UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

"RETI NEURALI CONVOLUZIONALI PER L'IDENTIFICAZIONE DI LEONI AFRICANI DA PATTERN AUDIO"

Relatore: Prof. Loris Nanni

Laureando: Matteo Spinato

ANNO ACCADEMICO 2021 – 2022

Data di laurea 21 luglio 2022

0. Abstract

Nell'ambito dell'Intelligenza Artificiale, le Reti Neurali Convoluzionali ricoprono un ruolo importante, se non principale, nel riconoscimento di immagini e video, di suoni, del linguaggio naturale, e di molti altri aspetti. La loro versatilità è resa possibile dalla tecnica del *Transfer Learning*, ovvero dalla capacità di alterare il tipo di oggetto da riconoscere (pattern) usando la conoscenza già assimilata: al posto di addestrare tutta la rete a riconoscere nuovi pattern, si addestrano solo gli ultimi strati connessi della rete in questione. Si può così, ad esempio, utilizzare una rete già addestrata al riconoscimento di immagini, cambiare i suoi ultimi livelli, e adattarla al riconoscimento di suoni.

In questo lavoro studiamo tramite la piattaforma Matlab le performance di tre reti neurali, create per il riconoscimento di immagini, nell'identificazione dei leoni dal loro ruggito. Abbiamo a disposizione un dataset di 164 ruggiti, e 26 tecniche di rappresentazione dei pattern audio. Il risultato conferma che con questi strumenti è possibile ottenere un alto livello di accuratezza nel problema di identificazione: con una singola rete si arriva fino a quasi il 94% di accuratezza nell'identificazione, ed ancora di più se si usa un *ensemble* di due o tre reti convoluzionali.

Parole chiave: CNN (Convolutional Neural Network), Identificazione, Pattern audio, Leoni Africani

Precisazioni: Questo lavoro di tesi è la diretta prosecuzione di un altro lavoro già svolto dal collega Martino Trapanotto [1], e comprende l'avanzamento del codice Matlab, la modifica del protocollo da usare, e l'aggiunta di nuove feature. Non esiste alcun conflitto di interesse.

1. Introduzione

Tra le infinite fonti di suoni e rumori con cui addestrare le reti neurali, quella scelta per questo lavoro è esemplare. La specie dei Leoni Africani (*Panthera Leo*), infatti, è caratterizzata dall'emissione di forti ruggiti, usati per identificare l'animale, definire il territorio, supportare gli individui del proprio gruppo ed allontanare gli altri o le specie non gradite.

Per raggiungere questi obiettivi, è di primaria importanza che il destinatario del ruggito decodifichi quale individuo ha lanciato il messaggio, e ciò ne determina la risposta. Nel messaggio, quindi, deve essere presente l'identità del mittente, e l'obiettivo del nostro lavoro è proprio quello di distinguere un individuo di leone dal suo ruggito tramite le reti neurali.

Studi in altri mammiferi come elefanti [2] e tigri [3] hanno confermato la presenza di componenti audio uniche nei versi di ogni esemplare, fondamentali per allegare informazioni sull'identità dell'animale.

Purtroppo, il verso emesso deve fronteggiare la degradazione progressiva del segnale in propagazione, in quanto le componenti audio in alta frequenza si corrompono velocemente, deformando il ruggito. Questa degradazione influenza l'abilità del ricevitore di estrarre le informazioni, come dimostrato da altri studi [4]. Malgrado questa perdita di informazione, i leoni sono comunque in grado di riconoscere ed analizzare anche un segnale degradato.

Conoscere il modo in cui nei versi vengono nascoste informazioni è fondamentale per capire meglio la comunicazione tra animali, ma è anche un nuovo approccio dei ricercatori per monitorare la popolazione di una specie in modo non invasivo. Tuttavia, siamo ancora distanti da mettere in pratica tutto questo.

