

UNIVERSITÀ DEGLI STUDI DI SALERNO

Corso di Laurea Informatica
Fondamenti di intelligenza artificiale



SGARB-AI

Autori:

Francesco Faiella Mat. 0512116141

Giuseppe Gambardella Mat. 0512116369

Marco Brescia Mat. 0512116819

Github: <https://github.com/Marco-Brescia/SGARB-AI>

ANNO ACCADEMICO 2024/2025

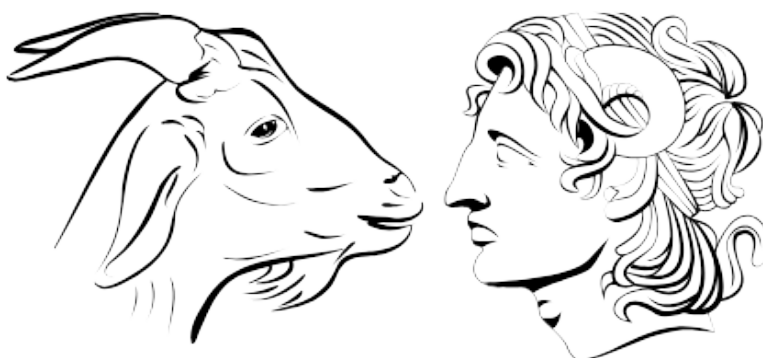
CONTENTS

1	Introduzione	3
1.1	Perché SGARB-AI?	3
1.2	Obiettivi del sistema	4
1.3	Struttura del documento	4
2	Analisi del problema	5
2.1	Specifica PEAS	5
2.1.1	Performance	5
2.1.2	Environment	5
2.1.3	Actuators	5
2.1.4	Sensors	5
2.2	Proprietà dell'ambiente	6
3	Dataset	7
3.1	Raccolta dei dati	7
3.2	Selezione dei dati	9
3.2.1	Processo di Selezione	9
3.2.2	Risultati e Vantaggi	10
3.3	Miglioramenti nel processo di Selezione	11
3.3.1	Modifica della misura usata per la selezione	11
3.3.2	Risultati e Vantaggi	11
4	Soluzione proposta	13
4.1	Soluzione proposta	13
4.2	Modello	13
4.2.1	Tecnologie del modello	13
4.2.2	Rete neurale	14
4.3	Dataset Training	15

4.3.1	Dataset Primo Training	15
4.3.2	Dataset Secondo Training	15
4.4	Strategia di addestramento	16
4.4.1	Funzione di ottimizzazione	16
4.4.2	Funzione di perdita	16
4.5	Risultati del Primo addestramento	17
4.6	Risultati del Secondo Addestramento	18
4.6.1	Descrizione dell'addestramento	18
4.6.2	Modifica della Funzione di Perdita	18
4.6.3	Analisi dei Risultati	18
4.7	Interfaccia con il modello	21
5	Miglioramenti e Sviluppi Futuri	22
5.1	Miglioramenti	22
5.1.1	Cross-validation	22
5.1.2	Ampliamento dei Dati di Training	23

CHAPTER 1

INTRODUZIONE



SGARB-AI

1.1 Perché SGARB-AI?

Ad oggi, gli algoritmi generativi sono capaci di produrre opere d'arte che possono essere indistinguibili da quelle create da artisti umani. Questo fenomeno ha aperto un dibattito etico e legale sulla natura dell'arte, sull'attribuzione di diritti d'autore e sul valore dell'autenticità. Il progetto SGARB-AI (System for Gauging Artistic Real and Bot-generated Artwork) nasce con l'obiettivo di esplorare queste tematiche, proponendo un modello capace di distinguere tra opere d'arte reali e quelle generate da algoritmi di intelligenza artificiale.

1.2 Obiettivi del sistema

Il sistema SGARB-AI mira a riconoscere e classificare opere d'arte in base alla loro origine, distinguendo tra immagini create da esseri umani e quelle generate da algoritmi di intelligenza artificiale. Il modello si basa sull'analisi di tratti distintivi per rilevare le differenze intrinseche tra opere autentiche e artificiali. Grazie al sistema, sarà possibile fornire uno strumento utile non solo per proteggere i diritti degli artisti e garantire la trasparenza nel mondo dell'arte digitale, ma anche per stimolare un dibattito consapevole sulle implicazioni etiche e culturali dell'uso dell'IA nell'arte.

1.3 Struttura del documento

Questo elaborato è organizzato per spiegare in modo chiaro e completo il progetto SGARB-AI. Nella prima parte viene presentato il contesto in cui il sistema è stato sviluppato, con un'analisi delle motivazioni e degli obiettivi principali, tra cui la necessità di distinguere le opere d'arte reali da quelle generate da algoritmi di intelligenza artificiale. Successivamente, si analizza il problema, evidenziandone le principali sfide e descrivendo l'ambiente in cui il sistema opera. Una sezione è dedicata alla raccolta e alla preparazione dei dati, descrivendo come il dataset è stato costruito e quali criteri sono stati adottati per garantire che fosse adeguato agli obiettivi del progetto. Il cuore del lavoro è rappresentato dalla descrizione della soluzione proposta, che include informazioni riguardanti le tecnologie e i modelli di machine learning utilizzati. Infine, il documento si chiude con un'analisi dei risultati ottenuti e una valutazione critica delle prestazioni del sistema, offrendo un quadro delle sue capacità.

