aggiornamento del database. Questi dettagli vengono poi utilizzati per aggiornare il record corrispondente nel database. Dopo aver apportato gli aggiornamenti necessari, la funzione conferma le modifiche e chiude la connessione al database in modo sicuro.

Gestione degli Errori e Risposta della Funzione Lambda

Infine, la funzione Lambda è dotata di meccanismi per gestire eventuali errori che possono verificarsi durante l'elaborazione o l'aggiornamento del database. Se tutto procede correttamente, la funzione restituisce un codice di stato 200 con i dettagli del task completato. In caso di errore, invece, viene restituito un codice di errore 500, segnalando che qualcosa è andato storto durante il processo.

9.1.2 Lambda Pipeline - funzione Lambda

Il codice Python definisce una seconda funzione Lambda su AWS, responsabile dell'interazione con S3, dell'elaborazione delle immagini e dell'invocazione di modelli di machine learning tramite Bedrock. Di seguito, viene fornita una descrizione dettagliata delle sue principali componenti.

Funzione Lambda Handler

La funzione principale si occupa di gestire gli eventi Lambda. Inizia estraendo l'ID del gruppo di log dall'oggetto contesto fornito dall'ambiente Lambda. Questo ID è utile per

tracciare l'esecuzione della funzione. Inoltre, viene registrato l'orario di inizio, un'informazione utilizzata per monitorare le prestazioni. La funzione poi avvia un blocco `try` per gestire eventuali errori che potrebbero sorgere durante l'esecuzione. All'interno di questo blocco, viene creato un client per interagire con lo storage cloud. Successivamente, la funzione estrae i valori image_key e task_id dall'evento Lambda e li registra nei log per facilitare il debug e la tracciabilità.

Se si verifica un errore durante questi passaggi, la funzione restituisce un codice 500 accompagnato da un messaggio di errore, segnalando che si è verificato un problema.

Download dell'Immagine dal Servizio di Storage Cloud

La funzione tenta di scaricare l'immagine specificata dal servizio di storage cloud. Dopo aver scaricato l'immagine, questa viene ridimensionata per adattarsi ai requisiti del modello di machine learning che verrà utilizzato in seguito. Se si verifica un errore durante il processo di download o ridimensionamento, la funzione restituisce un codice 500 con un messaggio di errore.

Recupero degli Embeddings

La funzione cerca di recuperare un file di embedding dallo storage cloud. Una volta recuperato, i dati vengono caricati e convertiti in una struttura idonea per ulteriori elaborazioni. La funzione poi calcola l'embedding dell'immagine scaricata, confrontandolo con quelli esistenti per identificare la corrispondenza più vicina.

Invio della Richiesta al Modello

Dopo aver elaborato l'immagine, la funzione invia una richiesta al modello di intelligenza artificiale con i dati elaborati. La risposta del modello viene analizzata per estrarre informazioni chiave come il produttore, il modello e il numero di serie.

Restituzione della Risposta

Infine, la funzione costruisce una risposta API con i dati estratti e restituisce un oggetto JSON con codice 200, segnalando il successo dell'operazione.

Funzioni di Supporto

- **Funzione di Ridimensionamento Immagine:** Ridimensiona l'immagine mantenendo il rapporto d'aspetto e la converte in un formato compatibile.
- **Funzione per l'Invio di Richieste al Modello:** Invia una richiesta al modello, codifica le immagini e costruisce il corpo della richiesta in formato JSON.
- **Funzione per il Calcolo degli Embeddings:** Calcola l'embedding ridimensionando l'immagine e inviandola a un servizio di machine learning.
- **Funzione per il Confronto degli Embeddings:** Confronta l'embedding dell'immagine con un database di embeddings, restituendo il dato più simile.

9.2 Funzionamento del Progetto

Descrivo, infine, il flusso operativo complessivo del progetto, spiegando in dettaglio come i diversi componenti già analizzati nei capitoli precedenti interagiscono tra di loro per realizzare il processo di elaborazione delle immagini delle targhette elettriche.