I leoni emettono un richiamo particolarmente basso e potente, il famoso ruggito. Un singolo ruggito è solitamente composto da uno o più gemiti leggeri, seguiti da molti ruggiti a squarciagola, che terminano in una serie di grugnii leggeri. La bassa frequenza che caratterizza questi suoni è attribuita alle lunghe corde vocali tipiche della specie. Sia maschi che femmine emettono questi suoni, entrambi per comunicare attraverso lunghe distanze e per allegare informazioni riguardo i confini del territorio. [6] Degli studi hanno poi mostrato come i leoni possano usare i loro ruggiti sia per accumulare informazioni sul numero del gruppo e calcolare le loro chance in uno scontro, sia per aggiungere informazioni sulla loro identità. [4] Altri hanno esaminato gli attributi dei ruggiti individuali e hanno trovato piccole differenze tra i maschi e le femmine, comunque ne sappiamo ancora poco riguardo la

differenza tra i due. La principale causa di questa mancanza di informazioni è dovuta alla difficoltà di avere grandi dataset di registrazioni, e questo sottolinea la necessità di nuovi approcci. [4]

Tornando al nostro lavoro, per essere in grado di identificare le informazioni all'interno del ruggito, è necessario estrarre le feature dal segnale audio. Metodi comuni della tecnologia odierna si basano sulle caratteristiche della frequenza fondamentale, sullo studio armonico, sull'analisi del *Mel-Frequency Cepstral Coefficients* (MFCC), o ancora e su tecniche di modellazione come i modelli di Markov nascosti per provare a riconoscere le informazioni sottostanti nei versi degli animali. [4][5]

Queste tecnologie non sono ancora considerate lo stato dell'arte, e raggiungono circa il 90% di accuratezza in questo dataset di cui anche noi disponiamo. Comunque vedremo meglio in seguito vari metodi utili per estrarre le feature audio.

In questo lavoro si esplorerà poi come queste e altre tecniche di rappresentazione del segnale, come lo *Spectrogram* o il *Mel Spectrogram*, possono essere usate per addestrare una rete neurale a riconoscere gli animali.

Andando ad analizzare il lavoro svolto, il training set e il test set sono stati definiti usando l'approccio *Leave One Out Cross Validation* (LOOCV), generando le feature desiderate dal nostro dataset e dando queste in pasto alla CNN, misurando l'accuratezza nel riconoscere i leoni dal loro ruggito. Questo approccio avanzato ci permette di raggiungere performance migliori, ad esempio quasi il 94% con una singola rete.

Sono stati usati due dataset LOOCV differenti, uno chiamato "*Day*", e l'altro chiamato "*Bout*". Il primo, con meno fold, è usato per misurare le performance della rete stessa, mentre il secondo, più complicato, è usato per verificare i risultati precedenti. È stato anche calcolato l'*Equal Error Rate* (ERR) delle varie combinazioni, uno standard per le tecnologie di identificazione biometrica, che ci permette di capire le capacità discriminatorie della rete.

Bisogna specificare che i campioni del nostro dataset sono stati ripuliti manualmente da interferenze o rumore, scartando qualsiasi cosa potesse disturbare le reti.

2. Rappresentazioni audio

Le reti utilizzate in questo lavoro sono reti neurali convoluzionali progettate per il riconoscimento di immagini, ma sono anche usate nel riconoscimento e nella classificazione di pattern audio con risultati impressionanti.

Le feature audio utilizzate, in totale 26, sono state fornite direttamente da Matlab, e alcune sono più tradizionali come lo *Spectrogram* o l'MFCC, e altre derivanti da approcci più moderni, come il *Mel Spectrogram*. Molte delle altre tecniche derivano comunque dallo *Spectrogram* o dal *Mel Spectrogram,* e il loro scopo è soddisfare al meglio i bisogni della rete neurale.

Segue la lista completa delle 26 feature audio utilizzate nell'ordine stabilito da Matlab: Vggish, Spectrogram, Mel Spectrogram, Bark Spectrogram, Erb Spectrogram, MFCC, MFCC Delta, MFCC Delta Delta, GTCC, GTCC Delta, GTCC Delta Delta, Spectral Centroid, Spectral Crest, Spectral Decrease, Spectral Entropy, Spectral Flatness, Spectral Flux, Spectral Kurtosis, Spectral Roll off Point, Spectral Skewness, Spectral Slope, Spectral Spread, Pitch, Harmonic Ratio, Zero-cross Rate, Short Time Energy.

Segue ora una lista delle principali e migliori feature usate.

2.1 Spectrogram

La prima rappresentazione presa in considerazione, e probabilmente la più importante, è lo spettrogramma. Questa rappresentazione audio è una delle più conosciute e più utilizzate, e si basa sulla trasformata di Fourier.

Lo spettrogramma mostra come la frequenza del segnale varia nel tempo, ossia come varia la distribuzione dell'energia del segnale nel dominio della frequenza per ogni slice temporale.