CHAPTER 2

ANALISI DEL PROBLEMA

Questa sezione esamina il problema identificato, fornendone una formalizzazione e una descrizione attraverso il modello PEAS. Inoltre, vengono illustrate le caratteristiche principali dell’ambiente in cui l’agente opera.

2.1 Specifica PEAS

2.1.1 Performance

La misura di performance del sistema si basa sulla capacità dell’agente di distinguere correttamente tra immagini create da esseri umani e quelle generate da algoritmi di intelligenza artificiale, con particolare attenzione all’identificazione accurata della classe AI (falsi).

2.1.2 Environment

L’ambiente consiste nelle immagini digitali caricate dal sistema, che includono opere d’arte generate da AI o realizzate da esseri umani.

2.1.3 Actuators

Gli attuatori consistono in un sistema di classificazione che assegna un’etichetta (“UMANO” o “IA”) e un’interfaccia che presenta i risultati.

2.1.4 Sensors

I sensori consistono in un modulo di ricezione di immagini digitali da analizzare.

2.2 Proprietà dell'ambiente

- Completamente osservabile: L'agente riceve una rappresentazione completa dello stato dell'ambiente, poiché ogni input è costituito da un'immagine che viene analizzata per la classificazione.
- Stocastico: Lo stato futuro dell'ambiente non è prevedibile con certezza, in quanto l'immagine fornita come input varia ogni volta che l'utente carica un nuovo file. Questo introduce un elemento di incertezza, poiché il modello non sa in anticipo quale immagine verrà analizzata.
- Episodico: Ogni classificazione è indipendente dalle altre. Ogni immagine è un episodio isolato, e le decisioni prese su una non influenzano quelle successive. Non ci sono interazioni o dipendenze tra le immagini caricate.
- Dinamico: L'ambiente è dinamico in quanto ogni nuova immagine rappresenta una situazione unica, che può essere un'opera d'arte realizzata da un artista umano o un'opera generata da un algoritmo di intelligenza artificiale. L'ambiente cambia in base al tipo di immagine che viene caricata.
- Discreto: L'ambiente è discreto, nel senso che il sistema opera su un numero finito di pixel (l'immagine), e la classificazione delle opere è binaria (Umana o AI). Non ci sono azioni o percezioni continue in gioco.
- Singolo: Il sistema prevede l'interazione di un solo agente con l'ambiente. Non sono presenti altri agenti che influiscono sul processo di classificazione o sulle decisioni del sistema.

CHAPTER 3

DATASET

3.1 Raccolta dei dati

Dopo alcune ricerche si è scelto di utilizzare il seguente dataset:

- *AI-ArtBench*

Si è osservato che il dataset iniziale era suddiviso in più cartelle. Sono state raggruppate le immagini generate dall'intelligenza artificiale e quelle umane nelle cartelle rispettivamente chiamate **AI** e **REAL**.

Tuttavia, nonostante questa suddivisione, il dataset risultava molto sbilanciato, con circa 100.000 immagini nella classe **AI** e 50.000 nella classe **REAL**. Analizzando più a fondo la classe **AI**, sono state individuate due sottocategorie: **LD** e **SD**, che si riferiscono al tipo di algoritmo utilizzato per la generazione delle immagini:

- **LD**: Latent Diffusion (immagini di dimensioni 256x256 pixel)
- **SD**: Standard Diffusion (immagini di dimensioni 768x768 pixel)

Poiché il numero di immagini nelle due categorie era relativamente bilanciato (circa 50.000 immagini per ciascuna), si è deciso di utilizzare le immagini della categoria **LD**, in quanto presentano una dimensione inferiore rispetto alle immagini **SD**.

Inoltre, dato che il modello pre-addestrato che si intende utilizzare prende in input immagini di dimensioni 224x224 pixel, la perdita di dettaglio nelle immagini **LD** è minore rispetto a quella delle immagini **SD**. Con questa selezione, è stato possibile non solo ridurre la quantità delle immagini, ma anche riequilibrare il dataset, ottenendo così un dataset più bilanciato e adeguato all'addestramento del modello.

