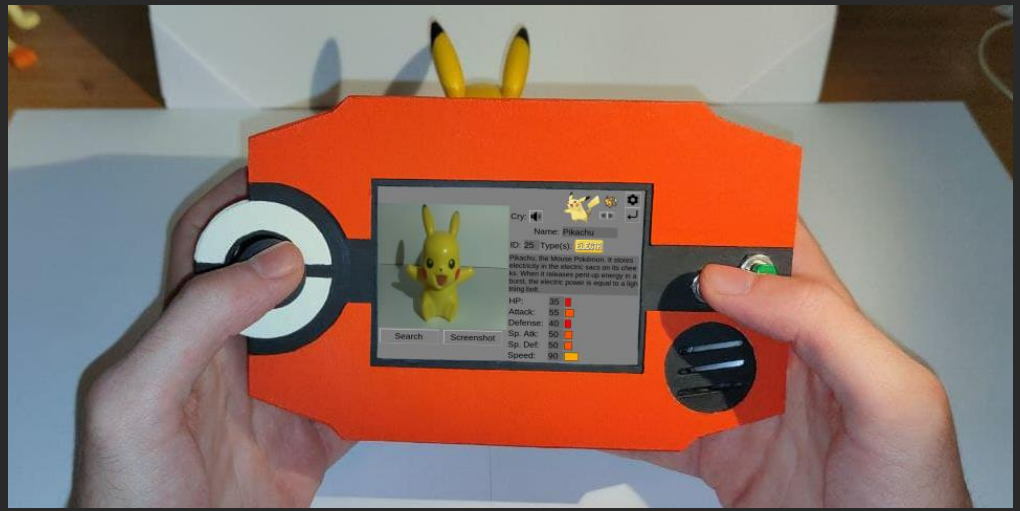
Poké-Pi-Dex

Classificazione di Pokémon tramite l'uso di CNN su dispositivi embedded



Sistemi Digitali M | karina-chichifoi michele-righi TryKatChup mikyll

Indice

Introduzione

Pokémon: Panoramica

Videogioco, gioco carte collezionabili, cartone animato, fumetto

Suddivisi in **tipi**, i.e. ELETT FUOCO ERBA

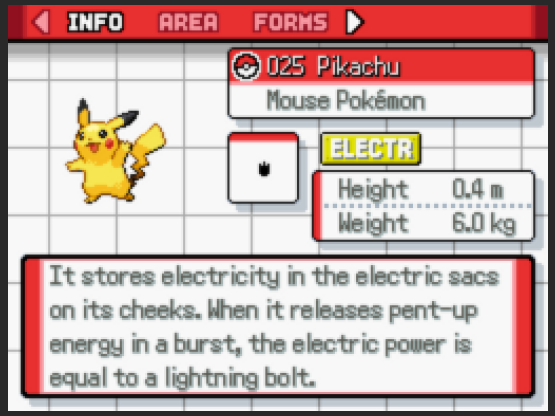
Pocket Monsters: possono essere catturati, allenati e usati in combattimento

Possiedono valori specifici (**statistiche**)

Pokédex

Dispositivo che **tiene traccia** dei Pokémon visti e catturati

Scansiona e **riconosce** il Pokémon catturato o visto in battaglia



Registra le **caratteristiche** del Pokémon catturato

Mostra **statistiche** Pokémon

Obiettivi del Progetto

Realizzazione di un dispositivo embedded simile a un **Pokédex** e **a portata di mano**