La sua rappresentazione deriva dalla *Short Time Fourier Transform* (STFT), che è calcolata, per ogni slice di tempo, estraendo la trasformata di Fourier della convoluzione tra il segnale e un filtro convoluzionale, di solito una funzione gaussiana.

$$STFT\{x(t)\}(t,\omega) = \int_{-\infty}^{\infty} x(\tau) * w(t-\tau)e^{-i\pi t}dt$$

2.2 Mel Spectrogram

La rappresentazione dello spettrogramma mel deriva chiaramente dallo spettrogramma, ma la scala lineare della frequenza viene sostituita da una scala mel. Non esiste una definizione standard della scala mel, ma una formula comunemente ritenuta valida è

$$m = 2595\, log_{10}\left(1 + \frac{f}{700}\right)$$

Questa deriva dagli studi psico-acustici del ventesimo secolo che si focalizzarono sulla correlazione tra la frequenza di un suono e le variazioni di altezza percepite.

2.3 Bark Spectrogram

Come per il caso precedente, una feature di tipo Bark Spectrogram è una rappresentazione derivata dallo spettrogramma, ma con una scala diversa per la frequenza. In questo caso, si tratta della scala Bark, proposta per far corrispondere distanze uguali di un suono a differenze uguali di percezione.

È correlata e simile alla scala mel, ma in genere è meno conosciuta e meno usata. È comunque un'ottima feature.

2.4 MFCC

MFCC è l'acronimo di *Mel-frequency cepstrum coefficients* ed è la rappresentazione di un segnale basata in una rielaborazione dello spettrogramma mel. Nello specifico, una feature MFCC è generata dapprima calcolando la *Short Time Fourier Transform* (STFT) del segnale, applicandola al filtro mel ottenuto dallo spettrogramma mel, e poi prendendo il logaritmo dello spettro e applicando la funzione inversa della trasformata di Fourier al risultato. In formula:

$$MFCC = \left|F^{-1}\left(log\left(mel(|F(S)^2|)\right)\right)\right|^2$$

2.5 Feature VGGish

Con feature VGGish ci si riferisce all'estrazione di *feature vector* generati da una rete VGG pre-addestrata che è stata "tagliata" per accedere ai vettori interni generati dai livelli convoluzionali. Il tipo e le caratteristiche della feature sono definiti da quale specifica parte della rete viene tagliata.

2.6 Short Time Energy

Questa feature è determinata sommando un'unità al quadrato del segnale modificato. Secondo la documentazione Matlab, questa rappresentazione è calcolata proprio con:

$$sTE = sum(xbw^2, 1)$$

dove *xbw* è il segnale modificato per la rete neurale.

2.7 Spectral Slope

Lo Spectral Slope è la pendenza spettrale, anche detto gradiente spettrale, ed è correlato ad altre feature che riguardano lo spettro del segnale audio in questione. Queste sono lo spettrogramma, il mel spettrogramma, il bark spettrogramma, e l'erb spettrogramma.

La pendenza spettrale misura quanto velocemente lo spettro di un pattern audio svanisce andando verso le alte frequenze, ed è calcolata usando la regressione lineare.

2.8 Erb Spectrum

Si tratta di un'altra feature derivante dallo spettogramma, ma a cui viene applicata un altro tipo di scala. ERB è acronimo di *equivalent rectangilar bandwidth,* cioè la larghezza di banda rettangolare equivalente, una scala che fornisce un'approssimazione delle larghezze di banda dei filtri dell'udito umano, usando, come semplificazione di modellazione, dei filtri rettangolari.

3. Reti Neurali Convoluzionali e Transfer Learning

Le reti neurali convoluzionali sono un tipo specifico di reti neurali, sviluppate negli anni 80, ma usate con molto più interesse solo negli anni 10 di questo secolo, dopo una serie di risultati incredibili al *ImageNet Large Scale Visual Nation Challenge*, dove una CNN basata sul GPU-computing vinse la competizione [8]. Quella rete era AlexNet.

