UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN INFORMATICA

INSEGNAMENTO DI **COMPUTER VISION**

DeXpression: Modello architetturale CNN per il riconoscimento delle espressioni facciali

DOCENTE

Prof. Francesco ISGRÒ

AUTORI

N97/347 Gennaro SORRENTINO

N97/393 Gianluca L'ARCO

ANNO ACCADEMICO 2022 - 23

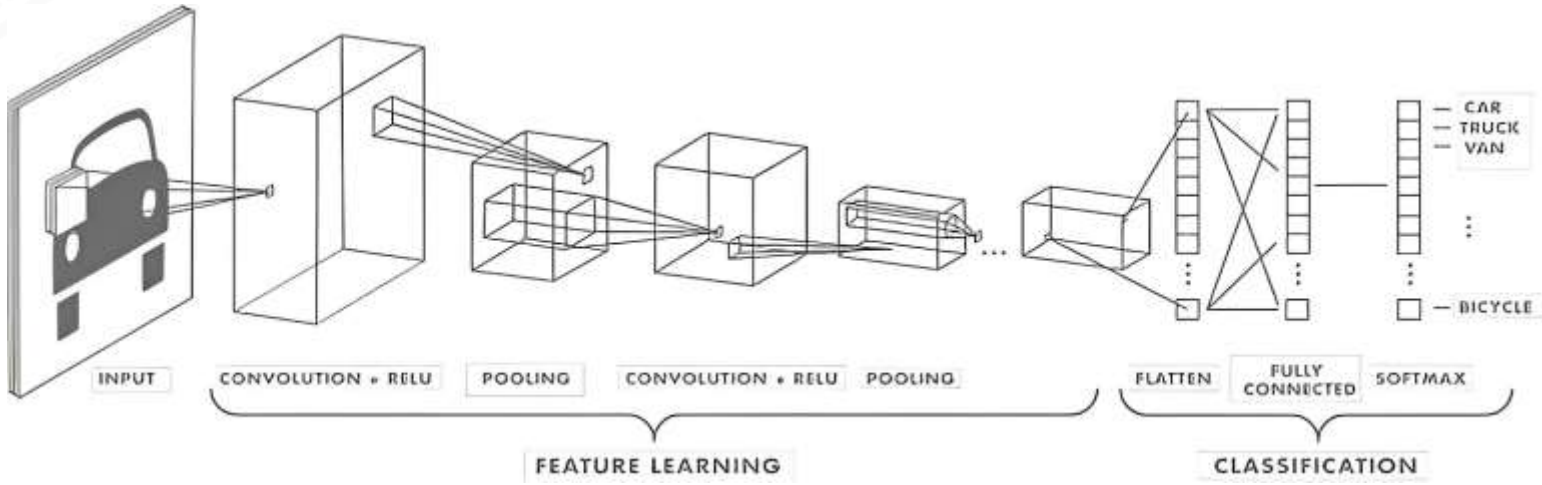
UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

Lo Stato dell'Arte

CNN: Convolutional Neural Network

DEFINIZIONE – Convolutional Neural Network

Una *Convolutional Neural Network* (CNN) è un tipo di rete neurale artificiale progettata specificamente per il trattamento di dati strutturati, come immagini e video. La sua architettura è ispirata dall'organizzazione della corteccia visiva animale.



CNN: Architettura

L'architettura di base di una CNN è costituita da diversi strati, ognuno dei quali ha una funzione specifica nell'elaborazione dei dati di input:

ELENCO – Classi di layer di una CNN

- Layer Convoluzionale
- Layer di Pooling
- Layer Fully Connected

I layer convoluzionali estraggono le caratteristiche dell'input, i layer di pooling manipolano le dimensioni mentre il layer completamente connesso si occupa della generazione dell'output finale.

CNN: Layer Convoluzionale

DEFINIZIONE – Layer Convoluzionale

Un *layer convoluzionale*, o strato convoluzionale, è un componente fondamentale di una rete neurale convoluzionale (CNN). Questo strato applica l'operazione di convoluzione agli input ricevuti per estrarre le caratteristiche rilevanti dai dati.

La convoluzione è un'operazione tra due funzioni di una variabile che consiste nell'integrare il prodotto tra la prima e la seconda traslata di un certo valore.

Siano $f(t), g(t): \mathbb{R} \to \mathbb{R} \Rightarrow$ la convoluzione di f e g è:

$$(f * g)(t) \coloneqq \int_{-\infty}^{\infty} f(t-\tau)g(\tau)\,d\tau$$

CNN: Layer Convoluzionale

Rappresentiamo un'immagine in scala di grigi I_G come una matrice A di pixel $M \times N$ con $A_{m,n} \in (0,1)$, supponendo che l'immagine sia stata precedentemente sottoposta ad un processo di normalizzazione.

Un layer convoluzionale è composto da uno o più filtri comunemente detti *kernel*. Un kernel è una matrice di pesi $W \in \mathbb{R}^{k \times k}$.

DEFINIZIONE – Operazione di convoluzione

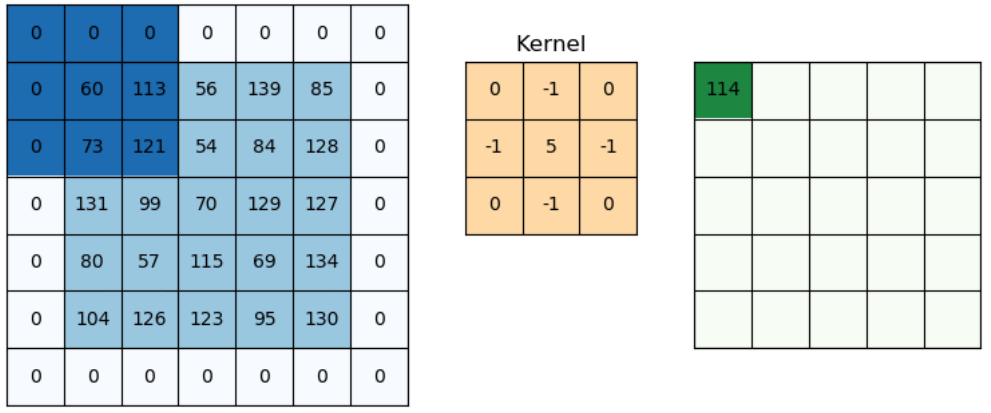
Nell'operazione di convoluzione, il kernel viene scansionato sull'input in modo da coprire tutte le possibili posizioni. A ciascuna posizione, viene eseguita un'operazione di prodotto tra gli elementi dell'input e i corrispondenti pesi del kernel. Il risultato di questa operazione viene sommato per ottenere l'elemento corrispondente nell'output convoluzionale.

CNN: Layer Convoluzionale

Matematicamente, l'operazione di convoluzione tra il kernel W e l'input A può essere espressa come:

$$O_{i,j} = \sum_{m=1}^{k} \sum_{n=1}^{k} w_{m,n} \cdot A_{(i-1)k+h,(j-1)k+n} + b$$

Di seguito un esempio illustrativo della convoluzione:



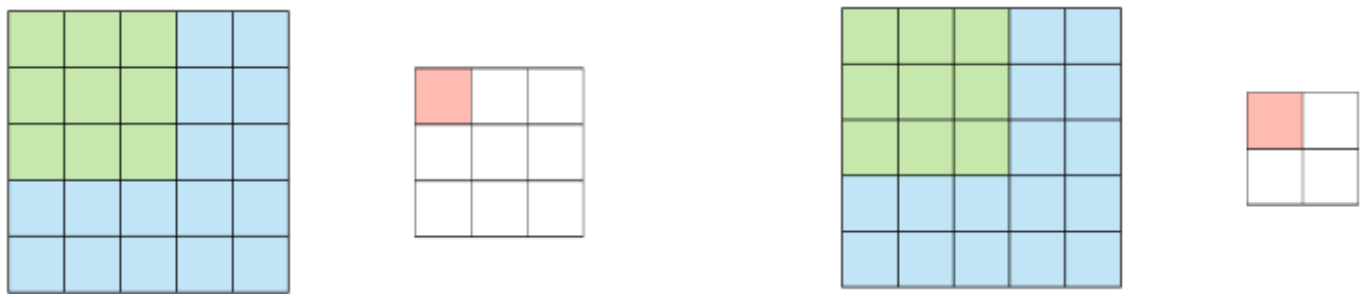
CNN: Layer Convoluzionale

Lo spostamento del kernel sull'input prende il nome di *stride*.

DEFINIZIONE – Stride

Lo *stride*, nella contesto di una rete neurale convoluzionale (CNN), è un parametro che definisce lo spostamento della finestra di convoluzione durante l'operazione di convoluzione. Indica di quanti passi il kernel si muove lungo l'input dopo ogni convoluzione.