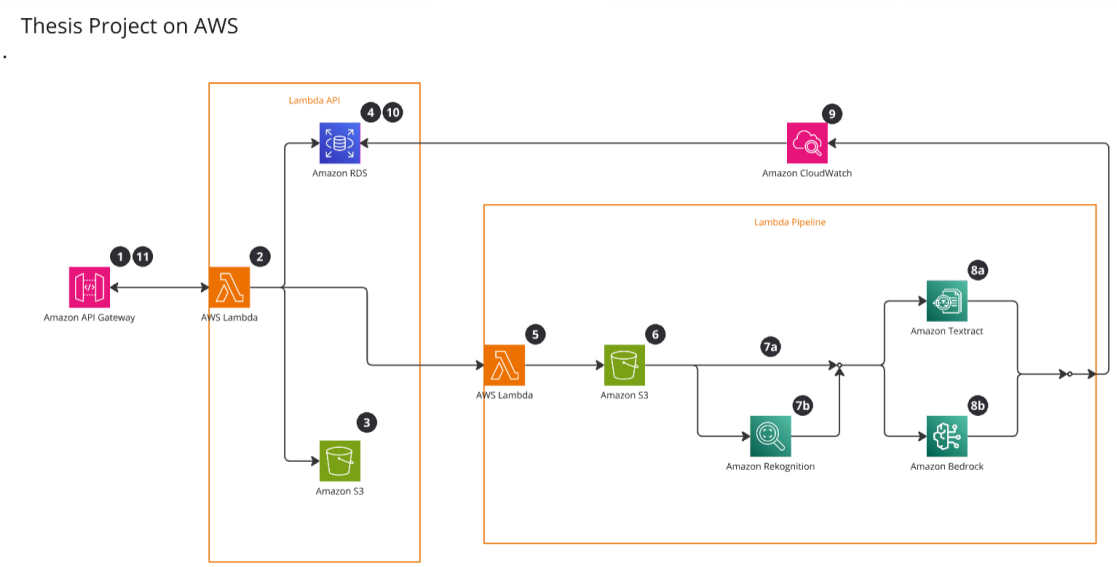

Figura 9.1. Diagramma architetturale del progetto AWS

9.2.1 Scatto della Foto e Invio al Gateway API

Il processo inizia con l'operatore che scatta una foto alla targhetta elettrica. Questa immagine viene inviata al Gateway API, il quale funge da punto di ingresso per l'intero sistema. Il Gateway API è responsabile di ricevere la richiesta contenente l'immagine e di inoltrarla all'Lambda API per l'elaborazione successiva.

9.2.2 Gestione della Foto tramite Lambda API

L'Lambda API riceve la foto dal Gateway API e svolge diverse funzioni. In primo luogo, salva l'immagine all'interno di un bucket S3. Successivamente, crea una nuova entry all'interno del database, dove verranno memorizzati i dati relativi all'immagine e i risultati dell'elaborazione.

Dopo aver completato queste operazioni, l'Lambda API attiva un'altra lambda (Lambda Pipeline) che gestisce l'elaborazione dell'immagine.

9.2.3 Lambda Pipeline: Recupero e Analisi dell'Immagine

La **Lambda Pipeline** è incaricata di recuperare l'immagine precedentemente salvata su S3 e, se necessario, di accedere al database contenente gli embeddings.

A questo punto, la Lambda Pipeline può eseguire diverse operazioni, a seconda della richiesta iniziale:

- **Rekognition**: Se richiesto, la Pipeline Lambda interagisce con Amazon Rekognition per eseguire un'analisi dell'immagine, cercando la targhetta e fare il crop dell'immagine, in maniera tale da eliminare il rumore intorno
- **Textract/Bedrock**: A seconda del tipo di elaborazione richiesto, la Pipeline Lambda invia l'immagine o il testo ricavato da essa a **Textract** o **Bedrock**. Entrambi i servizi restituiscono un output in formato JSON contenente i risultati dell'elaborazione.

9.2.4 Aggiornamento del Database e Restituzione del Risultato

Il JSON restituito da Textract o Bedrock viene inviato nuovamente all'Lambda API. Questa, a sua volta, utilizza i dati ricevuti per aggiornare l'entry corrispondente nel database, assicurandosi che tutte le informazioni pertinenti siano correttamente memorizzate.

9.2.5 Risposta dell'API Gateway

A seconda dell'esito delle operazioni precedenti, l'API Gateway restituisce una risposta all'operatore:

- **Codice 500**: Viene restituito se tutte le operazioni sono state eseguite correttamente e senza errori.
- **Codice 200**: Viene restituito se si è verificato un errore in uno dei passaggi. In questo caso, l'Lambda API includerà nel messaggio di risposta anche il tipo di errore rilevato.

Parte IV

Analisi e Risultati

Introduzione della parte IV

Dopo aver delineato la struttura generale e implementato la base del progetto nella parte precedente, in questa parte ci concentriamo sull'analisi e sui risultati ottenuti. Partendo dalla configurazione stabilita, ci siamo posti l'obiettivo di ottimizzare i componenti chiave del sistema per raggiungere le prestazioni desiderate.

Attraverso una serie di esperimenti e variazioni nei modelli, ho esplorato diverse configurazioni, cercando di individuare quelle che offrono i migliori risultati. In particolare, l'attenzione è stata rivolta alla modifica e all'ottimizzazione dei componenti principali per valutare l'impatto di ciascuno sulla qualità complessiva del sistema.