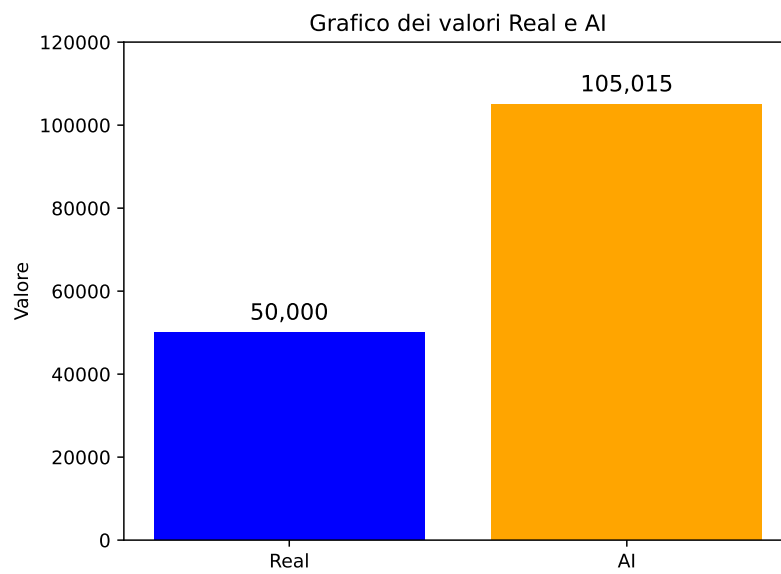


Figure 3.1: Grafico dei valori prima delle selezione

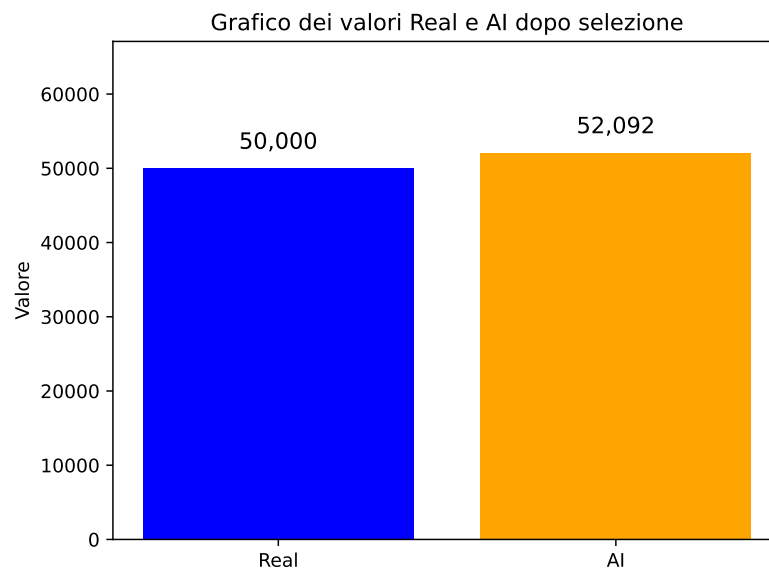


Figure 3.2: Grafico dei valori dopo la selezione

3.2 Selezione dei dati

Data la grande quantità di immagini disponibili in ogni cartella, è stato deciso di ridurre il numero a un massimo di 1000 immagini per cartella, per migliorare la qualità del dataset e semplificare il processo di addestramento. Per fare ciò, è stato implementato un processo basato sull'estrazione di feature e sull'analisi della varianza.

3.2.1 Processo di Selezione

Estrazione delle Feature con ResNet50

Per ciascuna immagine, è stato impiegato il modello **ResNet50**, pre-addestrato sul dataset **ImageNet**, per l'estrazione delle feature rappresentative. Le immagini sono state preprocessate mediante ridimensionamento a 224x224 pixel, seguite da una normalizzazione, e successivamente trasformate in vettori di feature attraverso il modello.

Riduzione della Dimensionalità con PCA

Per ridurre la complessità computazionale e focalizzarsi sui componenti principali delle feature, è stato applicato l'algoritmo PCA (Principal Component Analysis). Questo approccio ha permesso di ridurre dimensionalmente i vettori di feature mantenendo le informazioni più rilevanti.

Calcolo della Varianza

Dopo la riduzione dimensionale, è stata calcolata la varianza delle feature per ciascuna immagine. Questo parametro è stato utilizzato per determinare quali immagini includere in base alla loro categoria:

- Immagini REAL: sono state selezionate le immagini con maggiore varianza, poiché rappresentano una maggiore diversità visiva, utile per catturare la complessità della classe "REAL".
- Immagini AI: sono state selezionate le immagini con minore varianza, in quanto più omogenee e rappresentative della stabilità visiva tipica della classe "AI".

Selezione

Le immagini sono state ordinate in base alla varianza (in ordine decrescente per la categoria "REAL" e in ordine crescente per la categoria "AI") e sono state selezionate le prime 1000 per ogni cartella.

Salvataggio delle Immagini

Le immagini selezionate sono state salvate in una nuova struttura di directory organizzata per categoria "REAL" o "AI".

3.2.2 Risultati e Vantaggi

Questo approccio ha consentito di costruire un dataset ridotto. Grazie all'uso del modello ResNet50 per l'estrazione delle feature e alla successiva analisi della varianza, è stato possibile effettuare una selezione di immagini che migliorasse l'efficienza dell'addestramento.