Di seguito un esempio di stride pari a 1 e 2 (*rispettivamente sinistra e destra*):

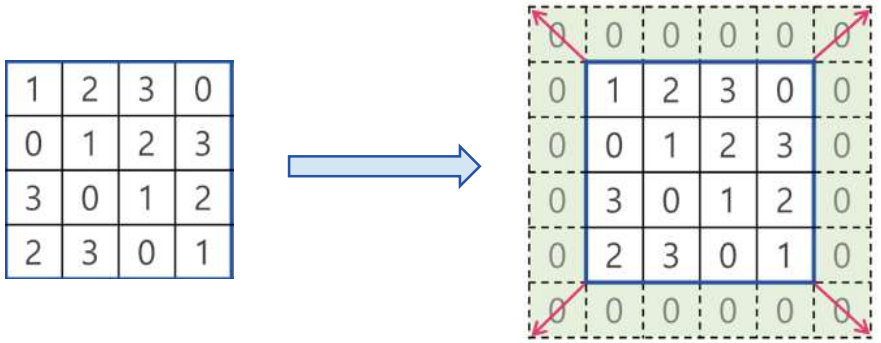


CNN: Layer Convoluzionale

Data la natura dell'operazione, la dimensione dell'output della convoluzione sarà generalmente più piccolo della dimensione dell'input. Per tale motivo si introduce una tecnica aggiuntiva detta *padding*.

DEFINIZIONE – Padding

Il *padding*, nell'ambito delle reti neurali convoluzionali (CNN), è una tecnica che consiste nell'aggiungere zeri (o valori costanti) intorno ai bordi dell'input prima di eseguire l'operazione di convoluzione. Questa tecnica viene utilizzata per preservare le dimensioni dell'input originale e controllare la dimensione dell'output convoluzionale.



CNN: Layer Convoluzionale

Un layer convoluzionale può elaborare immagini a tre canali RGB e quindi matrici tridimensionali (generalmente può elaborare qualunque matrice n-dimensionale).

Supponendo che I_C sia un'immagine a colori RGB, la matrice tridimensionale associata è A di dimensione $M \times N \times 3$ con $M \times N$ dimensione di I_C, l'operazione di convoluzione non cambia, tuttavia, è necessario che i vari kernel abbiano tutti la stessa profondità dell'immagine, ossia $W \in \mathbb{R}^{k \times k \times 3}$.

NOTA – Volumi di Feature Map

Si può notare che applicando l'operazione di convoluzione su ciascun kernel, si ottiene una sovrapposizione degli output che forma un volume di feature map. Pertanto, anche se viene fornita in input alla rete un'immagine bidimensionale, è molto probabile (basta applicare più di un kernel) avere durante il processo di calcolo un volume di feature map.

CNN: Layer Convoluzionale

ESEMPIO – Calcolo convoluzionale

- **Input:** Immagine di dimensione $32 \times 32 \times 1$ (scala di grigi)
- **Convoluzione 1 [5 kernel $3 \times 3 \times 1$ – padding = 1, stride = 1]:** Otteniamo in questo modo 5 feature map di dimensione $32 \times 32 \times 1$. Sovrapponiamo le feature map e otteniamo un volume di feature di dimensione $32 \times 32 \times 5$.
- **Convoluzione 2 [7 kernel $3 \times 3 \times 5$ – padding = 0, stride = 1]:** Otteniamo in questo modo 7 feature map di dimensione $30 \times 30 \times 1$. Sovrapponiamo le feature map e otteniamo un volume di feature di dimensione $30 \times 30 \times 7$.
- **Convoluzione 3 [1 kernel $3 \times 3 \times 7$ – padding = 1, stride = 1]:** Otteniamo in questo modo una sola feature map di dimensione $30 \times 30 \times 1$.

CNN: Layer di Pooling

DEFINIZIONE – Layer di Pooling

Un *layer di pooling* è una componente chiave nella rete neurale convoluzionale (CNN) poiché permette di ridurre la dimensione spaziale dei dati di input tramite operazioni di aggregazione locali, contribuendo a semplificare il modello e migliorare la capacità di generalizzazione.

Il pooling è un processo che prevede la divisione della mappa delle caratteristiche (feature map) in regioni spaziali chiamate pool. Su ciascuna di queste regioni viene applicata un'operazione di aggregazione al fine di ottenere un unico valore rappresentativo.

ELENCO – Operazioni di Pooling

- Max pooling
- Average pooling

CNN: Layer di Pooling

DEFINIZIONE – Max pooling

Il *Max Pooling* suddivide la mappa delle caratteristiche in una griglia di regioni non sovrapposte e restituisce il valore massimo presente in ciascuna regione.

Questa operazione consente di preservare le caratteristiche più rilevanti dell'input, riducendo al contempo le dimensioni complessive. Essa si basa sulla seguente formula:

$$MaxPooling(X)_{i,j,k} = \max_{m,n} X_{i \cdot s+m, j \cdot s+n, k}$$

dove X rappresenta l'input, i e j sono gli indici dell'output, k è l'indice del canale e s è lo stride.

CNN: Layer di Pooling

DEFINIZIONE – Average pooling

L'*Average Pooling* suddivide la mappa delle caratteristiche in una griglia di regioni non sovrapposte e calcola la media dei valori presenti in ciascuna regione.

Questa operazione è preferibile nei casi in cui è necessario smussare i dati di input in quanto aiuta a identificare la presenza di valori anomali. Esso si basa sulla seguente formula:

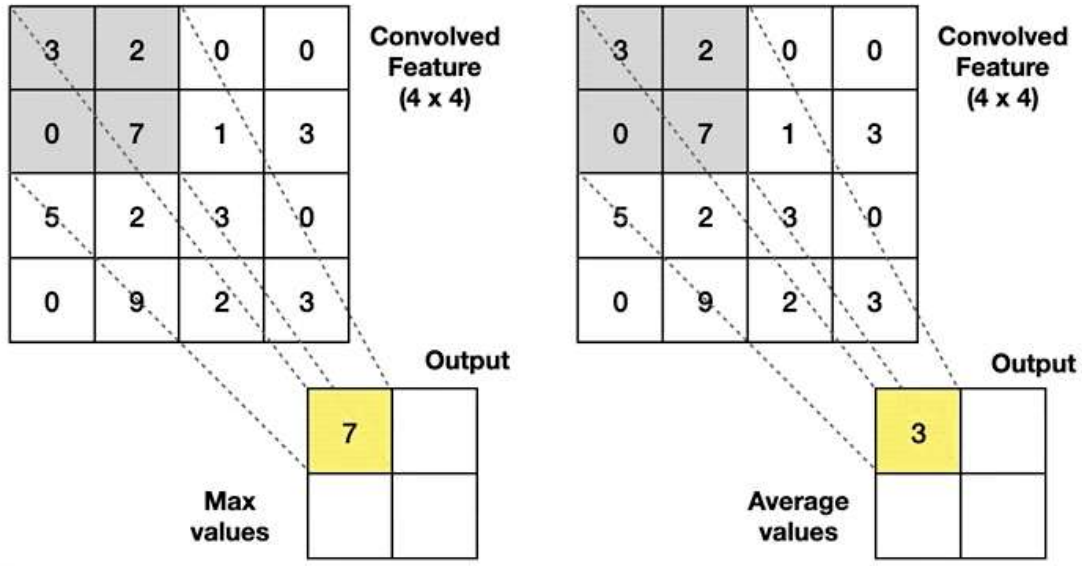
$$AvgPooling(X)_{i,j,k} = \frac{1}{K} \sum_{m,n} X_{i \cdot s+m, j \cdot s+n, k}$$

dove X rappresenta l'input, i e j sono gli indici dell'output, k è l'indice del canale, s è lo stride e K è la dimensione del kernel.

CNN: Layer di Pooling

ESEMPIO – Max & Average Pooling

Di seguito un esempio illustrativo di Max Pooling e Average Pooling (*rispettivamente sinistra e destra*):



CNN: Layer Fully Connected

DEFINIZIONE – Layer fully connected

Un *layer fully connected* è un tipo di layer in cui ogni neurone è connesso a tutti i neuroni del layer precedente, consentendo alla rete neurale di apprendere relazioni complesse tra le feature di input.

Un layer fully connected ha il seguente comportamento:

$$z = f_h\left(\sum_{j}^{m_{h-1}} w_{ij}^{h} \cdot z_j^{h-1} + b_i^h\right)$$

Dove f_h è la funzione di attivazione relativa al layer h, z_j^{h-1} è l'output del neurone j-esimo dello strato precedente, w_{ij}^h è il peso della connessione dal nodo i-esimo dello strato h al nodo j-esimo dello strato $h-1$ e infine b_i^h è il bias nel neurone i-esimo appartenente allo strato h.

CNN: Apprendimento

La fase di apprendimento consta di diverse fasi, supponendo di trattare un problema di classificazione:

ELENCO – Fasi di Apprendimento

- **Forward Propagation:** Calcolo dell'etichetta $\bar{y}_i$ dell' input I
- **Calcolo dell'errore:** Calcolo dell'errore di classificazione
- **Back Propagation:** Retro-propagazione dell'errore per l'aggiornamento dei pesi

L'errore della rete può essere calcolato in diversi modi, un esempio è l'errore quadratico medio MSE:

$$E_{MSE} = \frac{1}{2}\sum_{i}(y_i - \bar{y}_i)^2$$

Dove i rappresenta l'i-esimo input, y_i la sua etichetta e $\bar{y}_i$ l'etichetta ipotizzata dalla rete.

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

DeXpression

Architettura del Modello

DeXpression è un modello di CNN progettato per riconoscere le espressioni facciali umane, analizzando le caratteristiche del volto umano e identificando l'espressione emotiva corrispondente, come felicità, tristezza, rabbia, sorpresa, disgusto, paura o neutralità.