Tutti i risultati ottenuti sono stati confrontati con il ground truth, utilizzando la distanza di Levenshtein per calcolare la precisione dei modelli. La distanza di Levenshtein, che misura il numero minimo di operazioni necessarie (inserzioni, cancellazioni o sostituzioni di caratteri) per trasformare una stringa in un'altra, è stata fondamentale per quantificare l'accuratezza delle previsioni dei modelli.Levenshtein [1966]. Oltre alla precisione, abbiamo preso in considerazione anche il tempo di esecuzione dei modelli, poiché la velocità è un fattore critico in molti scenari applicativi.

In questa fase, l'approccio metodologico ha previsto un'analisi approfondita delle prestazioni del sistema in differenti scenari, seguita dalla comparazione dei risultati ottenuti. L'obiettivo principale è stato quello di identificare la configurazione ottimale che consenta di massimizzare l'efficienza e l'efficacia del progetto, fornendo una base solida per le conclusioni finali.

Nel dettaglio, questo capitolo presenterà i dati raccolti, i criteri utilizzati per l'analisi, le modifiche apportate ai componenti e le conseguenti performance del sistema. Concluderemo con una discussione critica dei risultati ottenuti, mettendo in evidenza i trade-off e le scelte progettuali che si sono rivelate più influenti.

Capitolo 10

Scelta miglior modello Textract

Nell'analisi delle opzioni offerte da AWS Textract per l'estrazione di informazioni dai documenti, emerge che ciascuna modalità ha i suoi vantaggi specifici in base alla tipologia di dati e alla struttura del documento stesso. Tuttavia, tra le opzioni disponibili, l'utilizzo delle **Custom Queries** si distingue come la scelta più efficace per ottenere risultati altamente pertinenti e mirati. Di seguito, vengono esaminate le ragioni principali per cui questa opzione risulta superiore rispetto alle altre.

10.1 Analisi delle Opzioni di Estrazione

- **Estratto di Testo Semplice (Plain Text)**:
  - **Vantaggi**: Fornisce un'estrazione completa del testo presente nel documento senza considerare la struttura. È rapido e diretto, utile per acquisire tutto il testo disponibile.
  - **Limitazioni**: Questa opzione è limitata dalla sua mancanza di contestualizzazione e organizzazione. Non è adatta quando è necessario distinguere tra informazioni strutturate e non, poiché tratta tutto il testo come un blocco unico. Inoltre, come menzionato, offre risultati molto simili all'OCR di Rekognition, rendendola ridondante in questo contesto.
- **Estrazione di Tabelle**:
  - **Vantaggi**: Efficace per documenti che contengono informazioni organizzate in tabelle. Textract è in grado di identificare e strutturare automaticamente i dati tabulari, facilitandone l'interpretazione e l'uso successivo.
  - **Limitazioni**: Questa opzione è strettamente utile solo quando il documento contiene tabelle. In assenza di esse, la sua utilità è nulla, limitando notevolmente l'applicabilità della funzione in scenari più complessi o con dati distribuiti in maniera non tabellare.

10.2 Perché *Custom Queries* è l'Opzione Migliore

L'opzione **Custom Queries** emerge come la soluzione più versatile e potente per diversi motivi:

- **Flessibilità e Precisione**: Custom Queries permettono di definire esattamente quali informazioni estrarre, adattandosi a un'ampia varietà di documenti, indipendentemente dalla loro struttura. A differenza delle altre opzioni, non si è vincolati a un formato specifico (come tabelle o moduli), ma si può estrarre esattamente ciò che è necessario, sia che si tratti di una frase, un concetto o una specifica informazione.
- **Contestualizzazione dei Dati**: Con le Custom Queries, è possibile formulare ricerche che tengono conto del contesto e delle relazioni tra le informazioni. Questo permette di ottenere dati non solo accurati ma anche significativi, rispondendo a domande specifiche che potrebbero non essere immediatamente evidenti attraverso un semplice OCR o un'estrazione standard.
- **Applicabilità Versatile**: Questa opzione è particolarmente utile quando si lavora con documenti complessi, come contratti, fatture, rapporti o qualsiasi documento che non segue una struttura fissa. La capacità di Textract di rispondere a query personalizzate rende questa funzione estremamente adattabile e utile in scenari dove altre opzioni fallirebbero o richiederebbero ulteriori elaborazioni manuali.
- **Efficienza Operativa**: L'utilizzo delle Custom Queries riduce significativamente il tempo necessario per ottenere le informazioni desiderate, poiché evita di dover filtrare manualmente grandi quantità di testo o dati estratti in maniera non strutturata. Questo si traduce in una maggiore efficienza operativa, riducendo al contempo gli errori umani nell'interpretazione dei dati.

