

ANIMAL CLASSIFICATOR

Il progetto tratta lo sviluppo di un classificatore di immagini di animali, in particolare è stato utilizzato un apprendimento supervisionato. La struttura della rete è ispirata ad una residual neural network.

Come dataset è stato fatto uso di un insieme di circa 28000 immagini appartenenti a dieci diverse classi di animali: cani, cavalli, elefanti, farfalle, galline, gatti, mucche, pecore, ragni e scoiattoli. Le immagini sono molto varie: alcune contengono un solo animale, altre invece ne presentano diversi ma sempre della stessa razza. Inoltre, in alcune immagini gli animali non sono completamente visibili, ma è visualizzabile solo una parte di essi.

Il dataset è stato inizialmente suddiviso in trainset e testset, con una proporzione 80 – 20. Il trainset è stato anche sottoposto ad un processo di image augmentation.

Dopo aver addestrato il modello e aver verificato l'efficacia sul testset, su un insieme più piccolo di immagini è stato testato come il modello reagisse a delle immagini non appartenenti a nessuna delle dieci classi di animali su cui era stato addestrato.

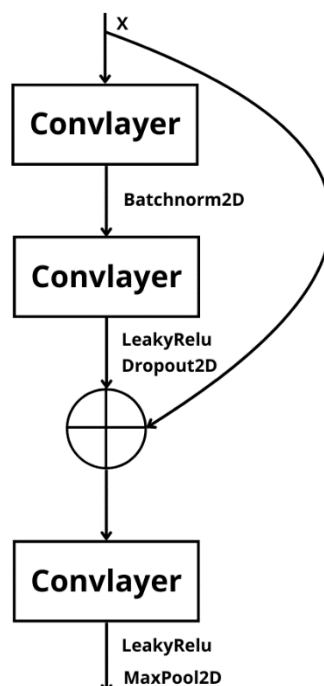
STRUTTURA DELLA RETE

La struttura della rete è ispirata ad una residual neural network, sfruttando un blocco chiave chiamato Resblock.

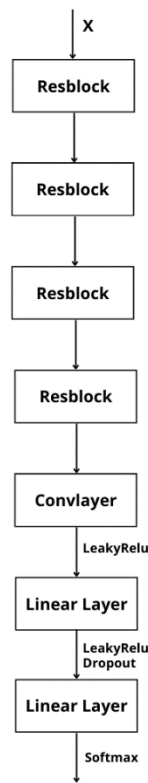
Ogni Resblock è composto da due layer di convoluzione, tra i due è applicata una batchnorm2D, che consiste in una normalizzazione dei dati attraverso re-scaling e re-centering. Esse sono delle tecniche di preelaborazione, il re-centering consiste porre a zero la media delle variabili, il re-scaling invece pone a 1 la deviazione standard delle variabili. A livello empirico si nota che l'utilizzo di queste tecniche migliorano la velocità di convergenza e riducono il rumore dato che tutte le feature sono scalate in modo uniforme.

Infine, viene utilizzato un Dropout2D, utile in fase di training per porre a zero casualmente il 10% dei layer riducendo così il rischio di overfitting; ciò significa che l'output di questa parte selezionata viene ignorato.

Questi dati elaborati vengono passati insieme agli originali ad un ultimo layer convolutivo evitando la perdita di informazioni tramite la concatenazione.

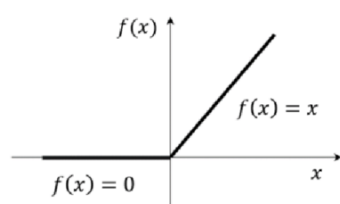


La rete è stata costruita con quattro Resblock, da un layer di convoluzione e due lineari.

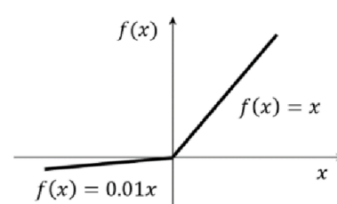


Inoltre, per la fase di training sono stati utilizzate due funzioni di loss differenti: una normale cross entropy e una cross entropy con dei pesi calcolati in base al numero di dati disponibili, in modo da compensare lo squilibrio dovuto al fatto che alcune specie di animali avevano un numero di immagini minori rispetto alle altre. Le due diverse funzioni sono state alternate in un ciclo di 25 epoch.

Come funzione di attivazione è stata sempre utilizzata una LeakyRelu e non una Relu in quanto questa funzione in caso di inputs negativi non restituisce zero ma piccoli valori al di sotto di esso riducendo il problema del vanishing gradient.