L’architettura del modello può essere suddivisa in 4 blocchi principali:

ELENCO – Architettura DeXpression

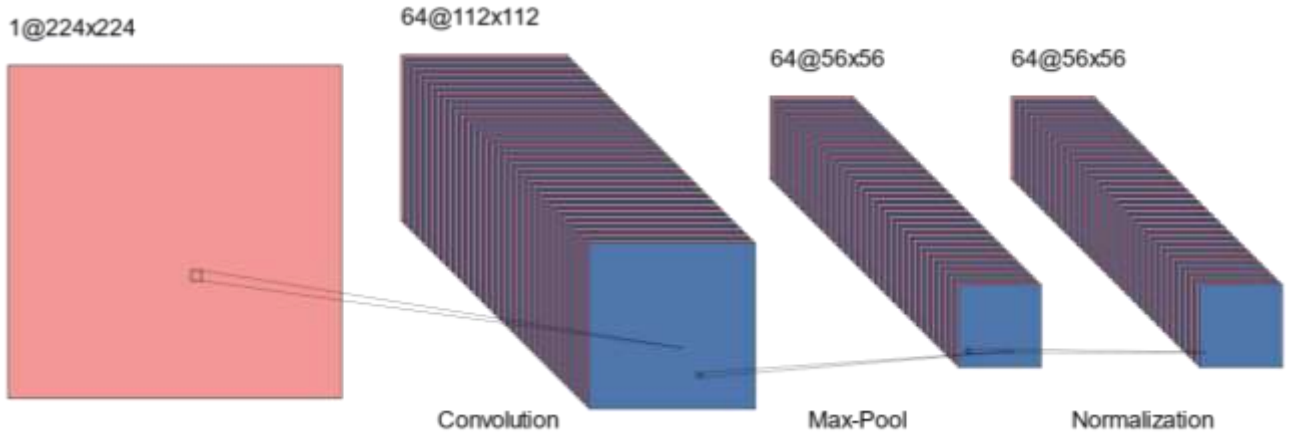
- **Blocco PPB**
- **Blocco FeatEx A**
- **Blocco FeatEx B**
- **Classificatore**

Architettura del Modello: Blocco PPB

Il blocco PPB (Pre-processing Block) si occupa di estrarre le informazioni più significative dall'immagine ricevuta in input e di prepararle per i blocchi successivi, ossia, i blocchi FeatEx.

ELENCO – Descrizione PPB

- Applicazione di *n* filtri per estrarre feature di basso livello dalle immagini.
- Ridimensionamento delle immagini, mantenendo le feature salienti estratte precedentemente.
- Applicazione di una layer di normalizzazione



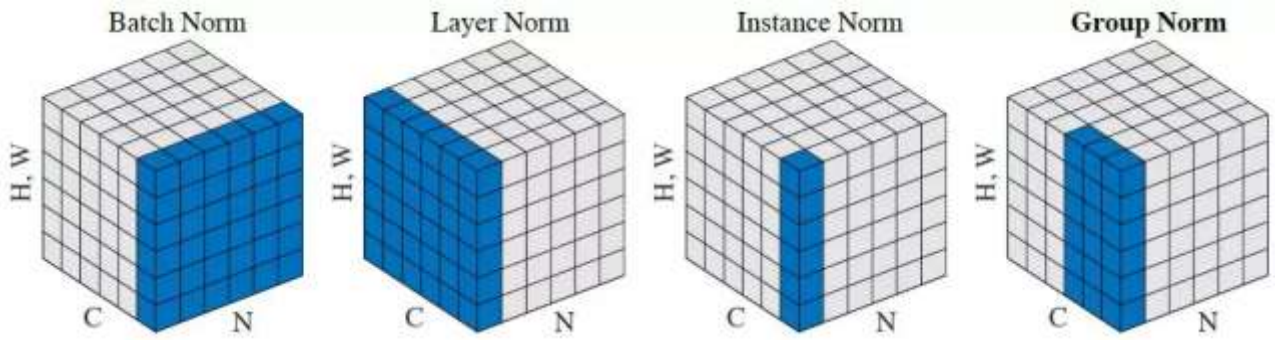
Architettura del Modello: Blocco PPB

DEFINIZIONE – Layer di Normalizzazione

La normalizzazione dell'input attraverso un layer di normalizzazione può essere espressa matematicamente come segue:

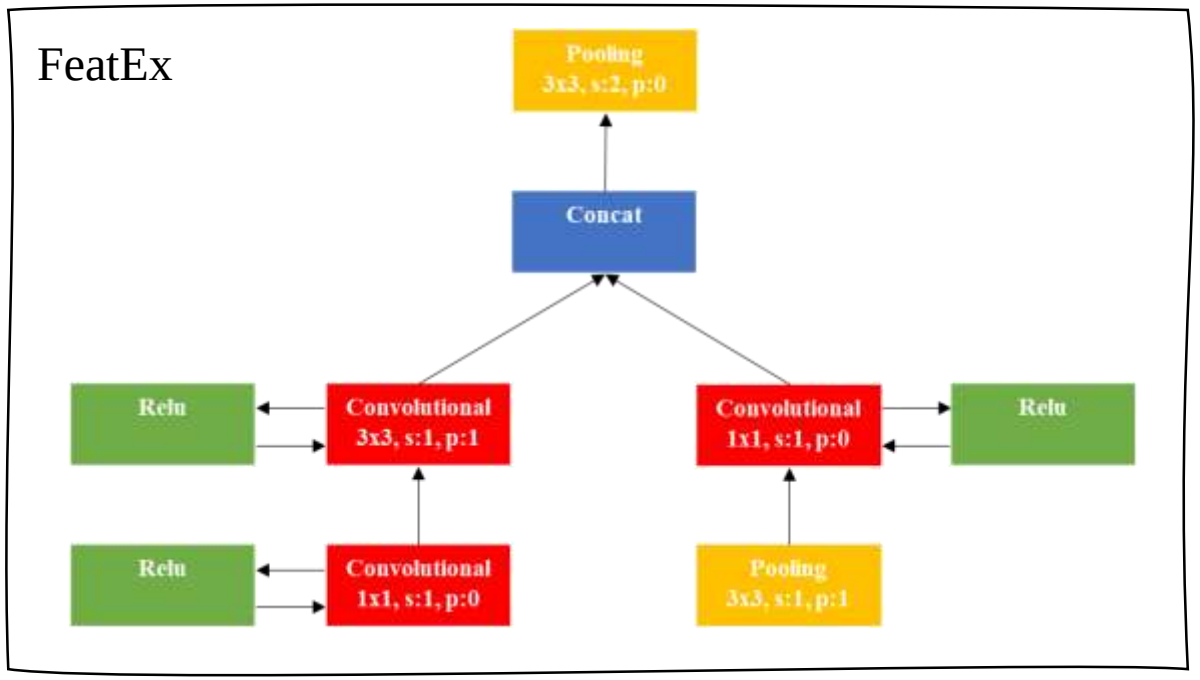
$$y = \frac{x - \mu}{\sqrt{\sigma + \varepsilon}} \cdot \gamma + \beta$$

dove γ e β sono parametri addestrabili, μ è la media, σ la deviazione standard e ε una costante



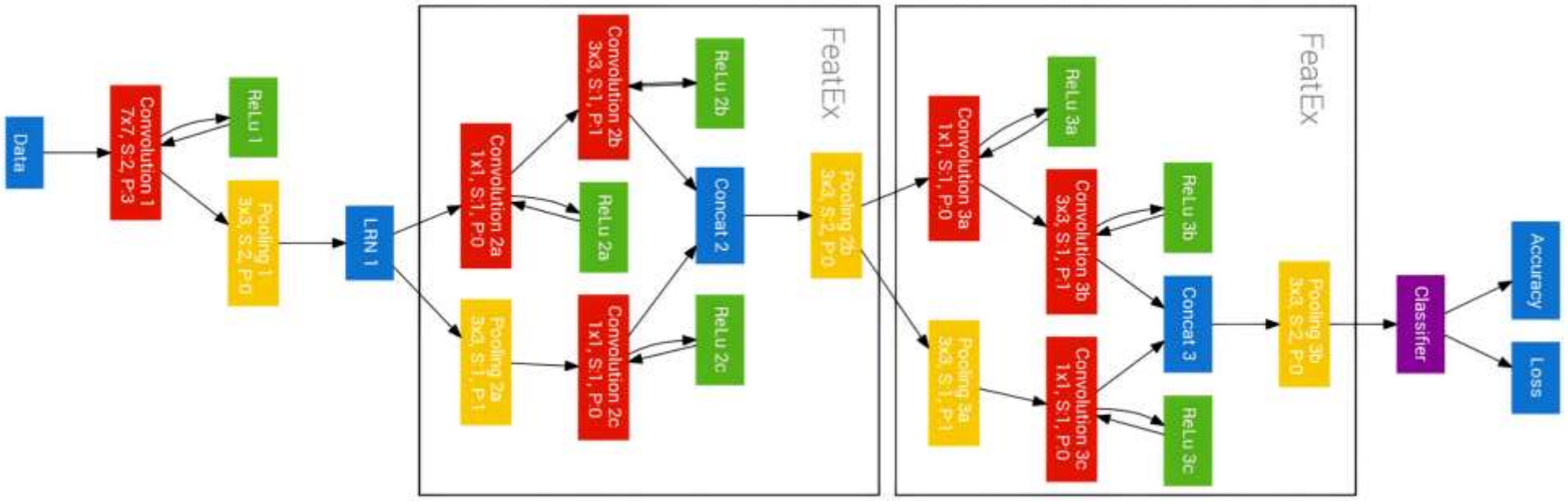
Architettura del Modello: Blocco FeatEx A|B

Il blocco FeatEx (Parallel Feature Extractor) costituisce il cuore del modello DeXpression. Esso permette la creazione di due percorsi paralleli con dimensioni differenti, al fine di concatenare i risultati e ottenere una rappresentazione dell'input più variegata, che si traduce in una gamma più ampia di informazioni e diverse prospettive dell'input.