Le CNN deviano dal normale funzionamento delle reti neurali in quanto utilizzano tipi diversi di strati, più nello specifico utilizzano strati convoluzionali che applicano una serie di filtri ai dati in input, generando differenti vettori in output per ogni filtro, e un singolo peso per ogni filtro. Minore è il numero di pesi, e meno i filtri sono soggetti a *overfitting*, e migliore è la propensione delle reti a riconoscere informazioni localmente, identificare autonomamente filtri rilevanti e caratteristici per quel determinato compito, sviluppare un'invarianza speciale per i dati in input e ridurre il numero di parametri da ottimizzare. Queste reti sono più veloci, più affidabili e più performanti nel compito di riconoscere l'immagine. [8][9][10][11]

Le reti neurali convoluzionali hanno anche dimostrato un grande successo nel riconoscimento di audio negli ultimi anni, sebbene questo tipo di reti fossero state progettate per l'identificazione di immagini, con un incremento della performance come modelli evoluti, a cui si somma l'uso di feature migliori. [7]

In questo lavoro sono state usate delle reti molto conosciute, ma è stato deciso di non addestrarle da zero. Infatti, per utilizzare al meglio il numero limitato dei nostri pattern e per limitare il tempo di addestramento, è stato usato un approccio chiamato *transfer learning*, nel quale reti pre-addestrate vengono usate per raggiungere grandi risultati con un minimo addestramento. [12] In particolare, l'ultimo segmento delle reti pre-addestrate, il livello *fully-connected* che funge da classificatore, è sostituito con un altro nuovo, più adatto al nuovo problema di classificazione. In questo modo possiamo sfruttare la potenza di questi modelli e il grande dataset (di cui noi non disponiamo) con cui i loro pesi sono stati addestrati, pur rimanendo abbastanza flessibili da risolvere il nostro problema specifico.

I modelli utilizzati in questo lavoro sono quindi AlexNet, VGG-16 e ResNet-50, pre-addestrati nell'ImageNet dataset, dai quali è stato rimosso il livello finale *fully-connected* e sostituito con uno nuovo non ancora addestrato.

Alla fine, è stata introdotta un'altra tecnica per raggiungere il miglior risultato possibile, ossia selezionare alcune delle reti più diverse tra loro, e combinarle in un ensemble; infatti, questi sistemi combinati possono migliorare l'accuratezza.

3.1 AlexNet

AlexNet è una delle più importanti e famose reti neurali convoluzionali degli ultimi anni. La sua celebrità è dovuta alla vittoria dell'ImageNet Challenge nel 2012 con un grande margine, che ha spedito la sua tecnologia dalle stalle alle stelle. È una combinazione di livelli convoluzionali e di max-pooling, ai quali si aggiunge una funzione di attivazione ReLU. Inoltre, AlexNet ha mostrato il grande potenziale dell'addestramento con GPU basato su CUDA, che fa crollare i tempi di addestramento per i suoi 60 milioni di parametri. [8]

3.2 VGG-16

VGG-16 è un altro tipo di CNN sviluppato dall'università di Oxford per ImageNet Challenge del 2014, e ripercorre le orme di AlexNet. Infatti, utilizza una combinazione simile di livelli convoluzionali e di max-pooling con un livello finale *fully-connected* per la classificazione.

Differisce però da AlexNet per la sua dimensione e per la sua complessità, con più di 138 milioni di parametri in totale, dovuti a filtri più piccoli, di dimensione 3x3, e una struttura più profonda con 16 livelli. Questo aumento di complessità influenza negativamente la performance dell'addestramento, del testing e anche del salvataggio delle risorse. Ma questi costi aggiuntivi si riflettono in performance finali più alte. Infatti, questa rete è considerata lo stato dell'arte nel design delle CNN.

3.3 ResNet-50

ResNet è un nuovo tipo di design che cerca di risolvere i problemi derivanti da modelli più nuovi e più profondi [13]: infatti, prima della sua nascita, anche se la tecnologia delle CNN era popolare da poco tempo, un'idea molto comune era: "*Deeper is better!*", ossia "Più i livelli sono profondi, meglio è!" Questa pratica era correlata al fatto che molti modelli raggiungevano grandi performance con livelli più profondi, ma questa moda per fortuna non è durata a lungo, in quanto modelli molto profondi possono annullare o anche ribaltare i guadagni in performance. Tutto questo è dovuto a molti

fattori, compreso il *vanishing gradient* per design molto profondi o per feature troppo astratte da poter essere rappresentate a livelli profondi. [14]

L'innovazione di ResNet, quindi, è stata il "*Residual Block*", cioè un cortocircuito tra i livelli, che permette alle feature di raggiungere i livelli più bassi della rete ancora non addestrata, e la rete può imparare più velocemente. È possibile avere quindi più livelli profondi, al massimo 161, pur mantenendo grandi performance, equiparabili ad una rete molto più leggera. Questo design è anche più facile da addestrare rispetto al modello di VGG-16, e richiede meno tempo computazionale. [15]

3.4 Ensemble

Gli ensemble sono una tecnica fondamentale nel machine learning [16], dove più modelli di reti sono addestrati separatamente e poi i loro risultati sono uniti in qualche modo, in maniera tale che la loro conoscenza collettiva riesca a raggiungere risultati migliori rispetto ad un singolo sistema, o anche rispetto ad un sistema più complesso con più risorse. Questa tecnica si è dimostrata molto efficace, specialmente in sistemi molto complessi, dove un singolo modello molto performante può essere costoso da sviluppare.