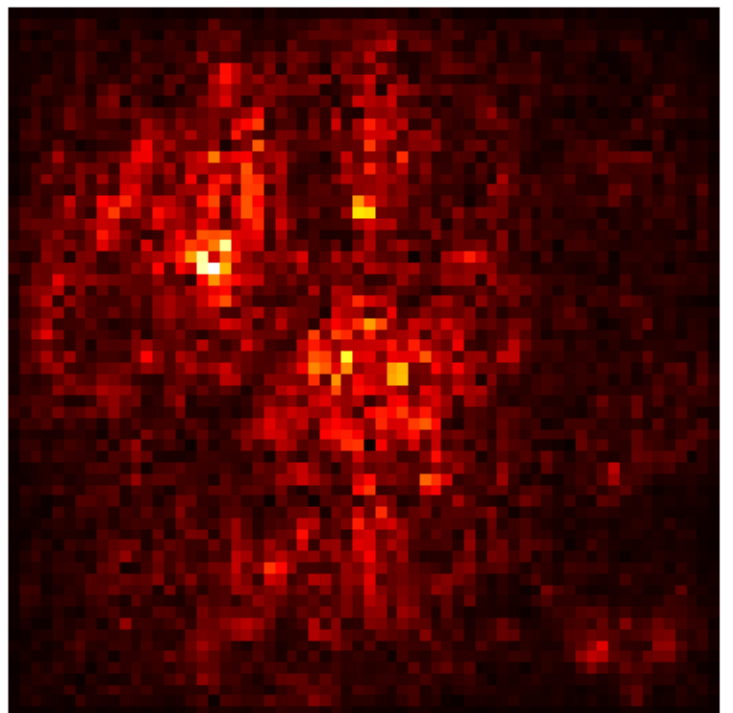
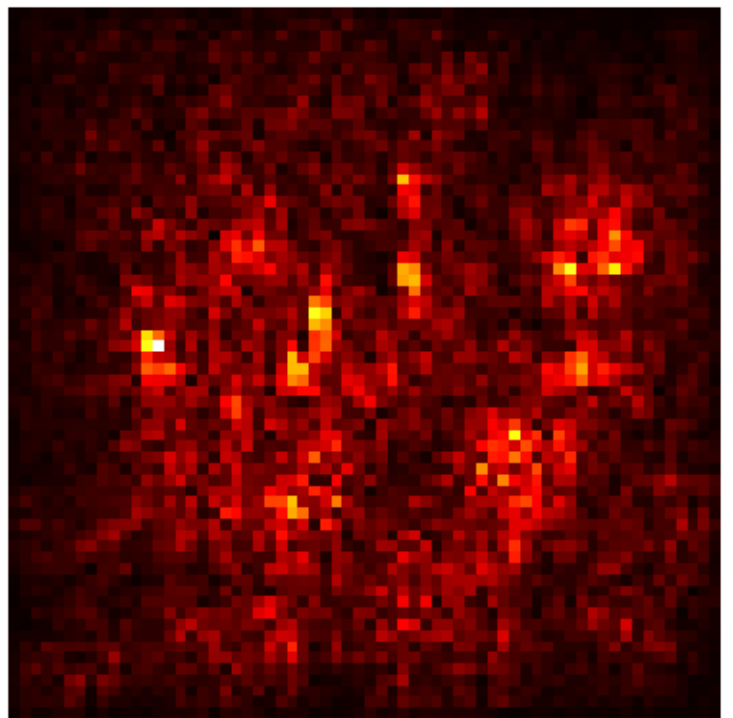
ReLU activation function