Architettura del Modello: Classificatore

Infine, l'output del blocco FeatEx B viene alimentato a un classificatore denso, che restituisce un vettore di dimensione C (dove C rappresenta il numero di espressioni riconosciute) contenente le probabilità di ciascuna espressione.



Dataset: CK+ (Extended Cohn-Kanade)

Il dataset CK+ (Cohn-Kanade Plus) è un popolare set di dati utilizzato nella ricerca sulla rilevazione ed espressione delle emozioni. È una versione estesa del dataset originale CK e contiene immagini di volti umani che rappresentano una vasta gamma di espressioni facciali. Le caratteristiche del dataset sono:

ELENCO – Caratteristiche del dataset CK+

- Le espressioni rappresentate sono 7: **rabbia, disgusto, paura, felicità, tristezza, sorpresa e disprezzo**
- Le immagini hanno risoluzione di 640×480 o 640×490, sono estratte da sequenze video e sono sia a colori che in scala di grigi
- Nel dataset hanno contribuito 210 individui distinti, con un'età compresa tra i 18 e i 50 anni, di entrambi i sessi. L'81% dei partecipanti è di origine Euro-Americana, mentre il restante 18% è di origine Afro-Americana.

Dataset: CK+ (Extended Cohn-Kanade)

Esempio – CK+

Di seguito un esempio delle immagini contenute nel dataset CK+:

Dataset: MMI Facial Expression

Il dataset MMI è anch'esso, come CK+, un popolare set di dati utilizzato nella ricerca sulla rilevazione ed espressione delle emozioni. Le caratteristiche del dataset sono:

ELENCO – Caratteristiche del dataset MMI

- Le espressioni rappresentate sono 6: **rabbia**, **disgusto**, **paura**, **felicità**, **tristezza** e **sorpresa**
- Il dataset è composto da 2900 sessioni video in cui ogni partecipante esprime alcune tra le emozioni sopra elencate
- I video sono a colori e presentano differenti risoluzioni
- I partecipanti al dataset sono 75 e tutti di diverse età, etnie e sesso.

Dataset: MMI Facial Expression

Esempio – MMI

Di seguito un esempio di frame estrapolati dai video contenuti nel dataset MMI:

Dataset: FKT (Merged MMI & CK+)

Il dataset FKT rappresenta la fusione dei dataset MMI e CK+. Il motivo di questa scelta è dato dalla non omogeneità nella rappresentazione delle espressioni emotive, la quale fornisce la possibilità di catturare un range emotivo più ampio. Le caratteristiche del dataset sono:

ELENCO – Caratteristiche del dataset FKT

- Le espressioni rappresentate sono 6: **rabbia**, **disgusto**, **paura**, **felicità**, **tristezza** e **sorpresa**
- L'espressione **disprezzo**, seppur presente nel dataset CK+, non è stata considerata a causa dei pochi campioni, dovuti all'assenza dell'espressione nel dataset MMI

Dataset: Pre-processing CK+

Per standardizzare il formato delle immagini nel dataset, sono state eseguite diverse operazioni:

ELENCO – Pre-processing CK+

- **Ritaglio**: Per garantire un apprendimento privo di interferenze causate dal "rumore" introdotto dallo sfondo, è stato applicato un processo di ritaglio alle immagini in modo da includere solo il volto del soggetto, le cui coordinate sono state individuate impiegando il modello pre-addestrato di CV2 (OpenCV) haarcascade frontalface.
- **Scala di grigi**: Essendo le immagini sia a colori che in scala di grigi, si è deciso di uniformare il formato in scala di grigi.

Dataset: Pre-processing CK+

ESEMPIO – Pre-processing CK+

Di seguito alcuni esempi di pre-elaborazione di immagini del dataset CK+:

Dataset: Pre-processing MMI

Il processo di pre-processing del dataset MMI è più articolato in quanto, essendo costituito da video, è necessario provvedere dapprima a un'estrazione dei key frame di ogni video:

DEFINIZIONE – Key Frame

Un *key frame* rappresenta un singolo frame o una singola immagine chiave all'interno di una sequenza di frame o immagini. Definiscono posizioni, aspetti visivi o altri attributi significativi in un'animazione.

Il processo di estrazione consta di diverse fasi:

ELENCO – Pre-processing CK+

- **Conversione dei frame in scala di grigi**
- **Applicazione di un filtro Gaussiano**
- **Estrazione dei Key Frame**

Dataset: Pre-processing MMI

La conversione in scala di grigi viene attuata perché l'informazione del colore non è necessaria alla computazione dei key frame.

Il filtro gaussiano invece viene utilizzato per:

ELENCO – Utilità del filtro Gaussiano

- Ridurre il «rumore» introdotto da varie fonti, come la compressione dell'immagine o le interferenze ambientali.
- Ottenere un effetto di sfocatura controllata sull'immagine al fine di ridurre dettagli indesiderati o per ottenere un'immagine più uniforme

Matematicamente, il filtro gaussiano consiste nell'applicare per ogni pixel il prodotto di due funzioni Gaussiane (una per ogni dimensione):

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$, con σ deviazione standard e (x, y) coordinate del pixel

Dataset: Pre-processing MMI

Una volta elaborati i frame si può procedere all'estrazione dei key frame, la cui operazione consta dei seguenti passaggi:

ELENCO – Estrazione dei Key Frame

- Per ogni frame i e il suo successivo s $= i + 1$, viene calcolata la somma delle differenze assolute pixel pixel (x_i, y_i), (x_s, y_s), matematicamente:

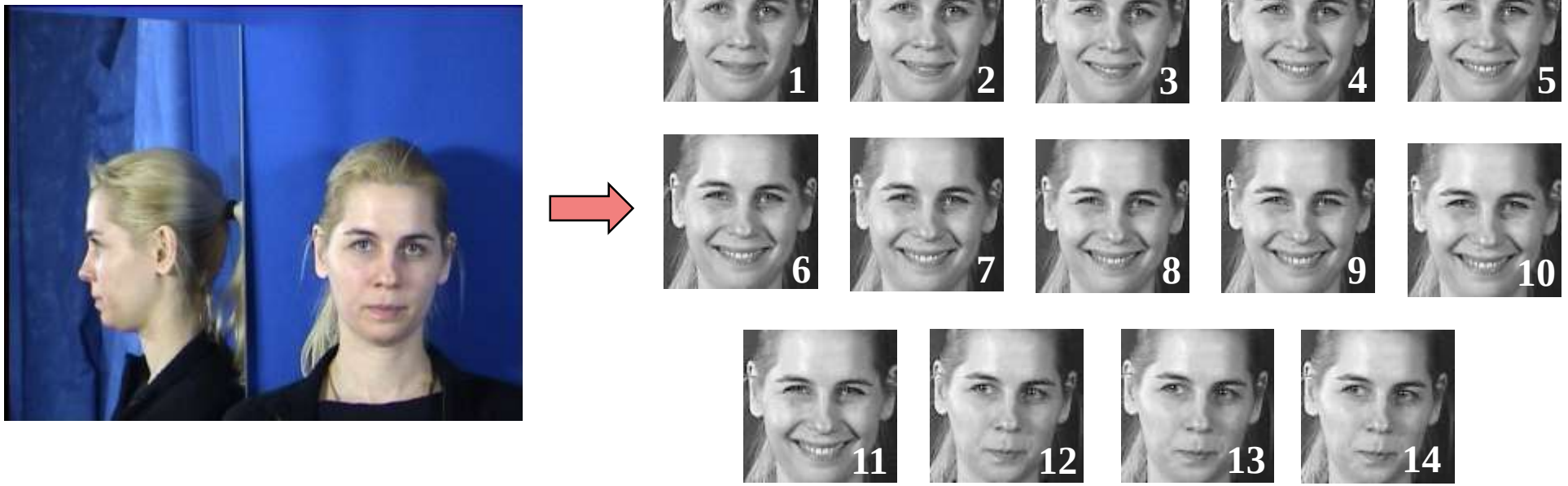
$$D = \left\{ d_{i,s} = \sum_{x,y} |p_{i,(x,y)} - p_{s,(x,y)}| \, , \forall i \in [0, n-1] \right\}$$