10.2.1 Tabelle di Confronto

Plain Text	Tabelle	Custom Queries
0.53	0.60	0.63

Tabella 10.1. Precisione Textract cambiando opzioni di utilizzo

10.3 Risultati finali

In sintesi, sebbene ogni opzione offerta da AWS Textract abbia i suoi vantaggi specifici, le **Custom Queries** rappresentano la scelta ottimale per l'estrazione di informazioni da documenti complessi e diversificati. La loro capacità di fornire risultati precisi e contestualizzati, adattandosi a un'ampia varietà di documenti e scenari, le rende uno strumento insostituibile per ottenere i migliori risultati nel progetto.

Capitolo 11

Scelta miglior modello Bedrock

11.1 Analisi delle Opzioni di Utilizzo di AWS Bedrock

Nell'analisi delle diverse modalità di utilizzo di AWS Bedrock per l'elaborazione dei dati, si evidenziano tre principali approcci che possono essere adottati in funzione delle specifiche esigenze del progetto: l'utilizzo del modello con *Prompt di Solo Testo*, con *Prompt di Testo e Immagini* e l'utilizzo del modello con *Embeddings*. Di seguito, viene presentata un'analisi approfondita di ciascuna opzione, con una discussione su quale sia la scelta più adatta per ottenere i migliori risultati nel contesto specifico del progetto.

11.1.1 Analisi delle Opzioni di Utilizzo

- **Utilizzo del Modello con Prompt di Solo Testo**:
  - **Vantaggi**: Questo approccio consiste nell'inserire direttamente il testo estratto tramite OCR in un modello generativo di AWS Bedrock. È semplice da implementare e particolarmente utile quando si dispone di testo già pulito e pronto per l'elaborazione. L'output generato è rapido e può essere sufficientemente preciso se il testo fornito è chiaro e ben strutturato.
  - **Limitazioni**: Questa opzione può risultare limitata quando il testo estratto non è sufficiente a fornire il contesto necessario al modello per generare una risposta precisa. In particolare, quando si trattano documenti complessi o quando il testo è estratto da immagini con informazioni visive significative, l'assenza di elementi visivi nel prompt può compromettere l'accuratezza e la rilevanza dei risultati.
- **Utilizzo del Modello con Prompt di Testo e Immagini**:

  - **Vantaggi**: L'utilizzo di un prompt che combina testo e immagini permette di fornire al modello un contesto più ricco e completo. Questa opzione è particolarmente potente quando si lavora con documenti dove l'informazione testuale è fortemente legata agli elementi visivi, come diagrammi, tabelle complesse o immagini che contengono testo. Il modello può elaborare congiuntamente testo e immagini, migliorando la precisione delle risposte generate.
  - **Limitazioni**: Sebbene offra un contesto più completo, questo approccio può richiedere più tempo e risorse computazionali, poiché il modello deve elaborare sia il testo che le immagini. Inoltre, la qualità dell'output può dipendere dalla risoluzione e dalla qualità delle immagini fornite, richiedendo potenzialmente una pre-elaborazione aggiuntiva delle immagini stesse.

- **Utilizzo del Modello con Embeddings**:
  - **Vantaggi**: L'approccio con embeddings consente di rappresentare il testo (e potenzialmente altri dati come immagini o metadati) in uno spazio vettoriale multidimensionale. Questo è estremamente utile per compiti che richiedono comprensione semantica profonda, come la ricerca di similarità, la classificazione o la clustering di documenti. Gli embeddings permettono al modello di catturare relazioni complesse tra concetti, migliorando l'accuratezza del recupero delle informazioni e l'elaborazione del testo.
  - **Limitazioni**: L'uso di embeddings richiede una fase preliminare di addestramento o selezione degli embeddings appropriati, il che può essere computazionalmente intensivo e complesso da configurare. Inoltre, l'interpretazione e l'utilizzo degli embeddings può risultare meno intuitiva rispetto a un semplice modello di generazione di testo, rendendo necessaria una conoscenza avanzata dei concetti sottostanti per sfruttarne appieno il potenziale.

11.1.2 Perché l'Utilizzo del Modello con *Embeddings* è l'Opzione Migliore

L'opzione **Utilizzo del Modello con Embeddings** emerge come la scelta ottimale per ottenere risultati più accurati e contestualizzati, per diverse ragioni:

- **Comprensione Semantica Profonda**: Gli embeddings consentono al modello di catturare relazioni semantiche complesse tra i concetti presenti nel testo. Questo è particolarmente vantaggioso in scenari dove è fondamentale comprendere il significato profondo delle informazioni e le connessioni tra esse, come in analisi di documenti tecnici o legali.