Inoltre, l'idea degli ensemble si basa nell'ipotesi che i modelli di reti siano il più indipendenti e il più diversi possibile, affinché la loro conoscenza combinata possa colmare al meglio il problema. Ci sono molte sfaccettature di questa tecnica, da una semplice somma di risultati, a prodotti, medie pesate, modelli Bayesiani avanzati, o addirittura usando un altro sistema di machine learning per creare l'ensemble.

In questo lavoro è stato scelto un approccio semplice, cioè sono stati sommati i risultati raggiunti dai test set ed è stata selezionata la coppia con le migliori performance, il tutto salvando su disco i risultati del training e usando uno script per calcolare le varie coppie e selezionare le migliori. Tuttavia, questa procedura può causare dell'overfitting nei nostri modelli, specialmente a causa del piccolo dataset a nostra disposizione, ma l'utilizzo di molte reti e molte feature diverse ci fa credere nei risultati ottenuti.

4. Dataset e tecniche di Cross Validation

4.1 Dataset

Il dataset utilizzato in questo lavoro è stato ottenuto dallo studio [4], condotto nella Bubye Valley Conservancy, una riserva privata nel sud Zimbabwe, con un'area di 3400 km quadri, sede di molte specie tipiche dell'Africa.

Per quanto riguarda i leoni, questi animali sono al più attivi durante la notte, inclusa la loro attività di richiami e ruggiti, rendendo molto difficile ottenere i dati tradizionalmente tramite registratori, e ancora più difficile associare un ruggito al leone che lo ha emesso. Quindi è stato usato un nuovo approccio, che prevede dei collari biometrici posizionati direttamente sugli animali, composti da un accelerometro, un microfono e un magnetometro. Nella riserva, quindi, sono stati catturati cinque individui maschi e tre individui femmine, a cui sono stati agganciati questi sensori, e poi rilasciati. I sensori hanno un prodotto tra 4 e 10 giorni di dati prima che la batteria si esaurisse, codificati con 8 bit e 16 khz di frequenza. Gli animali sono stati poi il ricatturati e i dati scaricati.

Le registrazioni sono state poi processate manualmente per isolare i ruggiti, visualizzare gli spettrogrammi e catalogare i campioni. In questo modo riconoscere quando un animale stava ruggendo è diventato un compito facile; infatti, i loro ruggiti erano molto più chiari e potenti, ovviamente. Anche i dati dell'accelerometro sono stati di grande aiuto, infatti i leoni compiono un movimento distintivo della testa quando ruggiscono. [4]

4.2 Tecniche di Cross Validation

Il dataset così ottenuto è composto da 164 ruggiti a squarciagola, divisi in 5 maschi. Purtroppo, nessuna femmina ha ruggito durante il periodo di registrazione. Questo piccolo dataset ci spinge ad utilizzare al meglio i dati che abbiamo, generando il test e il training set usando il paradigma *Leave One Out Cross Validation* (LOOCV), con due diversi formati:

Il primo, chiamato "*Day*", riserva come dati di test un singolo giorno di ruggiti a squarciagola, mentre tutti i rimanenti costituiscono il training set. Così, abbiamo un dataset di 20 fold.

L'altro set è stato chiamato "*Bout*", in quanto ogni test set è un singolo raggruppamento di ruggiti, tra uno e tre campioni, e il resto costituisce il training set. Quest'ultimo risulta essere molto più grande, con 78 fold. Si è deciso di non separare i singoli campioni del test set per evitare la forte correlazione

che ruggiti dello stesso *bout* hanno, che potrebbe compromettere l'indipendenza tra il test e il training set.

Gran parte del training e del testing in questo lavoro è stato fatto solo nel dataset *day*, per via del numero di fold minore e quindi della risultante velocità di calcolo. Il dataset *bout* è stato riservato per testare le performance effettive.