- Calcolo della media μ e della deviazione standard σ delle differenze calcolate D
- Calcolo del threshold $t = \mu + 0.65 \cdot \sigma$, con 0.65 valore empirico
- Selezione dei frame tale per cui la differenza con il precedente è maggiore del threshold $d_{i,s} \geq t$
- Rilevazione del volto e conseguente ritaglio

Dataset: Pre-processing MMI

Esempio – Pre-processing CK+

Di seguito un esempio di pre-processing di un video del dataset MMI:



Processo di Validazione

Per il processo di validazione è stata adottata la tecnica della K-Fold Cross Validation:

DEFINIZIONE – K-Fold Cross Validation

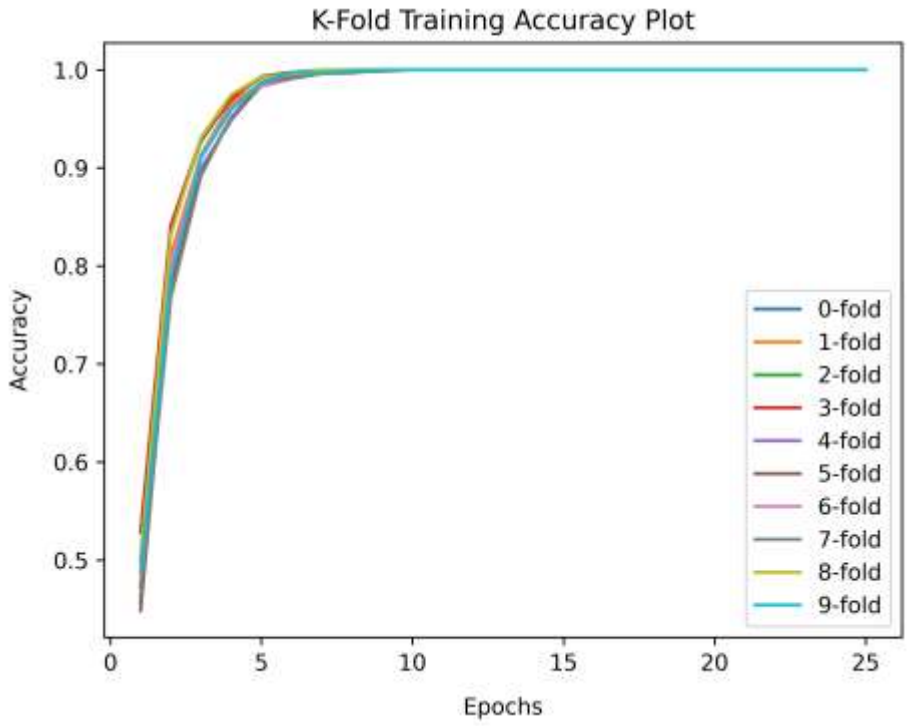
La *K-Fold Cross Validation* è una tecnica comune per valutare le prestazioni di un modello in modo robusto. Si suddivide il set di dati in k subset (fold) di dimensioni simili. Per ogni fold i, il modello viene addestrato sui $[k]\backslash\{i\}$ fold, mentre il fold i viene utilizzato per la valutazione. Alla fine delle k iterazioni, i risultati vengono combinati, ad esempio calcolando la media delle misure o la deviazione standard.

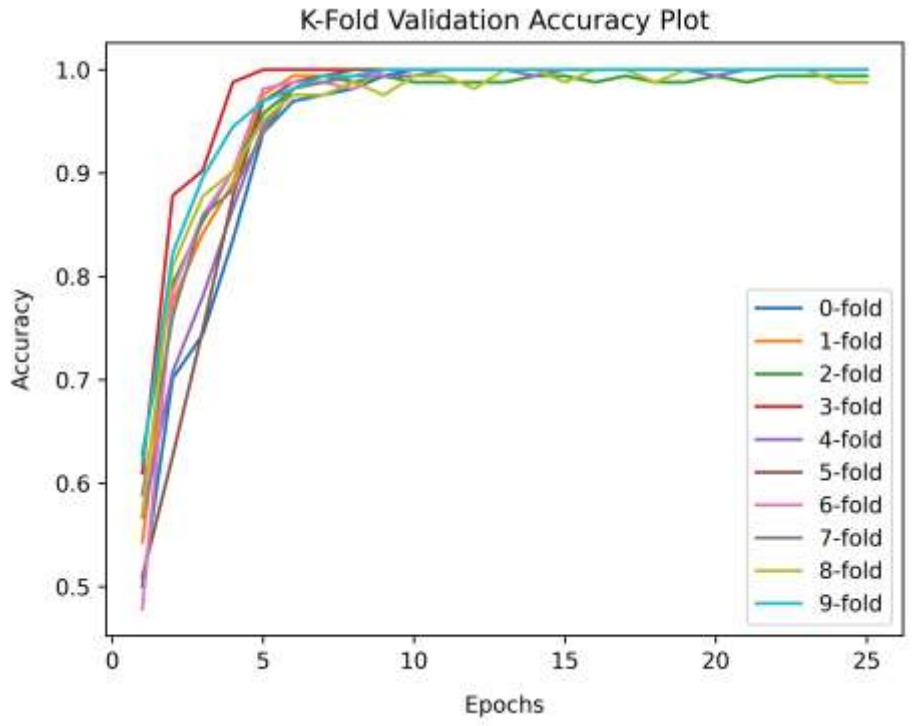
ELENCO – Parametri di Validazione

- **Numero di Fold:** 10
- **Dimensione del batch:** 128
- **Epoche:** 25
- **Tasso di apprendimento:** 0.0001