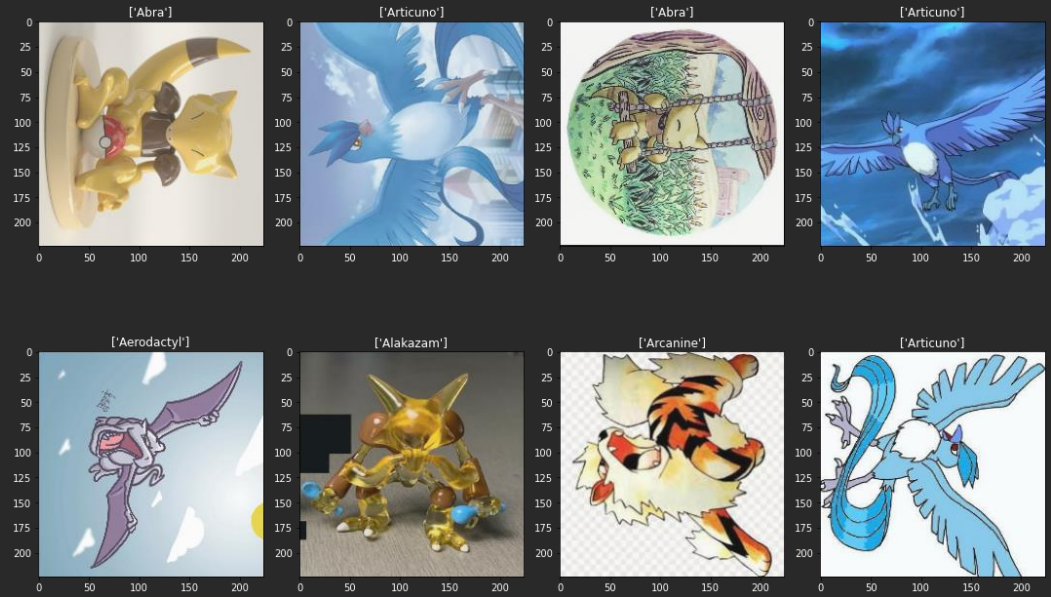
Riconoscimento di **carte**, **peluche**, **immagini** della serie animata, **modelli** 3D, pixel art