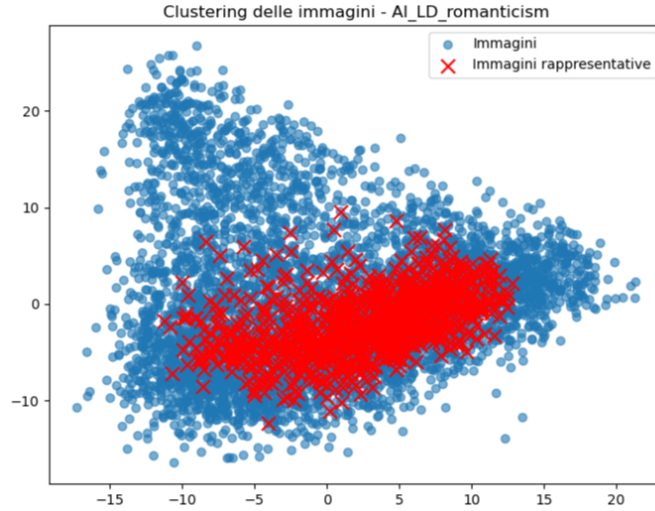


Figure 3.3: Grafico delle immagini selezionate per la classe AI

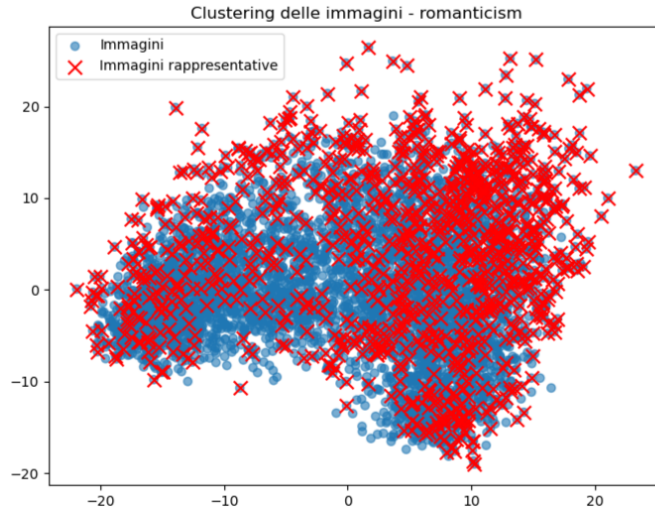


Figure 3.4: Grafico delle immagini selezionate per la classe REAL

3.3 Miglioramenti nel processo di Selezione

In questa sezione vengono descritti i miglioramenti implementati per ottenere una raccolta di immagini più robusta e completa, rispetto al metodo basato sull'utilizzo della varianza.

3.3.1 Modifica della misura usata per la selezione

Calcolo della Distanza Coseno

Dopo aver effettuato la riduzione dimensionale (vedi 3.2.1), è stata calcolata la distanza coseno ¹ di ciascuna immagine rispetto a un centroide, determinato come la media di tutte le feature estratte.

Questo approccio ha garantito una selezione ben distribuita nello spazio delle caratteristiche, permettendo di scegliere immagini distanti tra loro e mantenere così una maggiore diversità.

Selezione delle Immagini

Le immagini sono state ordinate in base alla distribuzione delle distanze coseno all'interno di ciascuna cartella. Per ogni categoria (**REAL** e **AI**), sono state selezionate 2000 immagini sparse e ben distribuite, ottenendo un totale di 40.000 immagini per il dataset di training (20.000 **AI** e 20.000 **REAL**).

3.3.2 Risultati e Vantaggi

Questo approccio ha consentito di ampliare il dataset con immagini uniformemente distribuite, migliorando significativamente la diversità dei dati utilizzati per l'addestramento del modello.

Grazie all'uso di ResNet50 per l'estrazione delle feature, PCA per la riduzione dimensionale e il calcolo della distanza coseno per la selezione, il dataset finale è caratterizzato da una distribuzione migliore, riducendo la ridondanza e garantendo immagini ben sparse, che ha portato all'addestramento di un modello che meglio rispecchia l'obiettivo del progetto.

¹La **distanza coseno** misura la dissimilarità tra due vettori in base all'angolo tra di loro. Indica quanto i vettori siano simili in direzione. In questo caso indica quanto le immagini sono simili (o dissimili) tra di loro.

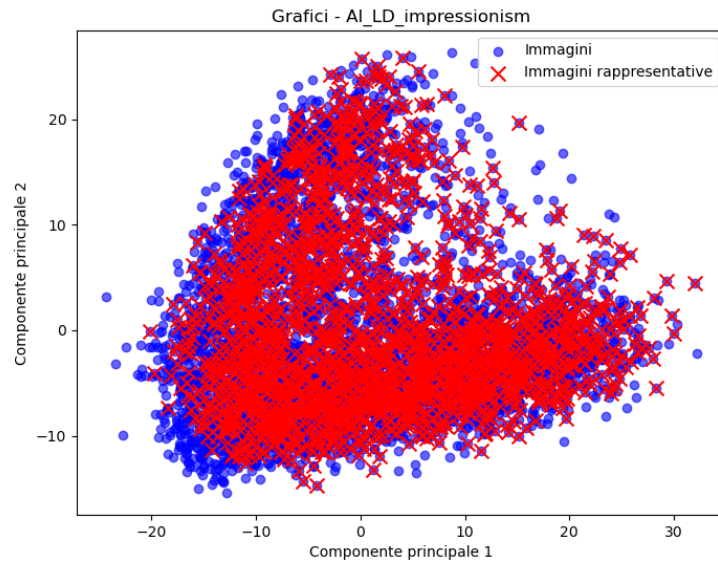


Figure 3.5: Distribuzione delle immagini selezionate per la classe AI.

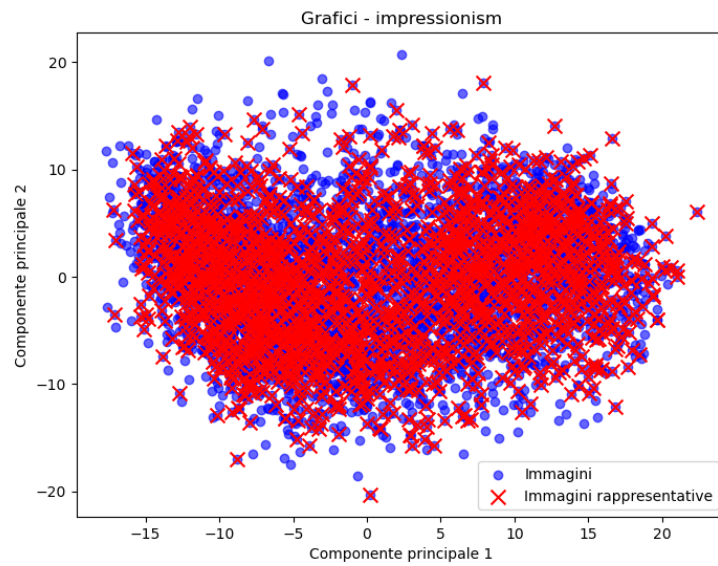


Figure 3.6: Distribuzione delle immagini selezionate per la classe REAL.