Processo di Validazione: CKP

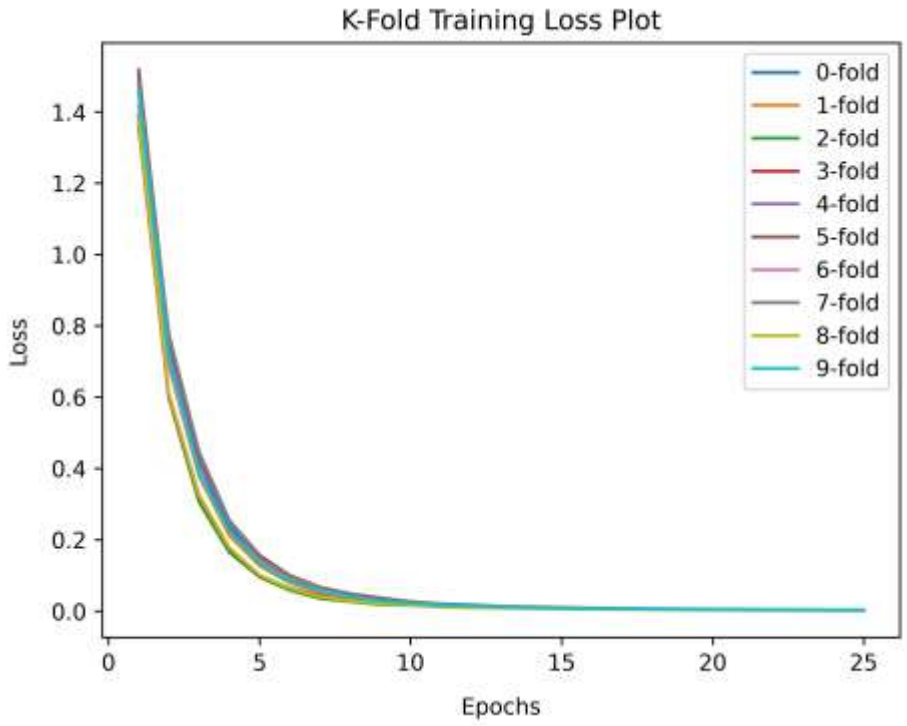
GRAFICI – Accuratezza K-Fold

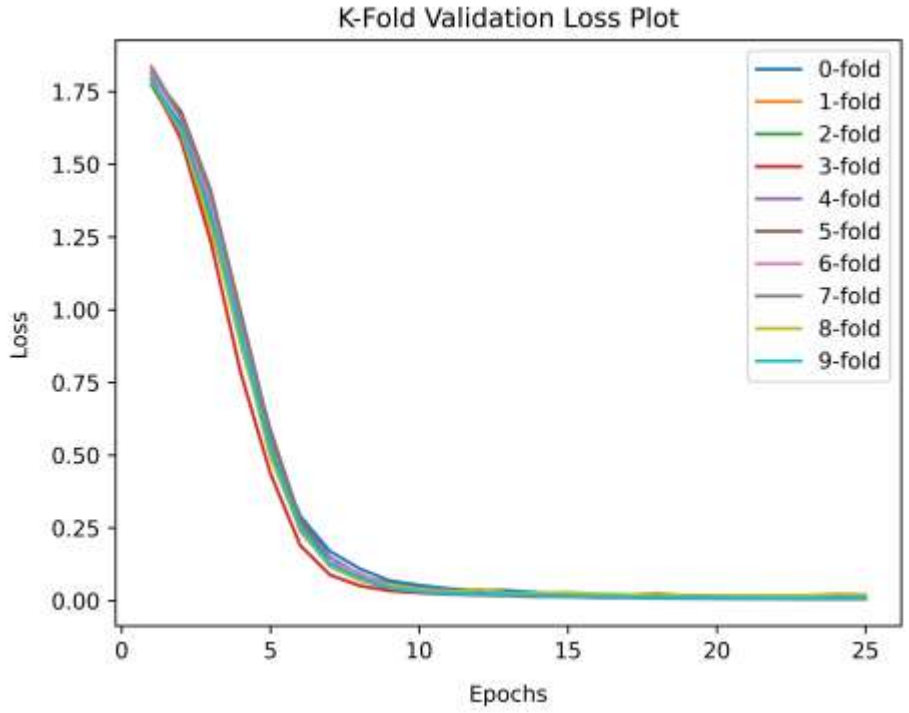




Processo di Validazione: CKP

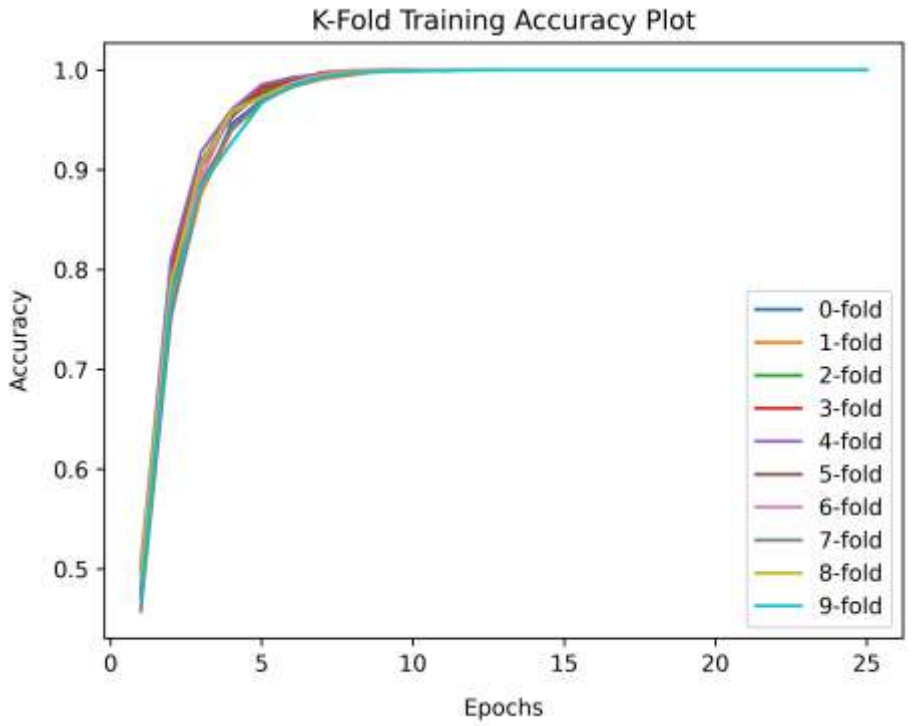
GRAFICI – Errore K-Fold

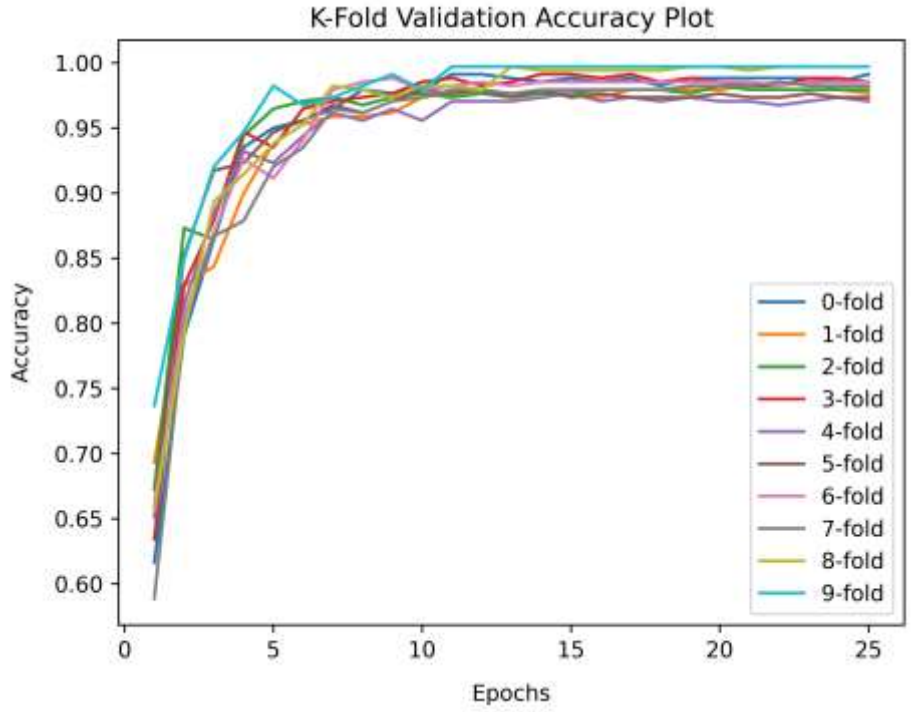




Processo di Validazione: MMI

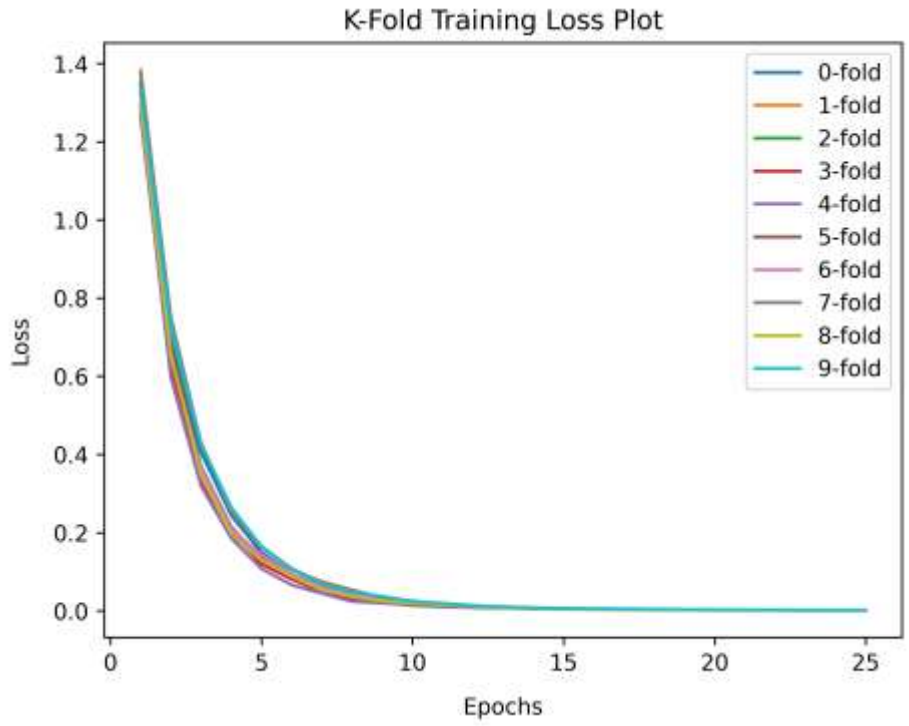
GRAFICI – Accuratezza K-Fold

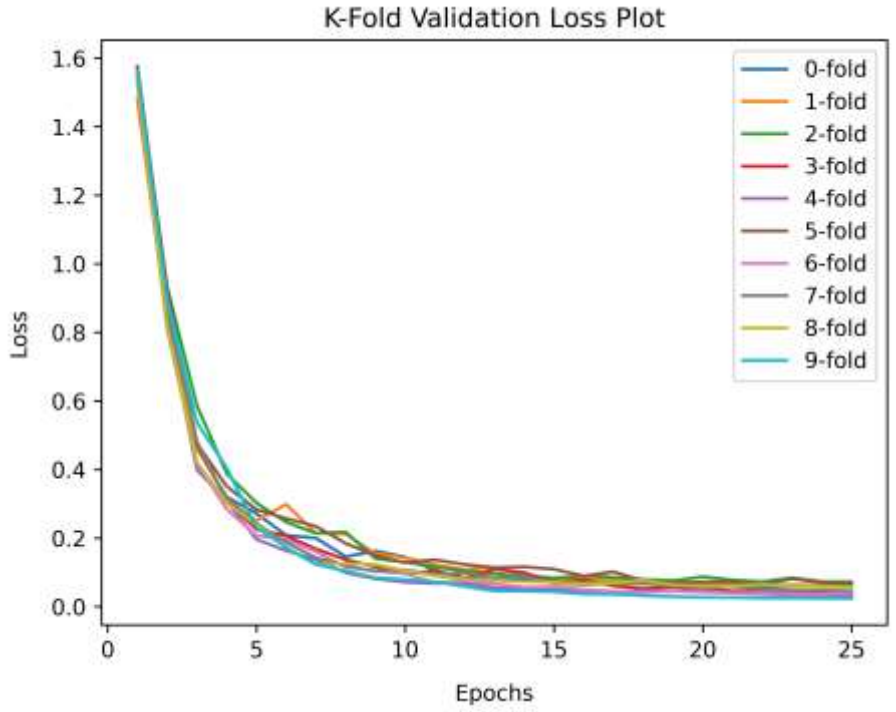




Processo di Validazione: MMI

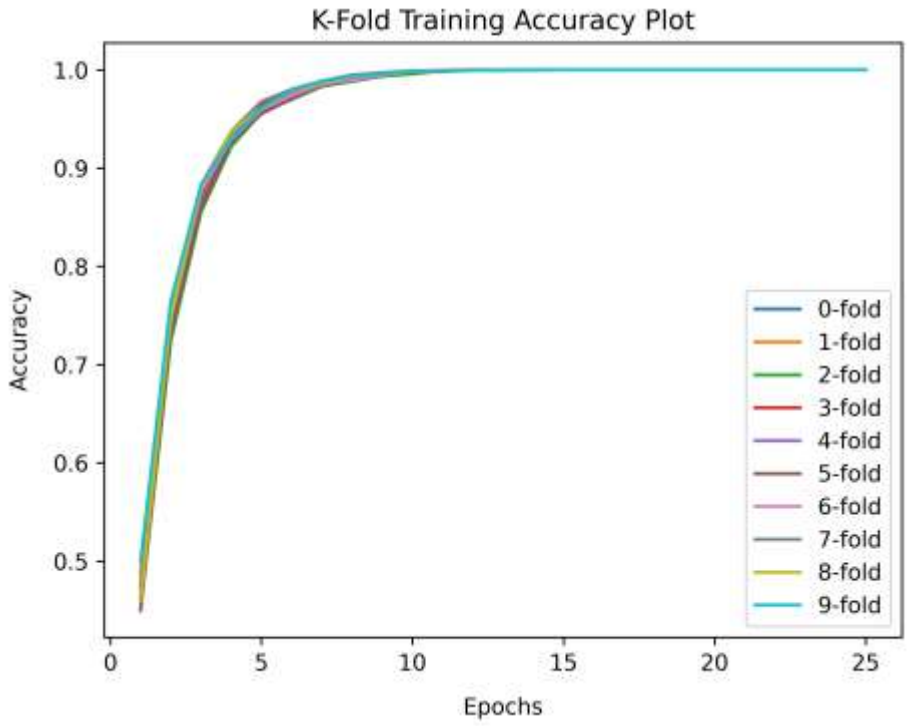
GRAFICI – Errore K-Fold

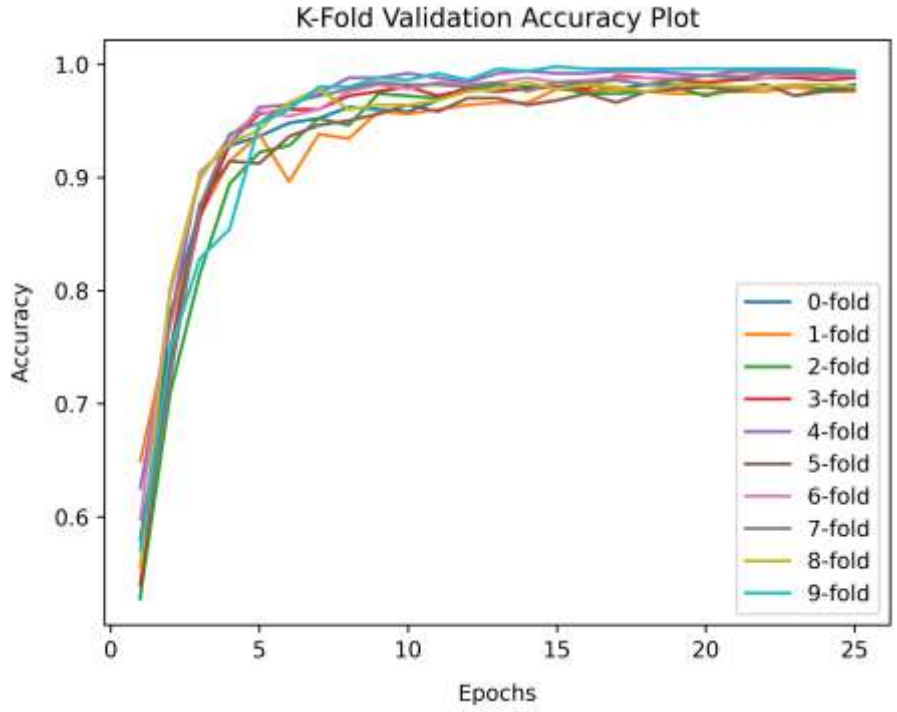