- **Ricerca e Similarità Migliorate**: Utilizzando embeddings, è possibile effettuare ricerche più precise e identificare similarità tra documenti o sezioni di testo che potrebbero non essere immediatamente evidenti con altre tecniche. Questo approccio è ideale per compiti di clustering, classificazione e recupero di informazioni basati su concetti piuttosto che su corrispondenze letterali.

- **Versatilità Applicativa**: Gli embeddings sono altamente versatili e possono essere applicati a vari tipi di dati, inclusi testo, immagini e metadati. Questa flessibilità rende l'opzione degli embeddings particolarmente utile in progetti complessi che richiedono un'elaborazione multidimensionale delle informazioni. Inoltre, una volta creati, gli embeddings possono essere riutilizzati in diverse applicazioni, migliorando l'efficienza complessiva del processo.

11.1.3 Tabelle di Confronto

Solo Testo	Testo e Immagini	Embeddings
0.63	0.67	0.72

Tabella 11.1. Precisione Bedrock cambiando opzioni di utilizzo

11.1.4 Conclusione

In sintesi, nonostante le diverse opzioni disponibili per l'utilizzo di AWS Bedrock, l'uso del **Modello con Embeddings** rappresenta la scelta più efficace e versatile per ottenere risultati ottimali. La capacità di catturare e rappresentare relazioni semantiche complesse all'interno dei dati permette di raggiungere un livello superiore di accuratezza e rilevanza nei risultati, rispondendo efficacemente alle esigenze specifiche del progetto.

Capitolo 12

Claude-3: Descrizione e Confronto tra le Varianti Haiku e Sonnet

12.1 Introduzione a Claude-3

Claude-3 è un modello avanzato di intelligenza artificiale sviluppato da Anthropic, progettato per elaborare e generare testi in modo estremamente sofisticato. Come i suoi predecessori, Claude-3 è stato addestrato su grandi quantità di dati testuali, rendendolo capace di comprendere e generare contenuti in linguaggi naturali con un alto grado di precisione. Questo modello è particolarmente utile in applicazioni che richiedono capacità di comprensione del linguaggio naturale, generazione di contenuti testuali, e interazioni basate su testo.

Claude-3 è disponibile in diverse varianti, ciascuna ottimizzata per specifici casi d'uso. In questo capitolo, ci concentriamo su due di queste varianti: **Claude-3 Haiku** e **Claude-3 Sonnet**, analizzandone le caratteristiche principali e confrontandone le performance nel contesto dell'integrazione con embeddings.

12.2 Confronto tra Haiku e Sonnet

Claude-3 Haiku e Claude-3 Sonnet sono due versioni del modello Claude-3, ciascuna progettata con obiettivi leggermente differenti:

12.2.1 Claude-3 Haiku

- **Descrizione**: Claude-3 Haiku è una versione leggera del modello, progettata per operazioni che richiedono una generazione di testo più veloce e meno complessa. Questo modello è caratterizzato da un numero inferiore di parametri rispetto a

Claude-3 Sonnet, rendendolo meno potente ma più efficiente in termini di risorse computazionali.

- **Vantaggi**: Grazie alla sua semplicità, Claude-3 Haiku è meno incline a generare risultati casuali o sovra-elaborati. Questo lo rende particolarmente adatto quando si lavora con embeddings, poiché il modello tende ad affidarsi maggiormente alle rappresentazioni semantiche fornite dagli embeddings piuttosto che tentare di produrre risposte complesse autonomamente.
- **Limitazioni**: Sebbene sia più efficiente, la minore complessità di Haiku può limitare la profondità e la ricchezza delle risposte generate, specialmente in contesti che richiedono una comprensione più sofisticata del testo.

12.2.2 Claude-3 Sonnet

- **Descrizione**: Claude-3 Sonnet è una versione più avanzata e potente del modello, con un numero significativamente maggiore di parametri rispetto a Haiku. Questo lo rende capace di generare testi più complessi e articolati, ideali per applicazioni che richiedono una generazione di contenuti di alta qualità.
- **Vantaggi**: Sonnet è progettato per produrre risposte più dettagliate e sofisticate, grazie alla sua capacità di elaborare contesti complessi. Questo lo rende particolarmente utile per scenari che richiedono un alto livello di comprensione e creatività.
- **Limitazioni**: Tuttavia, la sua potenza può essere un'arma a doppio taglio quando utilizzato con embeddings. La maggiore complessità del modello può portare a risultati casuali o sovra-elaborati, soprattutto se gli embeddings non forniscono un contesto sufficientemente chiaro. Questo rende Sonnet meno prevedibile e potenzialmente meno affidabile in contesti che richiedono precisione e coerenza.