CHAPTER 4

SOLUZIONE PROPOSTA

4.1 Soluzione proposta

La soluzione proposta è un sistema di Machine Learning progettato per classificare immagini, rilevando la presenza di specifiche caratteristiche indicative di un particolare fenomeno. Nel nostro caso, il modello analizza immagini per identificare elementi di interesse.

4.2 Modello

In questa sezione, verrà descritta l'architettura del modello proposto, insieme alle sue principali caratteristiche e ai criteri che ne hanno guidato la selezione.

4.2.1 Tecnologie del modello

Il modello scelto si basa su una rete neurale convoluzionale (Convolutional Neural Network - CNN), che è particolarmente efficace per la classificazione di immagini. Le CNN operano attraverso una serie di trasformazioni sui dati di input, identificando progressivamente caratteristiche rilevanti come bordi, forme e strutture più complesse.

Il modello è stato implementato utilizzando Python 3.10 e alcune librerie:

- Keras/TensorFlow: per la costruzione, l'addestramento e la valutazione della rete neurale.
- NumPy: per operazioni numeriche ad alte prestazioni.
- Keras Tuner: per ottimizzare gli iperparametri della rete.

4.2.2 Rete neurale

La rete è composta da diversi componenti principali, con una struttura generale che include:

- ResNet50: Il modello utilizza ResNet50 pre-addestrato su ImageNet come base, sfruttando i suoi strati convoluzionali per estrarre feature rilevanti dalle immagini. I pesi di ResNet50 sono congelati durante il processo di addestramento, lasciando che il modello apprenda le caratteristiche attraverso i livelli aggiunti.
- Livelli Aggiuntivi:
 - Global Average Pooling: Dopo l'estrazione delle feature da ResNet50, un layer di pooling riduce la dimensione dei dati, mantenendo le informazioni rilevanti.
 - Dense Layer: Un layer completamente connesso con un numero di unità variabile (tra 512 e 2048), selezionato tramite tuning degli iperparametri.
 - Dropout Layer: Una regolarizzazione applicata per prevenire l'overfitting, con una probabilità di dropout compresa tra 0.2 e 0.5.
- Layer di output: Il layer di output ha una singola unità con funzione di attivazione sigmoid. Questa scelta è motivata dalla natura binaria del problema di classificazione, dove l'output rappresenta la probabilità che un'immagine appartenga alla classe positiva. Il risultato viene arrotondato a 0 o 1 per determinare la classe finale.

Il modello accetta immagini RGB come input, ciascuna con 3 canali (rosso, verde, blu).

Funzione di attivazione

La funzione di attivazione svolge un ruolo fondamentale nel modello, introducendo non linearità nei calcoli e consentendo alla rete di apprendere relazioni complesse nei dati. Nel modello implementato sono state utilizzate due funzioni di attivazione principali:

- ReLu: Viene utilizzata nei livelli convoluzionali della rete per introdurre non-linearità dopo ogni convoluzione, consentendo alla rete di apprendere caratteristiche più complesse.
- Sigmoid: Utilizzata nel layer di output per trasformare l'output del modello in una probabilità che rappresenta la classe predetta.

4.3 Dataset Training

4.3.1 Dataset Primo Training

Il processo di training del modello si è svolto utilizzando un dataset di 20.000 immagini suddiviso in tre insiemi: training, validazione e test, costruito attraverso la raccolta di immagini con la varianza (Vedi 3.2.1)

- Training set: 80% dei dati, utilizzati per addestrare il modello. (16.000 immagini)
- Validation set: 20% dei dati, utilizzati per valutare le prestazioni durante il training. (4.000 immagini)
- Test set: immagini indipendenti utilizzate per la valutazione finale del modello. (10.000 immagini)

4.3.2 Dataset Secondo Training

Il processo di training del modello si è svolto utilizzando un dataset di 40.000 suddiviso in tre insiemi: training, validazione e test, costruito attraverso la raccolta di immagini applicando la distanza coseno (Vedi 3.3.1)

- Training set: 80% dei dati, utilizzati per addestrare il modello. (32.000 immagini)
- Validation set: 20% dei dati, utilizzati per valutare le prestazioni durante il training. (8.000 immagini)
- Test set: immagini indipendenti utilizzate per la valutazione finale del modello. (10.000 immagini)

Tutte le immagini sono state ridimensionate a una risoluzione di 224×224 pixel e normalizzate per migliorare la stabilità numerica durante l'addestramento.

4.4 Strategia di addestramento

Il training del modello è stato eseguito con un massimo di 25 epoche. Per prevenire l'overfitting, è stato implementato un callback di Early Stopping, che monitora la perdita sul validation set. Il training si interrompe automaticamente se non vengono osservati miglioramenti per 5 epoche consecutive.