Processo di Validazione: FKT

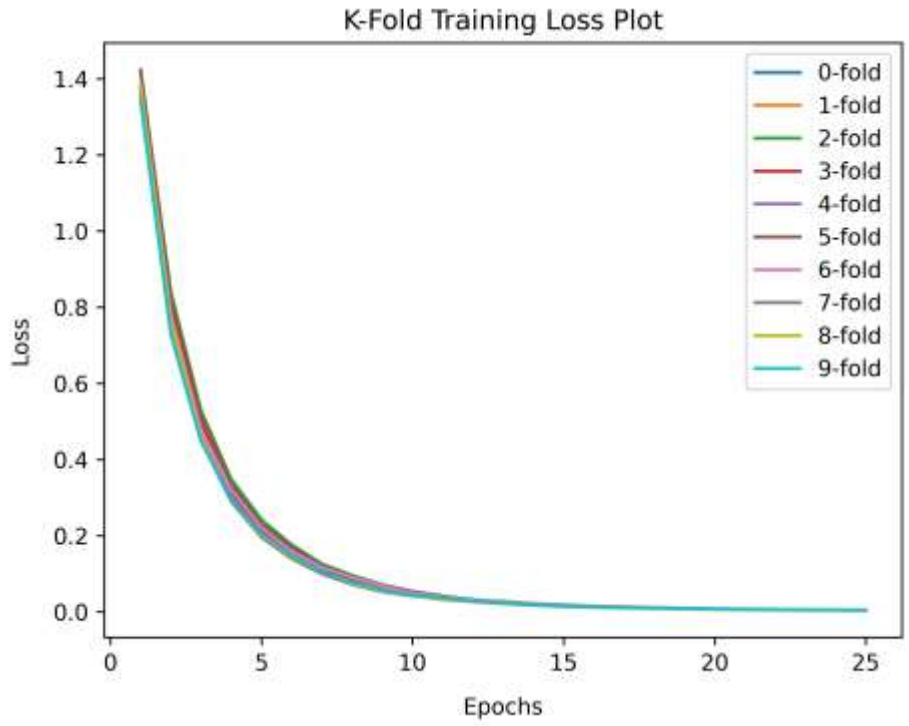
GRAFICI – Accuratezza K-Fold

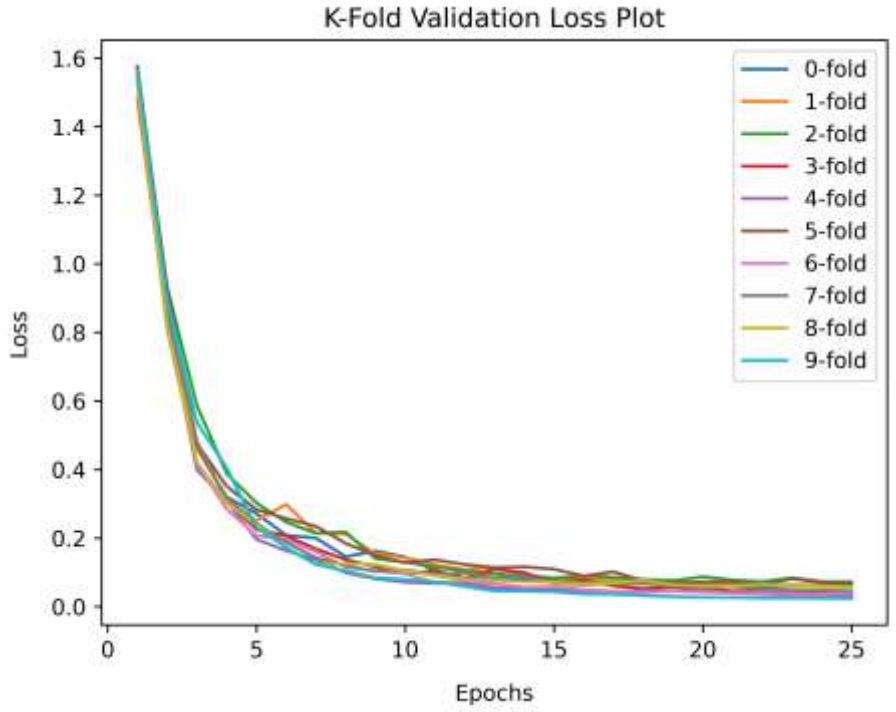




Processo di Validazione: FKT

GRAFICI – Errore K-Fold





Processo di Addestramento

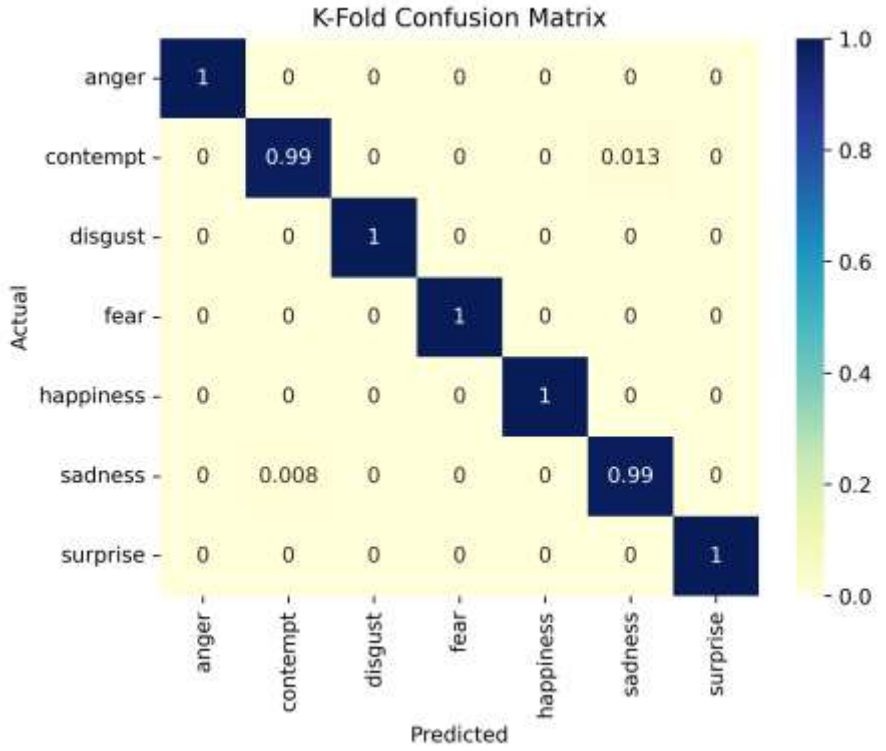
Dopo la fase di validazione, il modello è stato sottoposto al processo di addestramento effettivo per ciascun dataset. I modelli risultanti sono stati salvati in modo da poter essere successivamente ricaricati e utilizzati in altre applicazioni.

ELENCO – Parametri di Addestramento

- **Dimensione del batch:** 128