12.3 Scelta di Claude-3 Haiku per l'Integrazione con Embeddings

Nonostante Claude-3 Sonnet offra una maggiore potenza e capacità di generazione, la scelta ricade stranamente su **Claude-3 Haiku** per l'integrazione con embeddings. Questa decisione è motivata da diversi fattori chiave:

- **Affidabilità con Embeddings**: Claude-3 Haiku, essendo un modello più semplice e meno potente, tende ad affidarsi maggiormente agli embeddings forniti, piuttosto che cercare di generare autonomamente contenuti complessi. Questo porta a risultati più prevedibili e coerenti, riducendo il rischio di generare output casuali o non pertinenti.
- **Riduzione della Complessità**: L'uso di Haiku riduce la complessità del sistema, consentendo una maggiore efficienza computazionale e facilitando l'integrazione del modello in applicazioni che richiedono risposte rapide e affidabili. In questo contesto, la semplicità diventa un vantaggio, poiché minimizza le possibilità di errore e massimizza la coerenza dei risultati.

- **Risultati Basati su Contesto**: Grazie alla sua predisposizione a seguire più fedelmente gli embeddings, Haiku si comporta meglio in applicazioni che richiedono una stretta aderenza al contesto fornito dai dati, rendendolo la scelta ideale per compiti che richiedono precisione semantica.

12.3.1 Tabelle di Confronto

Da questo momento in avanti, è stata introdotta una distinzione nelle tabelle: la prima riga presenta i valori medi calcolati su tutte le foto disponibili, mentre la seconda riga è stata generata prendendo in considerazione esclusivamente le foto di qualità superiore, prive di imperfezioni come luce inadeguata, targhette rovinate o altri difetti visivi.

	Haiku	Sonnet
TUTTE LE FOTO	0.77	0.72
SOLO LE MIGLIORI	0.90	0.85

Tabella 12.1. Precisione di Bedrock cambiando modello (Haiku e Sonnet)

	Haiku 1 Embedding	Haiku 2 Embeddings	Haiku 3 Embeddings
TUTTE LE FOTO	0.77	0.82	0.83
SOLO LE MIGLIORI	0.90	0.82	0.83

Tabella 12.2. Precisione di Haiku cambiando il numero di embeddings

12.3.2 Conclusione

In conclusione, sebbene Claude-3 Sonnet rappresenti una scelta potente per la generazione di testi complessi e articolati, la semplicità e l'affidabilità di **Claude-3 Haiku** lo rendono la scelta ottimale per l'integrazione con embeddings. La capacità di Haiku di generare risposte prevedibili e coerenti, senza deviare dal contesto fornito dagli embeddings, lo rende uno strumento più efficace e affidabile per il nostro progetto.

Capitolo 13

Scelta finale miglior modello

13.1 Introduzione

In questo capitolo, verrà discusso un confronto tra due approcci distinti per l'estrazione e l'elaborazione delle informazioni da immagini: l'uso di Textract con Custom Queries e l'impiego di un modello generativo Bedrock con embeddings, nello specifico Claude-3 Haiku. Questi due strumenti rappresentano soluzioni avanzate per l'estrazione di testo e dati strutturati da immagini, ma si basano su metodologie e tecnologie differenti, che portano a vari trade-off in termini di precisione, efficienza e applicabilità. Inoltre, verrà analizzato come entrambi i metodi possano essere ulteriormente supportati da Amazon Rekognition, un servizio che può essere utilizzato per pre-processare le immagini, eliminando il rumore non necessario e migliorando la qualità dell'input per l'analisi.

13.2 Textract con Custom Queries

Textract è un servizio di Amazon Web Services (AWS) progettato per estrarre testo, tabelle, e altri dati strutturati da documenti digitali e immagini. L'uso di Custom Queries permette di definire con precisione le informazioni specifiche che si desidera estrarre, migliorando così la rilevanza e l'accuratezza dei risultati.

Vantaggi:

- **Precisione nella ricerca di informazioni specifiche:** Le Custom Queries consentono di ottenere esattamente i dati richiesti, riducendo la quantità di informazioni superflue.
- **Integrazione diretta con AWS:** Essendo parte dell'ecosistema AWS, Textract offre un'integrazione fluida con altri servizi cloud.

Svantaggi:

- **Limitazioni nella flessibilità:** Sebbene le Custom Queries siano potenti, possono essere limitate se le informazioni necessarie non sono facilmente esprimibili attraverso query predefinite.