Gli iperparametri chiave (ad esempio, il numero di unità nei layer fully connected, il tasso di dropout e l'ottimizzatore) sono stati ottimizzati utilizzando Keras Tuner. Questa libreria ha consentito di selezionare automaticamente i valori più efficaci, massimizzando l'accuratezza sul validation set.

Le metriche monitorate durante il training includevano:

- Accuracy
- Specificity¹

4.4.1 Funzione di ottimizzazione

La funzione di ottimizzazione è un componente essenziale nell'addestramento di una rete neurale, poiché ha il compito di regolare i pesi del modello in modo da minimizzare la funzione di perdita.

L'ottimizzatore scelto è stato selezionato tramite un processo di tuning automatico, per identificare il miglior algoritmo che potesse adattarsi al meglio ai dati e al compito di classificazione. L'obiettivo è stato quello di migliorare le prestazioni generali del modello, garantendo una convergenza stabile e una buona generalizzazione sui dati di test e validazione.

4.4.2 Funzione di perdita

Nella soluzione proposta, è stata scelta una funzione di perdita binaria, adatta per un compito di classificazione tra due possibili categorie. Per migliorare la performance del modello, è stata applicata una ponderazione alla funzione di perdita, dando maggiore attenzione alla classe AI.

¹La specificità misura la capacità del modello di identificare correttamente i veri negativi, ovvero quanti esempi negativi sono stati correttamente classificati come tali. È calcolata come il rapporto tra veri negativi e la somma di veri negativi e falsi positivi.

4.5 Risultati del Primo addestramento

Dall'analisi dei grafici e, in particolare, della matrice di confusione Figure 4.2, emerge chiaramente che, grazie agli accorgimenti implementati per prioritizzare il riconoscimento della classe AI, quest'ultima viene riconosciuta meglio, come evidenziato dal test effettuato.

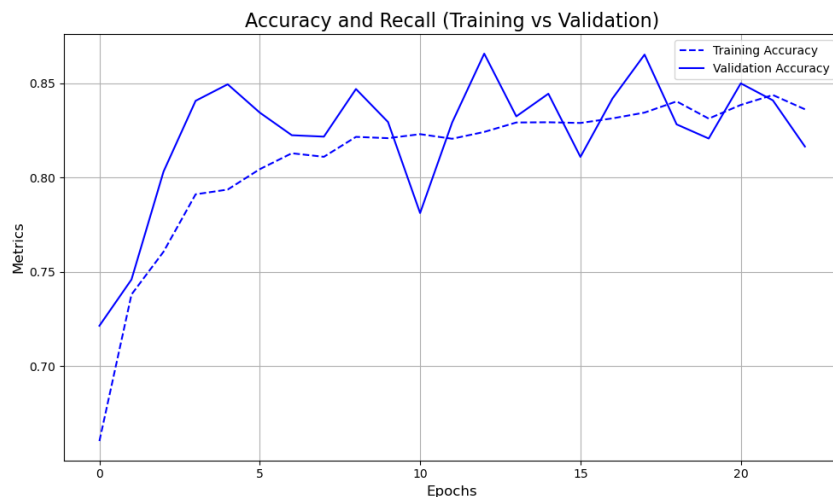


Figure 4.1: Accuracy durante il training e validazione

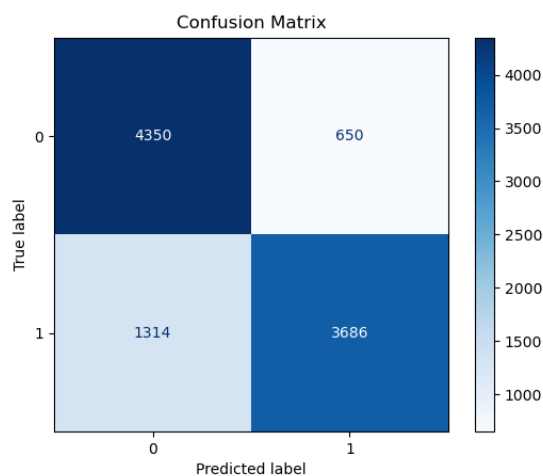


Figure 4.2: Matrice di confusione
Nota: Classe $AI(0)$, Classe $REAL(1)$

4.6 Risultati del Secondo Addestramento

4.6.1 Descrizione dell'addestramento

Il secondo addestramento è stato effettuato utilizzando un numero di immagini doppio rispetto al primo esperimento, passando da 1000 a 2000 immagini per cartella selezionate in base alla distanza coseno (section 3.3). Questo aumento ha permesso al modello di disporre di un set di dati più ricco e variegato, utile per migliorare la generalizzazione.

4.6.2 Modifica della Funzione di Perdita

La funzione di perdita è stata opportunamente modificata al fine di penalizzare in maniera più incisiva gli errori relativi alla classificazione della classe **AI**. L'obiettivo principale è stato quello di migliorare il riconoscimento della classe **AI**, mantenendo comunque un bilanciamento accettabile tra le metriche di **Accuracy** e **Recall**, per non compromettere eccessivamente il riconoscimento della classe **REAL**.

4.6.3 Analisi dei Risultati

Dai grafici e dalla matrice di confusione (Figure 4.6), si osserva che, seppur di poco, è migliorata la capacità del modello di riconoscere le opere **AI** rispetto alle opere **REAL**. Inoltre, grazie all'addestramento su un numero doppio di dati, il modello è diventato più robusto e riesce a riconoscere meglio le opere generate da intelligenza artificiale che appaiono più complesse e realistiche, rispetto al modello precedente.