- **Epoche:** 25
- **Tasso di apprendimento:** 0.0001
- **Dimensione del Validation Set:** 25% del Dataset

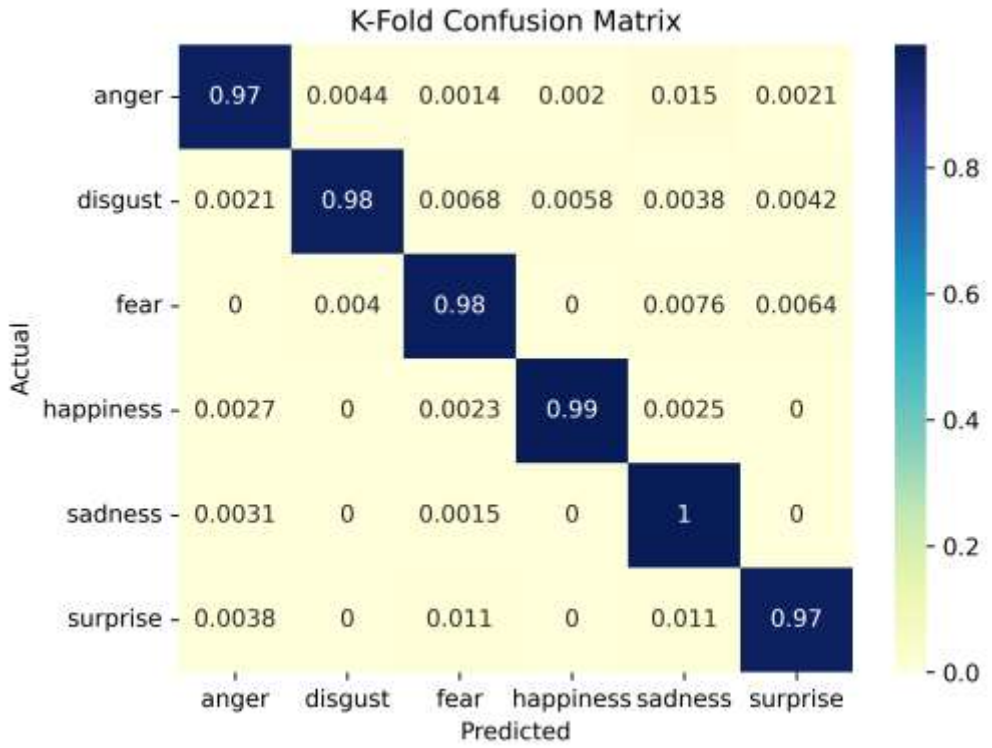
Processo di Addestramento: CKP

GRAFICI – Matrice di Confusione



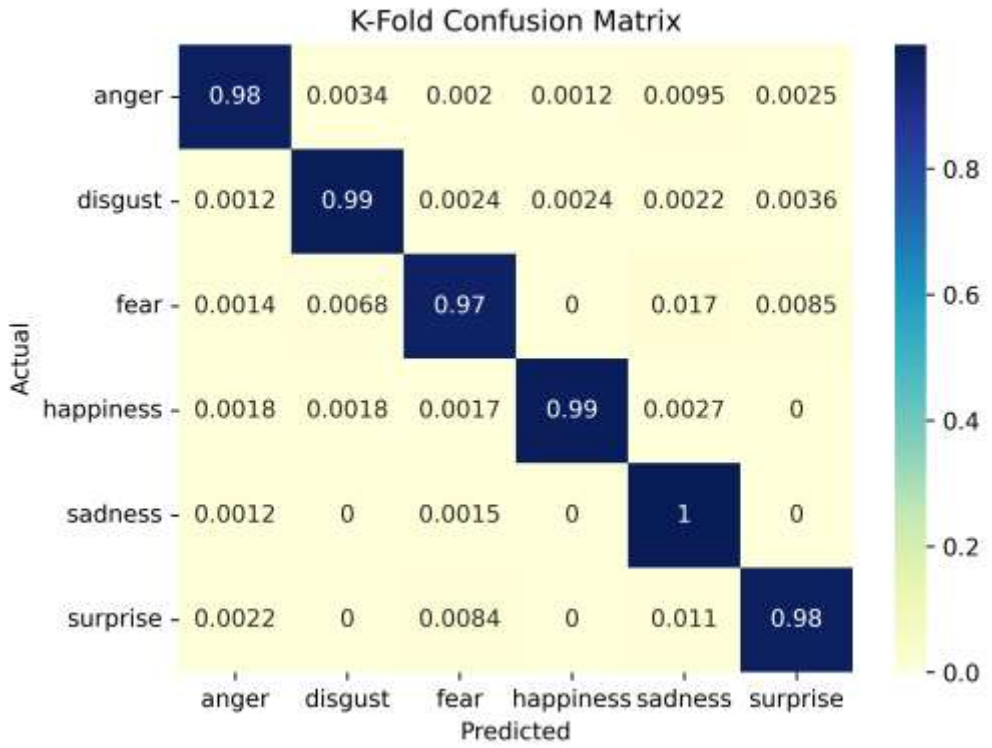
Processo di Addestramento: MMI

GRAFICI – Matrice di Confusione



Processo di Addestramento: FKT

GRAFICI – Matrice di Confusione



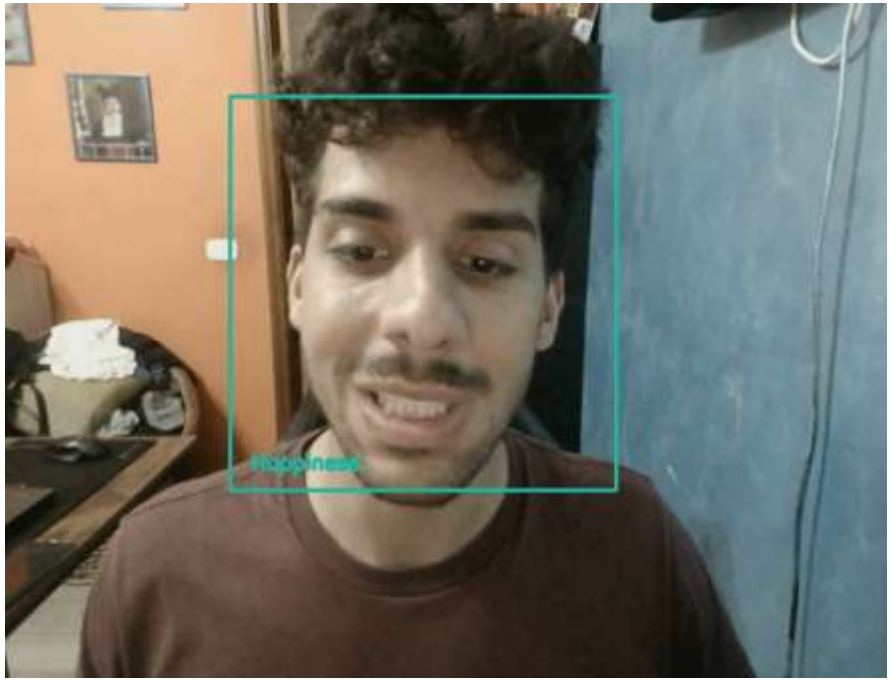
Esempi d'uso

Esempi d’uso: Offline

ESEMPIO – DeXpression FKT Offline

Esempi d’uso: Real Time

ESEMPIO – DeXpression FKT Real Time



Grazie per l'attenzione