- **Dipendenza dalla qualità dell'immagine:** La precisione di Textract è strettamente legata alla qualità delle immagini fornite. Se l'immagine è rumorosa o mal formattata, i risultati potrebbero essere imprecisi, anche con l'ausilio di Rekognition.

13.3 Modello Generativo Bedrock con Embeddings (Claude-3 Haiku)

Claude-3 Haiku è un modello generativo avanzato disponibile tramite la piattaforma Bedrock, che utilizza embeddings per comprendere e generare testo in modo contestuale. In questo contesto, può essere impiegato per estrarre e interpretare informazioni da immagini convertite in testo, offrendo un approccio flessibile e adattabile.

Vantaggi:

- **Flessibilità ed adattabilità:** Claude-3 Haiku non è limitato da query predefinite, il che lo rende adatto per estrarre una vasta gamma di informazioni, anche non previste inizialmente.
- **Maggiore capacità interpretativa:** Grazie all'uso degli embeddings, il modello può comprendere il contesto e le sfumature del testo, fornendo risposte più accurate e pertinenti.

Svantaggi:

- **Rischio di sovra-interpretazione:** A causa della sua natura generativa, il modello potrebbe interpretare eccessivamente alcune informazioni, producendo risultati non sempre precisi.
- **Tempi di risposta variabili:** Sebbene generalmente efficiente, l'elaborazione tramite modelli generativi può richiedere più tempo rispetto a Textract, specialmente in scenari complessi.

13.4 Supporto di Rekognition

Amazon Rekognition può essere utilizzato in combinazione con entrambi gli approcci per migliorare la qualità dell'immagine prima dell'elaborazione. Questo servizio è in grado di rilevare e rimuovere rumori non necessari, come bordi inutili o elementi di distrazione, fornendo un'immagine più pulita e focalizzata. Tuttavia, l'efficacia di Rekognition dipende dal contesto specifico e dal tipo di immagini da analizzare.

Integrazione con Textract:

- Rekognition può migliorare i risultati di Textract riducendo il rumore visivo, che è particolarmente utile in documenti con strutture complesse.

Integrazione con Claude-3 Haiku:

- Anche per Claude-3 Haiku, Rekognition può essere utile per garantire che l'immagine sia adeguatamente preparata prima dell'analisi. Tuttavia, la natura più flessibile e contestuale di Claude-3 Haiku riduce la necessità di un pre-processing intensivo.

13.5 Tabelle di Confronto

	Textract	Claude	Textract + Rekognition	Claude + Rekognition
TUTTE LE FOTO	0.61	0.73	0.63	0.69
SOLO LE MIGLIORI	0.72	0.86	0.74	0.81

Tabella 13.1. Confronto precisone tra tutti i modelli

	Textract	Claude	Textract + Rekognition	Claude + Rekognition
TUTTE LE FOTO	5.64	6.87	6.28	7.57
SOLO LE MIGLIORI	5.49	6.72	6.01	7.07

Tabella 13.2. Confronto tempi di esecuzione tra tutti i modelli (in secondi)

Criterio	**Textract (Custom Queries) + Rekognition**	**Claude-3 Haiku**
Precisione	Alta, ma dipendente dalla qualità dell'immagine	Molto alta, grazie agli embeddings
Flessibilità	Limitata a query predefinite	Molto alta, adattabile a diversi contesti
Tempo di Risposta	Veloce, ma può rallentare con immagini complesse	Buono, ma variabile a seconda della complessità
Integrazione con Rekognition	Necessario per migliorare la qualità dell'immagine	Non sempre necessario, grazie alla capacità del modello di gestire il contesto
Scalabilità	Ottima in ambienti AWS	Buona, richiede risorse adeguate per prestazioni ottimali

Tabella 13.3. Confronto tra Textract con Custom Queries e Claude-3 Haiku.

13.6 Conclusione

Dopo aver analizzato i due approcci, emerge che l'uso di Claude-3 Haiku senza l'ausilio di Rekognition rappresenta la soluzione ottimale nella maggior parte dei casi. Questo approccio offre una maggiore precisione nell'interpretazione dei dati, grazie alla sua capacità di comprendere il contesto attraverso embeddings, e mantiene un buon bilanciamento tra precisione e tempi di risposta. Sebbene Textract con Custom Queries e Rekognition possa

essere preferibile in scenari altamente strutturati e specifici, Claude-3 Haiku si dimostra più versatile e efficace in contesti meno definiti.