4.3 Equal Error Rate

Per validare ulteriormente le performance del nostro modello si è deciso di calcolare l'*Equal Error Rate* (EER) del sistema, una tecnica standard per validare le capacità di discriminazione dei sistemi biometrici. Questo valore si calcola trovando il valore di soglia lungo la curva ROC *(Receiver Operating Characteristic*) dove il *false acceptance rate* e il *false rejection rate* sono uguali. In generale, più basso è l'EER, più preciso è il sistema.

Per calcolarlo, abbiamo bisogno di ristrutturare il cross validation dataset in un problema binario: il cosiddetto "*one-to-many"* design.

Quindi, per ogni fold, è stato selezionato un leone, e i suoi ruggiti sono stati taggati come classe 1. Tutti gli altri leoni sono stati messi nella classe 2. È stata mantenuta la tecnica del *leave one out*, mantenendo un singolo *day* o *bout* di dati da ogni classe per il testing e tutti i rimanenti dati per il training. Questo ci dà di nuovo due dataset EER, un "*EER Day*" e un "*EER Bout*".

Entrambe le versioni sono state usate in tutte le combinazioni di reti per validare i risultati.

5. Risultati

In questo lavoro sono state utilizzate 26 rappresentazioni di feature diverse, testando metodi di estrazione differenti, e dando in pasto ciascuna feature ad ogni modello di CNN. Sono state provate, infatti, per ciascuno dei tre modelli di reti, tutte le 26 feature a disposizione, ma si riportano in seguito solo i risultati migliori. Come specificato prima, tutti i test sono stati eseguiti nel dataset *day*, che consiste in 20 fold per il training test, mentre il dataset *bout* è stato utilizzato solo per validare i risultati ottenuti. Bisogna dire che i risultati derivanti dal dataset *bout* sono sicuramente afflitti da un po'di overfitting, dovuto al piccolo dataset a nostra disposizione, spesso di un solo campione per fold. Sebbene questi ultimi non possano misurare con precisone le performance del sistema, mostrano comunque che i risultati del dataset *day* (molto più realistici) non sono accidentali.

Segue una tabella delle migliori combinazioni di reti e feature:

Rete	**Feature**	**Accuratezza *Day***	**Accuratezza *Bout***
ResNet-50	Bark Spectrogram	93.5%	95%
ResNet-50	Spectrogram	91.9%	95%
VGG-16	Short Time Energy	91.6%	96%
VGG-16	Spectrogram	91.1%	92%
VGG-16	MFCC	90.1%	93%
ResNet-50	Spectral Slope	89.1%	92%
AlexNet	Short Time Energy	87.3%	90%
ResNet-50	Erb Spectrogram	85.7%	89%

5.1 Risultati EER

I risultati EER sono stati usati per validare l'abilità del sistema a discriminare i campioni audio come descritto prima. Segue la tabella dei risultati EER per entrambi i dataset *day* e *bout* per le migliori combinazioni di reti e feature.

Rete	Feature	*EER Day*	*EER Bout*
ResNet-50	Bark Spectrogram	7.5	2.1
ResNet-50	Spectrogram	6.97	1.9
VGG-16	Short Time Energy	9.29	4.2
VGG-16	Spectrogram	7.27	2.3
VGG-16	MFCC	11.6	6.0
ResNet-50	Spectral Slope	4.9	3.2
AlexNet	Short Time Energy	11.52	8
ResNet-50	Erb Spectrogram	8.61	3.9

È possibile, inoltre, andare ad aumentare questi risultati utilizzando un ensemble di due o tre reti, con una combinazione di feature diverse tra loro, in maniera tale da rendere la combinazione più indipendente possibile. Con questa tecnica, i risultati aumentano fino al 96% con un ensemble di due reti, e fino al 97% con tre reti.

6 Discussione

Si nota subito, dai primi due risultati migliori, che la rete con maggior successo nel nostro lavoro è ResNet-50, sebbene la tecnologia della rete VGG-16 sia considerata oggi lo stato dell'arte. Ciò è dovuto al tipo di rete in questione, che ha il giusto peso, né troppo profonda, né troppo leggera, per riuscire ad adattarsi bene al problema di identificazione con i 164 pattern di cui noi disponiamo. Quasi sicuramente, se il dataset a nostra disposizione fosse stato più corposo e variegato, la rete che avrebbe riscosso il miglior risultato sarebbe stata VGG-16. Bisogna anche specificare, però, che il successo di una rete rispetto ad un'altra dipende anche dal tipo di feature utilizzate. Alcune feature, infatti, non sono adeguate tanto quanto altre all'architettura della rete a cui vengono sottoposte.