01 Classificatore

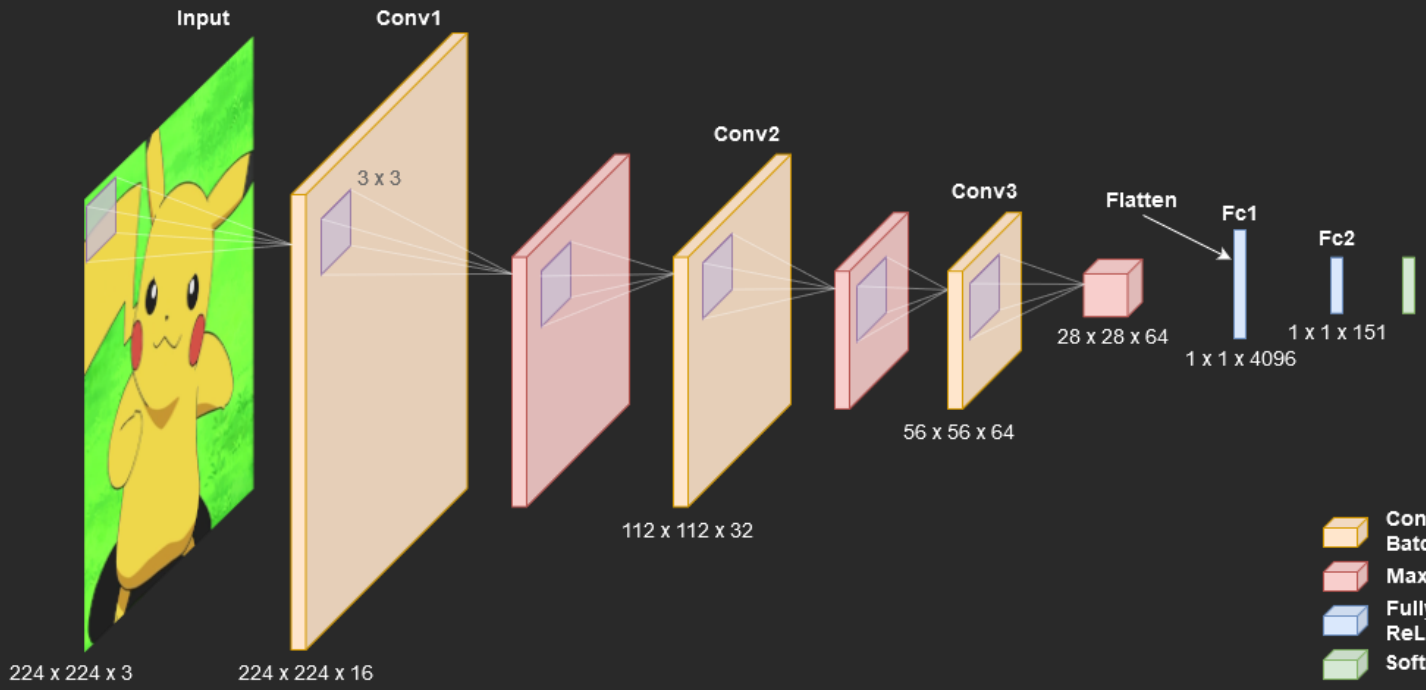
Preprocessing dei Dati

Dataset di **7000 immagini** trovato su Kaggle e opportunamente diviso

Sono state effettuate operazioni di **Data Augmentation**



Architettura della Rete



Allenamento del Modello

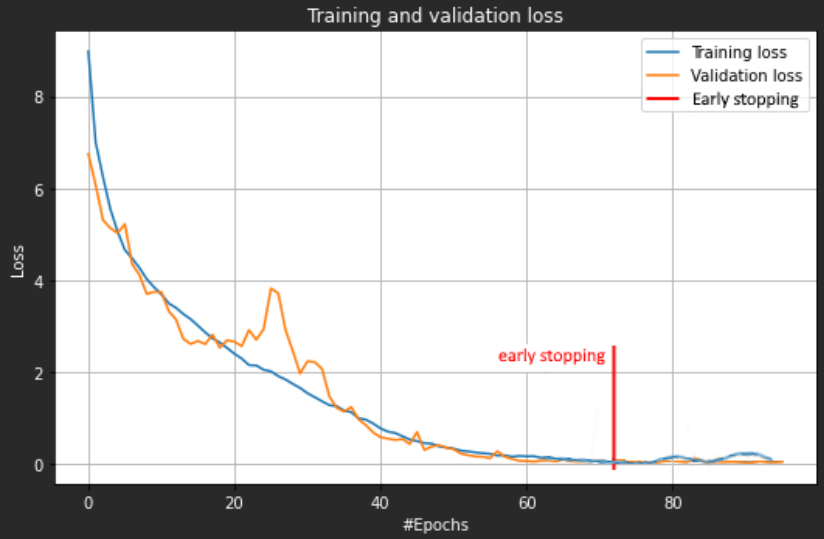
Framework **Tensorflow 2.6** + Keras (integrato)

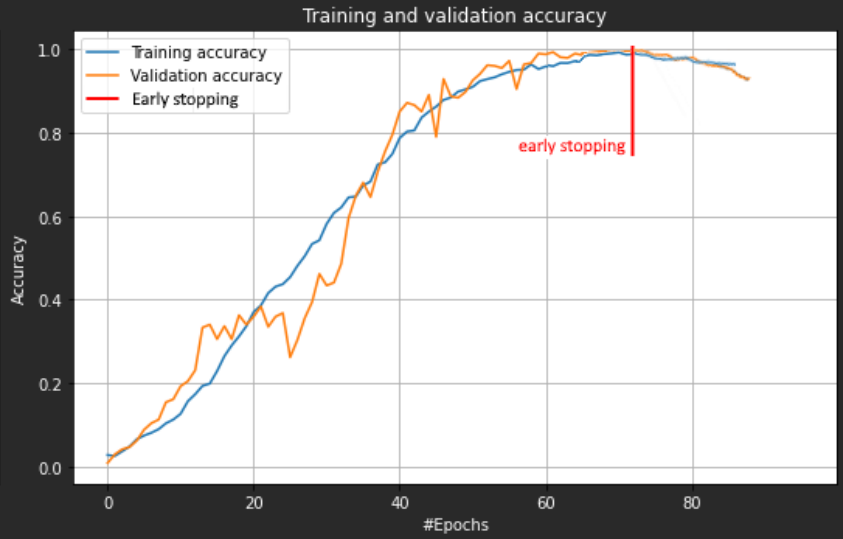
Ottimizzatore: **Adam**

Early stopping: necessario per fermare l'allenamento in caso di peggioramento di performance del modello

