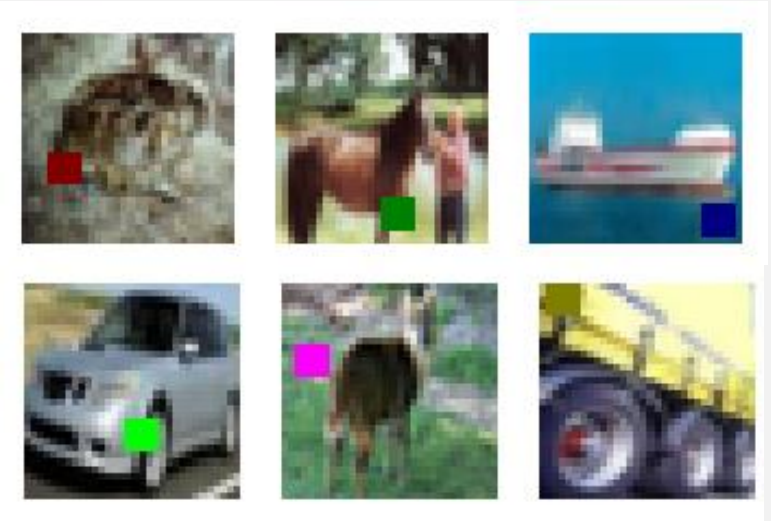
HOMEWORK 3

Benedetta Nassi - 10000059819

IL DATASET:

Training set:



Per un MLP, ogni pixel dell'immagine è un input **indipendente**. Un'immagine RGB viene appiattita in un vettore di valori numerici, ciascuno dei quali rappresenta l'intensità di un canale di colore per quel pixel specifico. L'MLP impara delle relazioni tra questi singoli valori numerici e le classi di output.

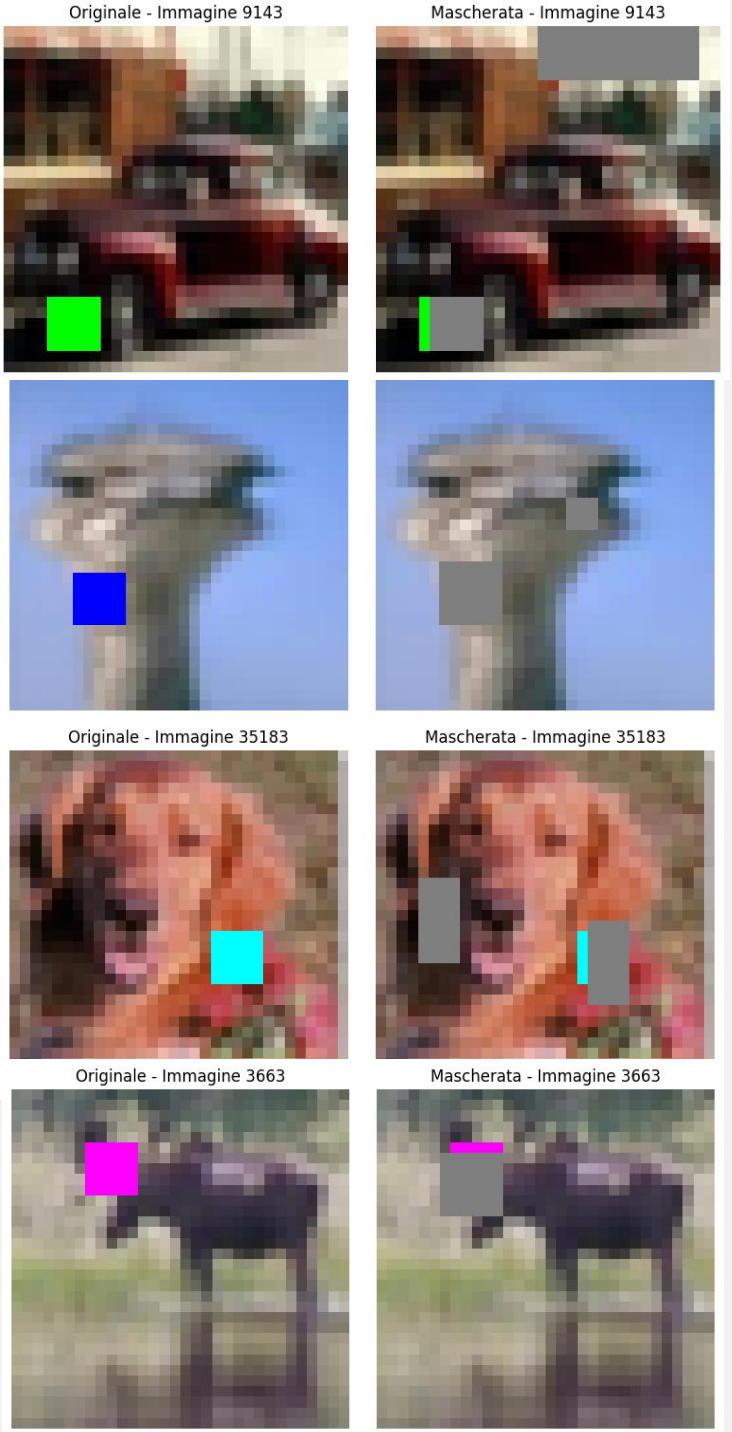
Validation set:

Test set:

Le immagini del training set contengono un quadrato 5x5 di colore casuale posizionato in modo casuale.

PROBLEMA DI *SHORTCUT LEARNING*

L'MLP può facilmente rilevare correlazioni tra i valori RGB di quei pixel e le etichette delle classi, imparando a sfruttare questa "***scorciatoia***" a discapito della comprensione del contenuto effettivo dell'immagine, evitando di apprendere le feature **significative.**



SOLUZIONE

Ho creato la funzione ***maschera_quadrati*** che analizza ogni immagine dividendola in piccole patch 5x5 . Per ogni patch, calcola la sua media e deviazione standard dei valori RGB. Queste statistiche vengono poi confrontate con la media e la deviazione standard dell'intera immagine.

Rilevamento dei quadrati:
Se la media del colore di una patch o la sua deviazione standard differisce in modo significativo dalla media o variabilità dell'intera immagine, quella patch viene considerata un'anomalia (cioè il nostro quadrato).

Mascheramento con il grigio:
Le patch anomale vengono mascherate, cioè i loro pixel vengono impostati su un colore **grigio** (valore di 0.5 per i canali RGB)

Il grigio perchè è il colore più neutro, quindi non apporta informazioni di colore specifiche che potrebbero, a loro volta, diventare nuove "scorciatoie" per l'MLP.

TECNICHE CONTRO L'OVERFIT:

REGOLARIZZAZIONE L2: riduce i parametri verso valori piccoli e uniformi. Impedisce quindi ad una singola feature di influenzare troppo la previsione, **riducendo** così **l'overfitting** e migliorando la capacità del modello di generalizzare a nuovi dati.

PCA: concentrandosi sulle componenti più significative, la PCA può filtrare efficacemente il rumore presente nei dati originali dei pixel.

BAGGING: crea più versioni di MLP addestrandoli su diversi campioni bootstrap dei dati di addestramento. Mediando le previsioni di più modelli, il bagging porta spesso a migliori prestazioni di **generalizzazione** su dati non visti.

EARLYSTOPPING: monitora le prestazioni del modello durante l'addestramento e interrompe il processo di addestramento quando le prestazioni iniziano a degradare (indicando overfitting) quindi **trova il punto ottimale per la generalizzazione**.

MODEL SELECTION: GRID-SEARCH

`hidden_layer_sizes`

definisce l'**architettura della rete neurale**, ovvero quanti hidden-layer e quanti neuroni sono presenti nell'MLP

- **troppo elevato:** può portare a una rete **eccessivamente complessa** capace di memorizzare il rumore del set di addestramento, perdendo la capacità di generalizzare su nuove immagini.
- **troppo basso:** rende la rete **sottodimensionata** (non ha abbastanza capacità computazionale).

`alpha`

controlla l'intensità della **regolarizzazione L2**

- **troppo alto:** penalizza eccessivamente i pesi, limitando la capacità di apprendimento della rete e potendo portare a **underfitting.**
- **troppo basso:** equivale poca o nessuna regolarizzazione il che rende il modello più incline all'**overfitting.**

`learning_rate_init`

determina la **dimensione dei passi** che la rete compie durante l'ottimizzazione per aggiornare i pesi. Influenza la velocità e la stabilità del processo di ottimizzazione dei parametri

- **troppo alto:** può far sì che l'ottimizzazione "salti" oltre il punto ottimale, rischiando di oscillare e non convergere.
- **troppo basso:** rende il processo di apprendimento molto lento, rischiando che questo si blocchi in un minimo locale.

RISULTATI

```
Migliori parametri trovati: {'alpha': 0.1, 'hidden_layer_sizes': (100, 50, 10), 'learning_rate_init': 0.01}
```

Al seguito della mia soluzione sono riportati i risultati ottenuti, il modello classifica con un'accuracy del 52% circa

```
Test Accuracy: 0.5199
F1 Macro: 0.5145257194999784
```