Figure 4.3: Confronto tra Training e Validation Loss

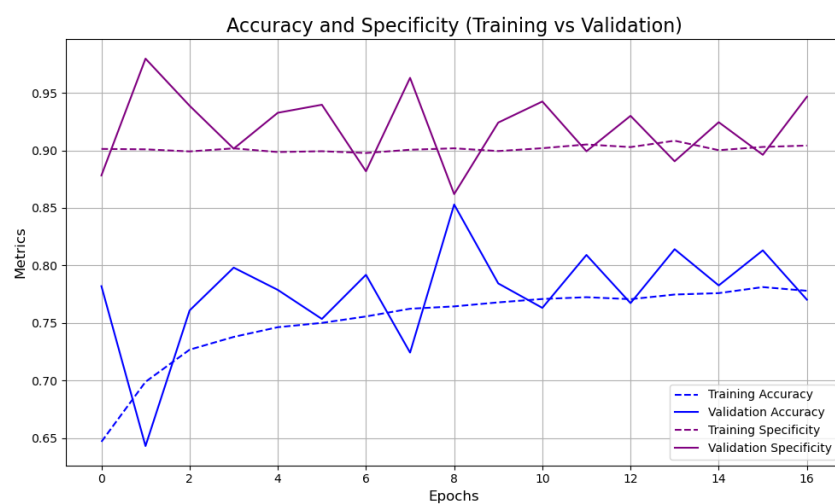


Figure 4.4: Andamento di Accuracy e Recall durante Training e Validazione

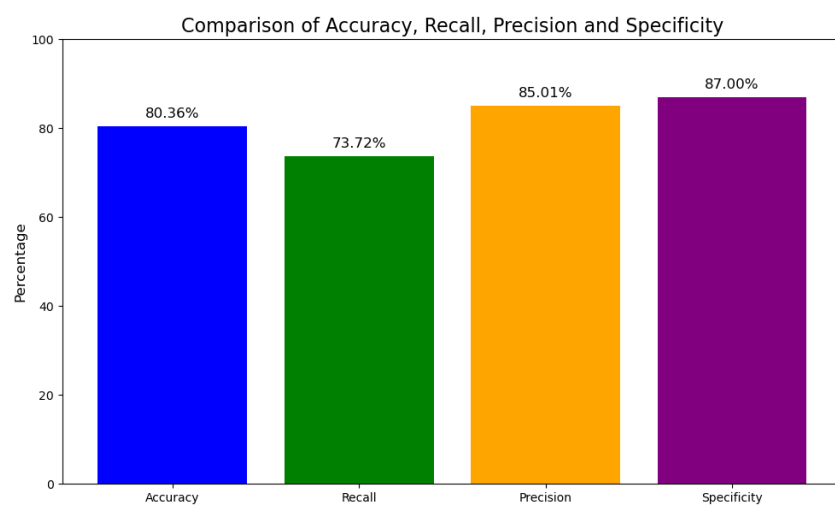


Figure 4.5: Andamento di Accuracy e Recall durante Training e Validazione

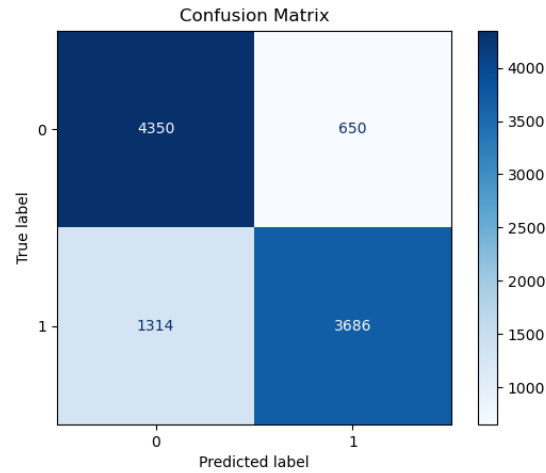


Figure 4.6: Matrice di Confusione
 Nota: Classe *AI*(0), Classe *REAL*(1)

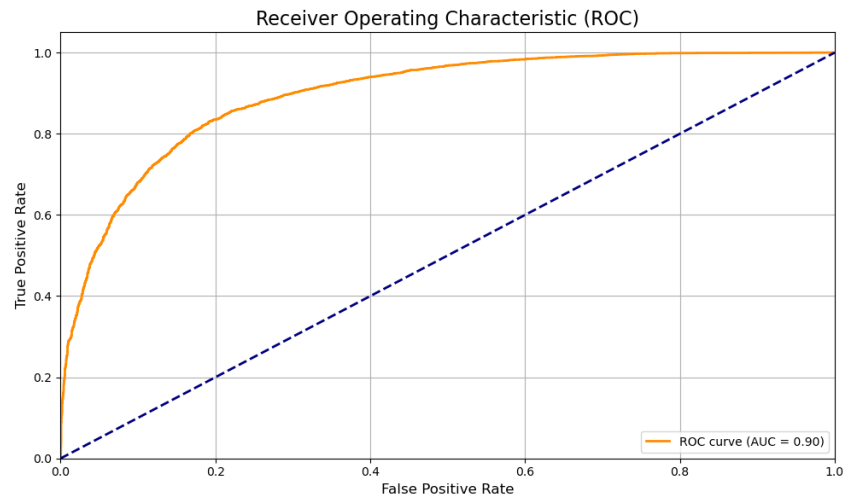


Figure 4.7: Matrice di Confusione

Questo risultato è stato ottenuto accettando un compromesso nella capacità di riconoscimento della classe *REAL*, in linea con gli obiettivi del progetto.