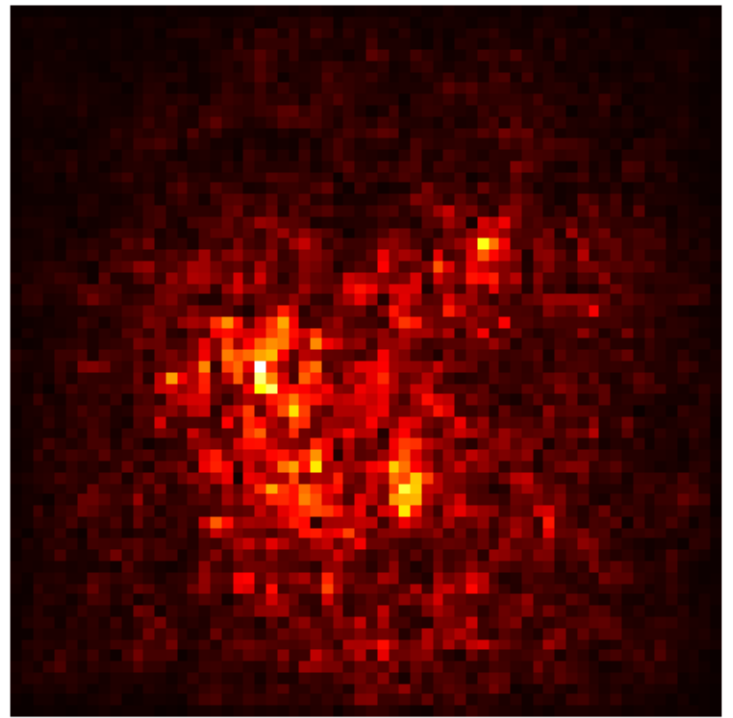


LeakyReLU activation function

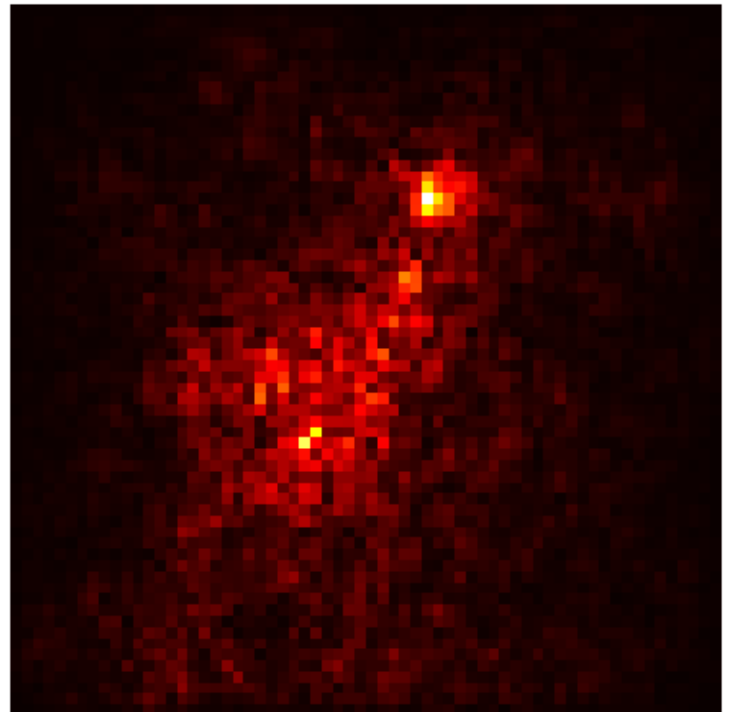
SALIENCY MAP

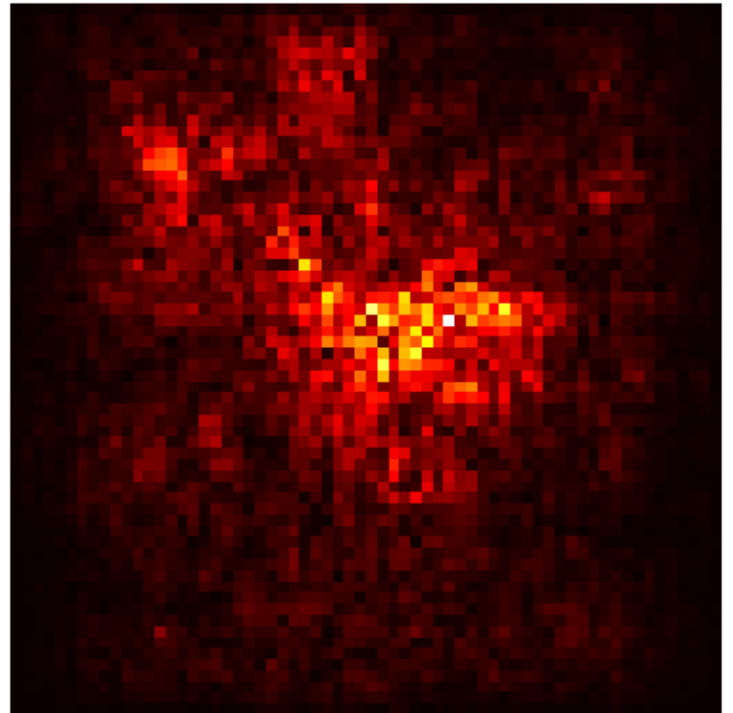
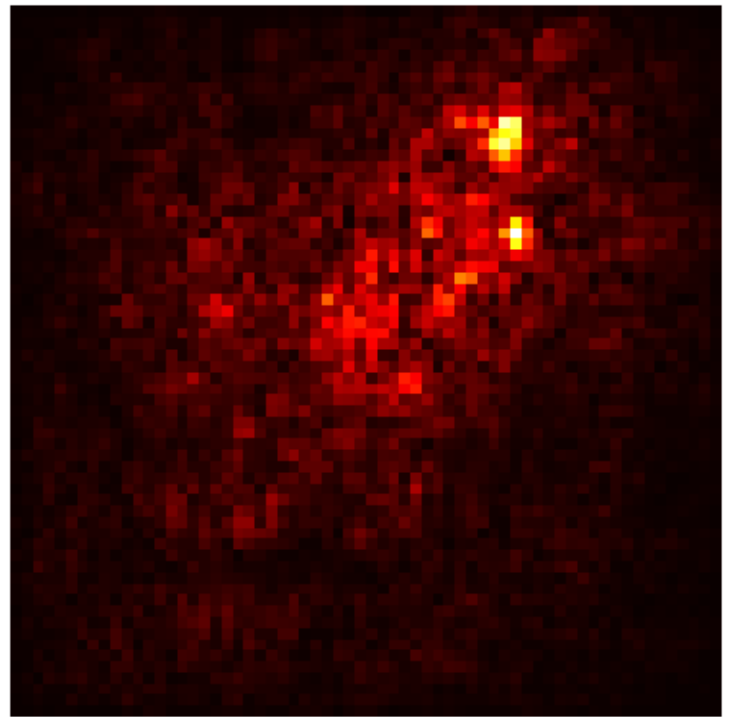
Terminata la fase di training è stata implementata anche la visualizzazione delle saliency map. Le saliency map possono essere utilizzate per capire quali parti dell'immagine il modello utilizza per classificarla. Ad esempio, studiando alcune immagini della classe gallina e la visualizzazione della relativa saliency map si può ipotizzare che gli elementi che il classificatore utilizza per determinare l'appartenenza a questa classe siano principalmente le code e in secondo luogo la forma della testa.