Parte V

Conclusione

Capitolo 14

Conclusione

La tesi conferma che l'integrazione di tecnologie avanzate di Machine Learning e Robotica offre un significativo miglioramento nella gestione delle cabine elettriche. Dopo aver testato diverse opzioni, l'approccio che combina l'uso di AWS Textract con modelli generativi basati su embedding, come Claude-3 Haiku, si è rivelato il più efficace. Questo sistema permette una lettura automatizzata delle targhette elettriche con un livello di precisione che supera le tecniche tradizionali, riducendo al minimo gli errori umani e migliorando la velocità di esecuzione.

L'utilizzo delle Custom Queries di Textract è stato particolarmente utile per estrarre dati specifici da documenti complessi, ma è emerso che l'approccio generativo offre una flessibilità superiore, capace di adattarsi a diverse tipologie di documenti e condizioni operative. Le conclusioni principali indicano che l'adozione di modelli generativi, supportati da robusti servizi cloud come quelli offerti da AWS, può portare a una trasformazione radicale delle pratiche di manutenzione e gestione delle infrastrutture elettriche.

Oltre ai benefici operativi immediati, la tesi suggerisce che questo sistema può essere ulteriormente sviluppato e applicato ad altri contesti industriali, dove la gestione dei dati e la manutenzione preventiva sono critiche. Il lavoro future potrebbe esplorare l'integrazione di ulteriori modelli di intelligenza artificiale e la possibilità di adattare queste tecnologie a nuovi scenari operativi, migliorando ulteriormente l'efficacia, la sicurezza e la sostenibilità delle soluzioni implementate.

Sviluppi Futuri

Alla luce dei risultati ottenuti, ci sono diversi possibili sviluppi che potrebbero migliorare ulteriormente l'efficacia e la portata dell'integrazione tra Machine Learning, Robotica e gestione delle cabine elettriche.

Un possibile sviluppo riguarda la fusione tra i modelli utilizzati da AWS Textract e AWS Bedrock. Durante i test, è emerso che l'approccio basato su embedding, utilizzando modelli generativi come Claude-3 Haiku, è particolarmente efficace. Tuttavia, in alcuni casi, potrebbe non essere possibile ottenere un embedding sufficientemente simile per

permettere a Bedrock di funzionare in modo ottimale. In queste situazioni, un'integrazione più stretta tra Textract e Bedrock potrebbe offrire una soluzione, con il sistema che passa automaticamente da Bedrock a Textract quando l'embedding non è sufficientemente accurato. Questo approccio ibrido garantirebbe la massima flessibilità e affidabilità, adattandosi in tempo reale alle diverse condizioni operative e tipologie di documenti.

Inoltre, un altro sviluppo significativo potrebbe essere l'estensione di questo sistema oltre il settore energetico. La tecnologia sviluppata ha il potenziale per diventare un servizio universale, applicabile a molteplici contesti industriali in cui la gestione dei dati e la manutenzione preventiva sono critiche. Settori come quello manifatturiero, delle telecomunicazioni, dei trasporti e delle infrastrutture urbane potrebbero trarre vantaggio dall'implementazione di soluzioni simili, migliorando l'efficienza operativa e riducendo i costi di manutenzione.

Infine, un'ulteriore direzione di ricerca potrebbe esplorare l'integrazione di altri modelli di intelligenza artificiale e tecnologie robotiche avanzate per migliorare ulteriormente la sicurezza e la sostenibilità delle soluzioni proposte. Ad esempio, si potrebbe considerare l'uso di droni autonomi per ispezionare infrastrutture difficilmente accessibili oppure sostituendo l'operatore umano con il cane robot della Boston Dynamics, SPOT. Quest'ultima idea è stata facilitata dalla collaborazione tra Reply Sprint e Boston Dynamics, che ha permesso di sfruttare SPOT per scopi avanzati di automazione. SPOT, equipaggiato con telecamere ad alta risoluzione e sensori avanzati, potrebbe muoversi autonomamente all'interno delle cabine elettriche, identificare le targhette e acquisire immagini con una precisione superiore a quella umana, eliminando il rischio di errori o di esposizione a condizioni pericolose.

Questi sviluppi futuri rappresentano non solo un'evoluzione delle tecnologie esistenti, ma anche un'opportunità per rivoluzionare interi settori industriali, portando a una maggiore automazione, sicurezza e efficienza operativa.

Bibliografia

J. Barr. *Host your web site in the cloud: Amazon Web Services made easy*. SitePoint, 2013.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*, volume 10, pages 707–710, 1966.

Reply. Reply official website, a. URL https://www.reply.com/.

Sprint Reply. Sprint reply overview, b. URL https://www.reply.com/sprint-reply/it/.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 2021.

Bruno Siciliano and Oussama Khatib. *Springer Handbook of Robotics*. Springer, 2016.

R. Smith. An overview of the tesseract ocr engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition*, 2007.

Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.