4.7 Interfaccia con il modello

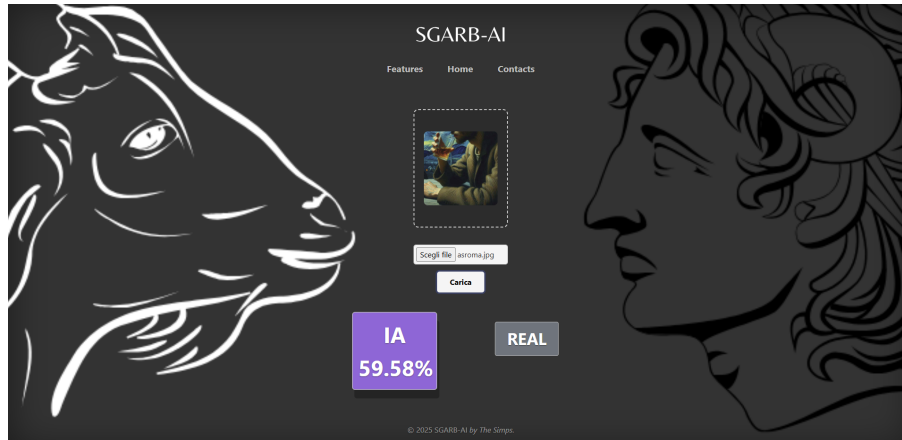


Figure 4.8: Sito web SGARB-AI

Il sito web di SGARB-AI è stato sviluppato per offrire un'interfaccia per consentire l'utilizzo del modello di intelligenza artificiale sviluppato. Attraverso il sito, l'utente può caricare un'immagine da analizzare, accedendo così alle funzionalità del modello senza necessità di interagire direttamente con il codice o con strumenti tecnici complessi.

Una volta caricata l'immagine, il modello di classificazione elabora il contenuto visivo e determina se si tratta di un'immagine generata artificialmente o reale. Il risultato della classificazione viene quindi presentato in modo chiaro e immediato attraverso l'interfaccia grafica, che utilizza un sistema di blocchi dinamici per evidenziare il risultato.

CHAPTER 5

MIGLIORAMENTI E SVILUPPI FUTURI

5.1 Miglioramenti

In questa sezione, si descrivono i potenziali miglioramenti che possono aumentare l'efficacia del modello.

5.1.1 Cross-validation

Nell'approccio attuale viene utilizzata una semplice cross-validation per valutare il modello. Un miglioramento significativo consisterebbe nell'adozione della **k-fold cross-validation**, una tecnica più robusta che consente di ottenere una valutazione più completa delle prestazioni del modello.

Con la *k-fold cross-validation*, il dataset viene suddiviso in k parti uguali (*fold*). In ogni iterazione, un fold viene utilizzato per la validazione, mentre i rimanenti $k - 1$ fold vengono impiegati per l'addestramento. Questo processo viene ripetuto k volte, garantendo che ogni porzione del dataset venga utilizzata sia per l'addestramento sia per la validazione.

L'implementazione della *k-fold cross-validation* permetterebbe di valutare il modello su diverse porzioni di dati, migliorando la capacità di generalizzazione. Inoltre, questa tecnica renderebbe l'analisi delle prestazioni più accurata e stabile.

5.1.2 Ampliamento dei Dati di Training

L'ampliamento del dataset di training può rappresentare un elemento cruciale per migliorare la capacità del modello di distinguere tra immagini reali e generate dall'IA, specialmente in scenari dove il dataset è sbilanciato o limitato. Per il problema specifico, è possibile considerare i seguenti approcci:

- **Espansione del Dataset con Fonti Esterne:** Identificare e integrare nuove immagini provenienti da fonti diverse, ad esempio dataset open-source, che contengano immagini reali o generate da altre tecnologie di IA. Questo consentirebbe di aumentare la varietà dei dati e migliorare la capacità del modello di generalizzare.
- **Filtraggio dei Dati:** Verificare la qualità dei dati esistenti per assicurarsi che siano privi di artefatti o distorsioni che potrebbero influenzare negativamente l'addestramento.
- **Sbilanciamento Controllato:** Includere deliberatamente un numero maggiore di immagini della classe "AI" rispetto alla classe "REAL" durante l'addestramento, data la necessità di minimizzare gli errori di classificazione di immagini AI come reali. Questa strategia può migliorare la sensibilità del modello per la classe critica.
- **Aggiornamento Dataset:** Sostituire o integrare il dataset attuale con immagini più recenti, in quanto le tecnologie di IA evolvono rapidamente e le immagini generate di recente potrebbero differire significativamente rispetto a quelle di un anno fa. Questo aggiornamento garantirebbe che il modello sia addestrato su dati rappresentativi e aggiornati.

Questi approcci consentono di arricchire il dataset e di migliorare l'efficacia del modello nel discriminare accuratamente tra le due classi, aumentando la robustezza e la capacità di generalizzazione.