Training con **epochs = 100** e **batch size = 64**

Performance Primo Modello





Loss: 0.0176
Accuracy: 0.9983
Numero di epoche: 75

Osservazioni e Conseguenze

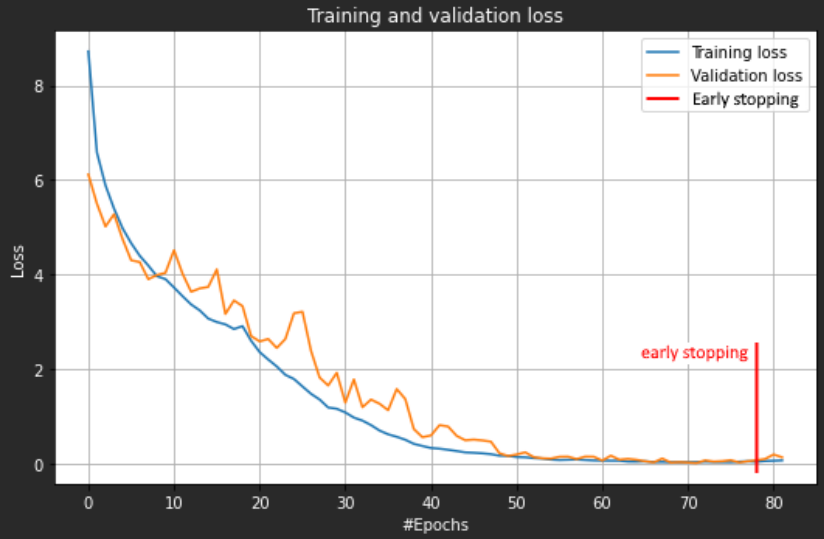
Abbiamo visto che le **performance** sono molto **soddisfacenti**

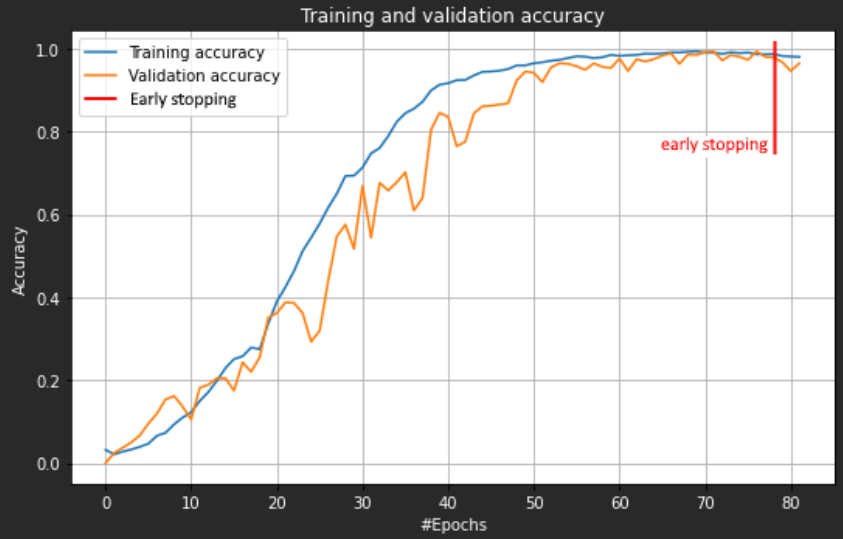
Provando sul dispositivo abbiamo visto che **non riconosceva molti Pokémon**

Dataset di qualità scarsa, con numerosi errori

Abbiamo ricreato un **nuovo dataset da 11945** immagini, di cui soltanto 8238 sono state usate

Performance Secondo Modello





Loss: 0.0262
Accuracy: 0.9933
Numero di epoche: 72

Conversione in TFLite

Il modello ha un peso di 1.2 GB. Per questi motivi:

È stato esportato in **Tensorflow Lite** per limiti di memoria e di spazio su sistemi embedded

È stato compresso tramite tecniche di **quantizzazione**

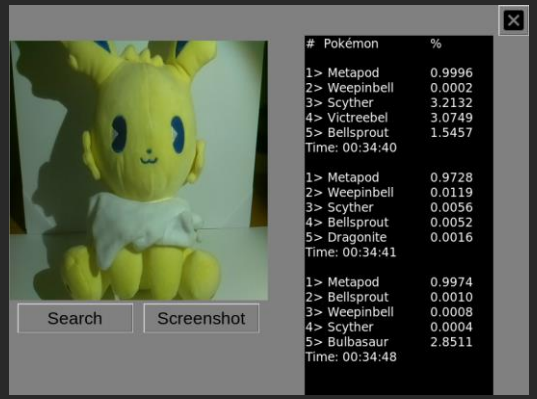
Peggioramento di accuracy pari all'**1%** rispetto al modello originale

Peso finale: **93 MB**

04 Conclusioni

Conclusioni

Riconoscimento in circa **200ms**



85% accuratezza della predizione in casi reali

Predizione **semi-errata** per **Pokémon molto simili** o con sfondi fraintendibili

È necessario **illuminare sufficientemente** la zona inquadrata

https://github.com/TryKatChup/Poke-Pi-Dex