Comunque, rispetto al lavoro originale [4], ci sono stati degli avanzamenti nei risultati. Originariamente, al posto delle nostre feature, venivano utilizzati i modelli di Markov nascosti, che raggiungevano nello stesso dataset un'accuratezza del 90%. Ora, senza addirittura usare un ensemble, riusciamo a raggiungere quasi il 94% con una singola rete. Questo ci fa riflettere ancora sulla reale praticità e potenza dell'architettura CNN. Dobbiamo però tener conto che questo lavoro è stato fatto con dei pattern ripuliti a mano da rumore e interferenze, e quindi particolarmente favorevoli. Inoltre, il dataset a nostra disposizione era particolarmente piccolo e limitato per quanto riguarda la varietà dei pattern.

Inoltre, nel nostro lavoro, non è stato aggiunto alcun pattern derivante da tecniche di *data augmentation*, in quanto si poteva notare una variazione degli score minima, e un aumento del tempo di training significativo. Non disponendo di grandi risorse computazionali, si è preferito usare il dataset originale del lavoro [4] senza servirsi di *data augmentation*. Tutti gli addestramenti in questo lavoro sono stati eseguiti tramite una scheda video Nvidia GTX 1060 di un personal computer.

Bibliografia e sitografia

[1] Convolutional Neural Networks for identification of African Lions from individual vocalizations, Martino Trapanotto, Loris Nanni, Gianluca Maguolo

[2] Clemins, Johnson, Leong, Savage, Automatic classification and speaker identification of african elephant (loxodonta africana) vocalizations, The Journal of the Acoustical Society of America, 2005

https://doi.org/10.1121/1.1847850

[3] Ji, Johnson, Walsh, McGee, Armstrong, Discrimination of individual tigers (panthera tigris) from long distance roars, The Journal of the Acoustical Society of America, 2013

https://doi.org/10.1121/1.4789936

[4] Wijers, Trethowan, Preez, Chamaillé-Jammes, Loveridge, Macdonald, Markham, Vocal discrimination of african lions and its potential for collar-free tracking, 2020

https://doi.org/10.1080/09524622.2020.1829050

[5] Spillmann, van Schaik, Setia, Sadjadi, Who shall i say is calling? Validation of a caller recognition procedure in bornean flanged male orangutan (pongo pygmaeus wurmbii) long calls, 2017

https://doi.org/10.1080/09524622.2016.1216802

[6] McComb, Packer, Pusey, Roaring and numerical assessment in contests between groups of female lions, panthera leo, Animal Behaviour, 1994

https://doi.org/10.1006/anbe.1994.1052

[7] Hershey, Chaudhuri, Ellis, Gemmeke, Jansen, Moore, Plakal, Platt, Saurous, Seybold, Slaney, Weiss, Wilson, Cnn architectures for large-scale audio classification, 2017

https://doi:10.1109/ICASSP.2017.7952132

[8] Krizhevsky, Sutskever, Hinton, Imagenet classification with deep convolutional neural networks, 2017

https://doi:10.1145/3065386

[9] Chauhan, Ghanshala, Joshi, Convolutional neural network (cnn) for image detection and recognition, 2018

https://doi:10.1109/ICSCCC.2018.8703316

[10] Tianyu, Zhenjiang, Jianhu, Combining cnn with hand-crafted features for image classification, 2018

https://doi:10.1109/ICSP.2018.8652428

[11] Simonyan, Zisserman, Very deep convolutional networks for largescale image recognition (2015)

[12] Hedjazi, Kourbane, Genc, On identifying leaves: A comparison of cnn with classical ml methods, 2017

https://doi:10.1109/SIU. 2017.7960257

[13] Srivastava, Greff, Schmidhuber, Highway networks (2015)

[14] He, Zhang, Ren, Sun, Deep residual learning for image recognition, 2016.

[15]
https://github.com/Kulbear/deeplearningcoursera/blob/master/Convolutional%20Neural%20Networks/Residual%20Networks% 20-%20v1.ipynb

[16] Rokach, Ensemble-based classifiers, Artificial Intelligence Review 33, 2010

https://doi.org/10.1007/s10462-009-9124-7