Un secondo esempio sono i cavalli: studiando le saliency map si può ipotizzare che l'elemento principale utilizzato per il riconoscimento siano testa e collo, in maniera minore corpo e orecchie.





Per ogni classe studiata potrebbero poi essere fatte delle considerazioni per stabilire quali elementi dell'immagine sono utilizzati maggiormente per stabilire la classificazione.

RISULTATI E CONCLUSIONI

Terminato la fase di training è stata stampata una confusion matrix, utilizzando il testset, per visualizzare l'accuratezza della classificazione:



Sulle colonne sono rappresentati i valori predetti e sulle righe quelli reali. I valori sulla diagonale maggiore rappresentano la percentuale corretta di ogni classe. L'elemento sulla riga i e sulla colonna j rappresenta la percentuale dei casi in cui il modello ha classificato la classe reale i come classe j . È quindi possibile capire quali sono le classi più confuse analizzando questa matrice. In particolare, possiamo notare che le classi che il modello confonde maggiormente sono ragni e farfalle. I valori sono approssimati a quattro cifre significative, eventuali errori sono dovuti a questo fatto.

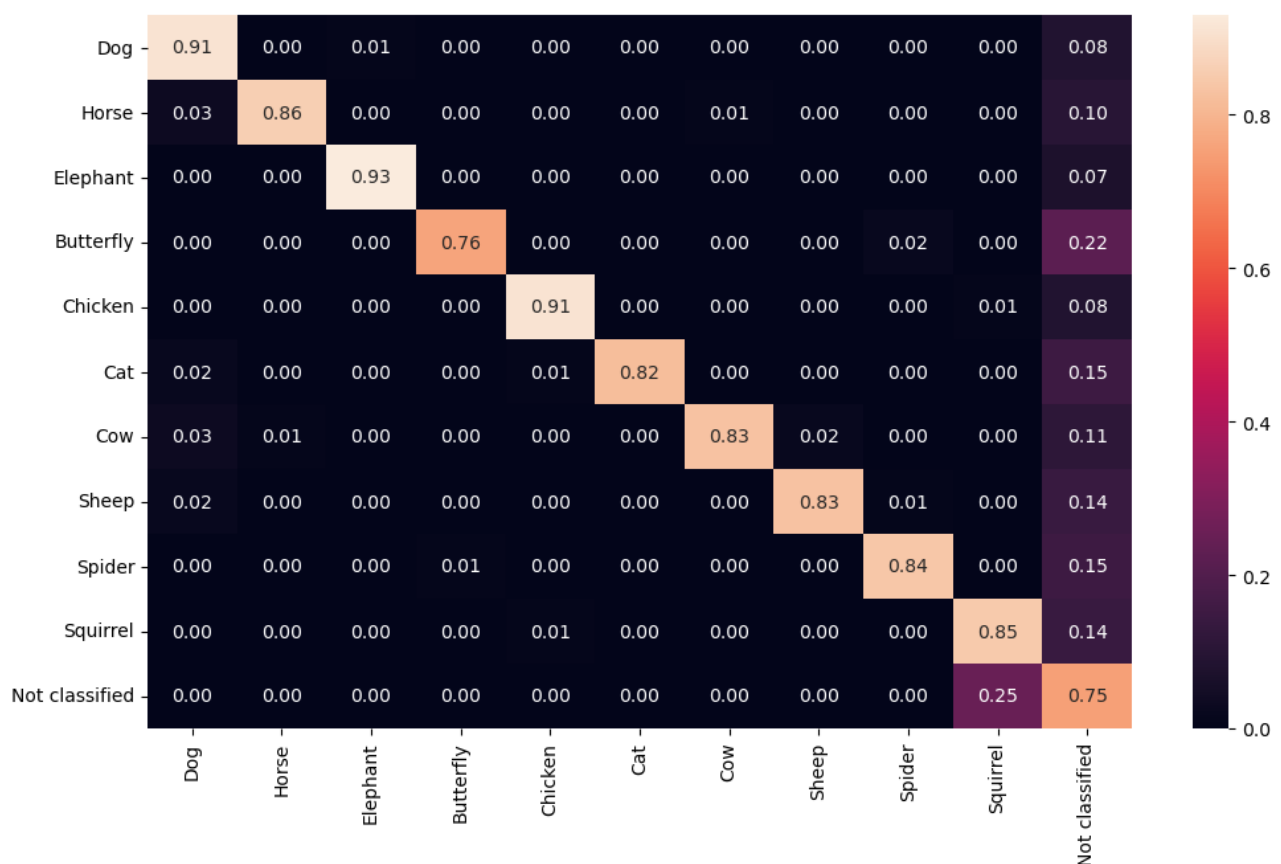
Inoltre, l'accuratezza della classificazione raggiunta sul trainset è del 94,2%.

In conclusione, su un insieme ridotto del testset è stata testata anche la capacità di riconoscere delle immagini che non appartengono a nessuna delle dieci classi su cui è stato fatto training. L'approccio scelto per questa funzione fa utilizzo di un valore di threshold. Per descrivere questo approccio è necessario prima spiegare quale sia l'output del modello.

L'output del classificatore rispetto ad ogni immagine di input altri non è che un vettore di valori numerici, la cui dimensione è uguale al numero di classi, i valori sono tutti compresi tra 0 e 1, quello più alto rappresenta la classe che il modello identifica come reale e ne rappresenta la "sicurezza" con cui l'ha identificata.

Per verificare che un'immagine non appartenga a nessuna delle dieci classi è necessario confrontare il valore massimo del vettore di output dell'immagine stessa con quello di threshold, se il primo valore è minore del secondo allora il modello classificherà l'immagine come non appartenente a nessuna classe, altrimenti essa verrà catalogata normalmente. Il valore di threshold scelto è di 0.999,

Anche in questo caso è stata stampata la relativa confusion matrix:



Il classificatore riconosce il 75% delle immagini sottoposte che non appartengono a nessuna delle dieci classi conosciute. Ovviamente la precisione del modello diminuisce leggermente in quanto alcune immagini

vengono classificate come non appartenenti a nessuna classe. Inoltre, l'errore più comune è quello di identificare l'immagine come non appartenente a nessuna classe.

La principale criticità riscontrata nella costruzione del modello è stata la difficoltà nel riconoscimento di ogni classe, in particolare nei primi prototipi del classificatore la classe dei gatti non veniva riconosciuta e la maggior parte degli esemplari veniva confusa con i cani. Il problema probabilmente era dovuto al fatto che il numero di esemplari di gatti nel dataset era il più basso tra le varie classi e quello dei cani invece il più alto. Per risolvere questo problema sono stati introdotti dei pesi personalizzati per ogni classe, calcolati in base alla percentuale di esemplari di ogni classe rispetto al totale.

L'altra piccola criticità è che il modello è leggermente in overfitting; infatti, è sempre molto sicuro delle sue previsioni, per questo motivo il threshold utilizzato è molto alto. Una possibile proposta per risolvere questo problema e migliorare ulteriormente il modello sarebbe implementare una funzione personalizzata della cross entropy impostando una label che non punta a 1 ma ad un valore leggermente più basso (ad esempio 0.95) in modo che il classificatore non si sbilanci troppo.