**TIPO TRACCIA:** riconoscimento immagini

**NOME GRUPPO: segnIA**

**COMPONENTI:** Aldo, Martina, Maurizio, Nicola

**DATA:** 17/09/2024

**DESCRIZIONE:** l’obiettivo è quello di riconoscere il linguaggio dei segni;

**PASSI DA SEGUIRE:**

1. Studio librerie e metodi: si prevede l’utilizzo di Pandas, numpy, Keras, TensorFlow, matplotlib, Scikit-Learn
2. Trovare i dataset necessari:

<https://www.kaggle.com/datasets/datamunge/sign-language-mnist>

1. Trovare repository funzionanti da studiare in modo approfondito:

<https://github.com/MaSTERmIKK/AulaMLandAI/blob/main/Day2/riconoscimento/Riconoscimento.py>

<https://github.com/MaSTERmIKK/riconoscimento_immagini/blob/main/modello_updated.ipynb>

<https://www.kaggle.com/code/sayakdasgupta/sign-language-classification-cnn-99-40-accuracy>

1. Pre-processing dei dati
2. Scelta del modello da addestrare
3. Addestramento
4. Test
5. Implementazione
6. Lavorare sulla presentazione dei risultati ottenuti

**INPUT RICHIESTI:** per l’addestramento usiamo un dataset di immagini codificate in file .csv; per l’utilizzo finale cercheremo di dare come input immagini vere e proprie e codificarle in file.csv per effettuare le predictions

**OUTPUT ATTESO:** riconoscimento del segno dato in input

**NOTE AGGIUNTIVE:** durante le fasi di lavoro ci proponiamo di impegnarci a tenere traccia dei progressi attraverso una documentazione dettagliata

### 

### 

### 

### **Fase 1: Pre-processing e Gestione dei Dati**

1. Ricerca e Raccolta dei Dati
2. Pre-elaborazione dei Dati (Dataset dei Pixel)

* Normalizzare i pixel, ridimensionare e preparare i dati per l'input al modello

.**Fase 2: Sviluppo del Modello**

1. Studio del modello da addestrare

* Consultare documentazioni relative ai modelli da utilizzare

1. Prototipo del modello CNN

* Implementare e addestrare il modello CNN di base

1. Ottimizzazione del modello

* Migliorare l'architettura del modello (ad esempio, aggiungendo più layer o ottimizzando l'ipertuning dei parametri) per aumentare l'accuratezza e ridurre l'overfitting

### **Fase 3: Testing e Validazione**

1. Valutazione del Modello sui Dati Sintetici

* Testare il modello sui dati di test sintetici (dataset dei pixel) e creare rapporti di performance (matrice di confusione, accuratezza, precisione, richiamo).

1. Trasformazione delle Immagini Reali e Testing

* Trasformare le immagini reali nel formato compatibile con il modello basato sui pixel e testare l'efficacia del modello su queste immagini.

1. Report e Analisi degli Errori

* Analizzare gli errori commessi dal modello e identificare le aree di miglioramento (ad esempio, immagini mal classificate, errori comuni).

### **Fase 4: Sperimentazioni Aggiuntive**

1. Data Augmentation

* Implementare tecniche di data augmentation (rotazioni, variazioni di luminosità, scaling, ecc.) per migliorare la robustezza del modello e gestire meglio le immagini reali.

### **Fase 5: Applicazioni e Output**

10. Riconoscimento in Tempo Reale: passare un'immagine reale o un video in tempo reale per il riconoscimento del linguaggio dei segni

11. Preparazione del Report Finale e Presentazione: redigere un rapporto finale che includa il processo, i risultati, le sfide affrontate e i possibili sviluppi futuri. Preparare una presentazione per mostrare i risultati.