UNIVERSITÀ DEGLI STUDI DI PERUGIA
Dipartimento di Matematica e Informatica

Corso di Laurea in Informatica

Tesi di Laurea

Adversarial Machine Learning for face recognition

Laureando:

Fabrizio Fagiolo

Relatori:

Dr. Valentina Poggioni

Dr. Alina Elena Baia

Anno accademico 2020/2021

Indice

Sommario

Le alte prestazioni dei sistemi basati su reti neurali profonde hanno attratto molte applicazioni nel riconoscimento di oggetti e nel riconoscimento facciale. Questa potenza presenta però anche degli svantaggi. Le reti sono altamente sensibili ad attacchi e quindi, tendono ad essere inaffidabili e mancano di robustezza. Tali attacchi avversari, visivamente impercettibili, possono avere un effetto fuorviante sul riconoscimento facciale e possono causare un decadimento delle prestazioni del sistema. Per difendersi dagli attacchi avversari, vengono utilizzati sofisticati approcci di rilevamento e mitigazione.

In questa tesi verranno presentate alcune delle strategie di attacco, difesa e mitigazione piu' utilizzate per il riconoscimento facciale. Inoltre verrà presentato un toolbox per l'implementazione di queste strategie e verranno riportati i risultati sperimentali che attestano l'efficacia delle tecniche proposte nel toolbox.

Capitolo 1

1 Introduzione adversarial machine learning

1.1 Cos'e' l'adversarial machine learning

Negli ultimi anni si è assistito a un rapido aumento dell'uso dell'apprendimento automatico, attraverso il quale i computer possono essere programmati per identificare i modelli e fare previsioni sempre più accurate nel tempo.
L'apprendimento automatico è una tecnologia fondamentale alla base dell'intelligenza artificiale (AI) e viene utilizzata per applicazioni preziose come i filtri antispam, il rilevamento di malware, nonché per tecnologie più complesse come il riconoscimento vocale, il riconoscimento facciale, la robotica e le auto a guida autonoma. Sebbene i modelli di apprendimento automatico presentino molti vantaggi, possono essere vulnerabili. I ricercatori della sicurezza informatica si riferiscono a questo processo come "adversarial machine learning", poiché i sistemi di intelligenza artificiale possono essere ingannati (da aggressori o "avversari") facendogli fare valutazioni errate. Un attacco avversario può compromettere le prestazioni di un modello di apprendimento automatico sia durante l'addestramento sia con l'introduzione di dati progettati in modo dannoso per ingannare un modello già addestrato, sempre con lo scopo di fargli commettere errori.

1.2 Come ingannare una rete neurale

Alcuni modelli di machine learning già utilizzati nelle applicazioni pratiche potrebbero essere vulnerabili ad alcuni attacchi. Ad esempio, posizionando alcuni piccoli adesivi sul terreno in un incrocio, i ricercatori hanno dimostrato che potrebbero far sì che un'auto mobile a guida autonoma emetta un giudizio anomalo e si sposti nella corsia opposta del traffico. [1]
I ricercatori hanno dimostrato anche che modificando solo un pixel in un'immagine è possibile ingannare gli algoritmi di deep learning per la classificazione di immagini. [17] E' stato dimostrato anche che l'immagine di un cane opportunamente modificata puo' essere classificata come un gatto. [1]
L' Apprendimento automatico avversario comprende una serie di tecniche volte a compromettere il corretto funzionamento di un sistema informatico che fa uso di algoritmi di apprendimento automatico. L'inganno avviene tramite la costruzione di input speciali in grado di raggirare tali algoritmi. Nello specifico, lo scopo di tali tecniche è quello di causare la miss-classificazione. La maggior parte delle

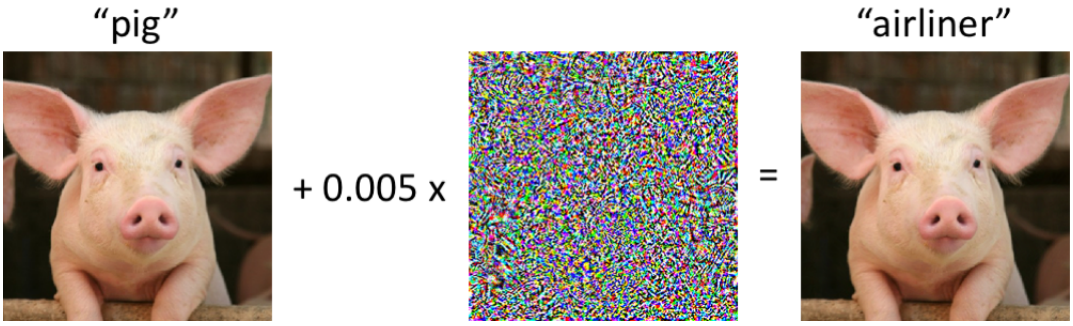

Immagine tratta da:

https://www.notizie.ai/la-difesa-della-privacy-ai-tempi-dellintelligenza-artificiale/

Figura 1: La prima immagine da sinistra rappresenta la classificazione corretta del maiale. Le 2 figure successive rappresentano l'aggiunta del rumore con la conseguente classificazione errata della foto.

tecniche è stata progettata per funzionare su insiemi di problemi specifici in cui i dati di addestramento e di test sono generati dalla stessa distribuzione statistica. Quando questi modelli vengono applicati al mondo reale, gli avversari possono fornire dati che violano tale presupposto statistico.
All'interno del documento tratteremo il problema dell'adversarial machine learning per quanto riguarda il riconoscimento facciale.

Capitolo 2

2 Introduzione al riconoscimento facciale

Il riconoscimento facciale (FR) è una tecnologia biometrica principalmente utilizzata per l'autenticazione dell'identità ed è ampiamente utilizzata in diversi settori, come la finanza, l'esercito, la sicurezza pubblica e la vita quotidiana. L'obiettivo finale di un tipico sistema FR è identificare o verificare l'identità di una persona da un'immagine digitale o da un fotogramma video.

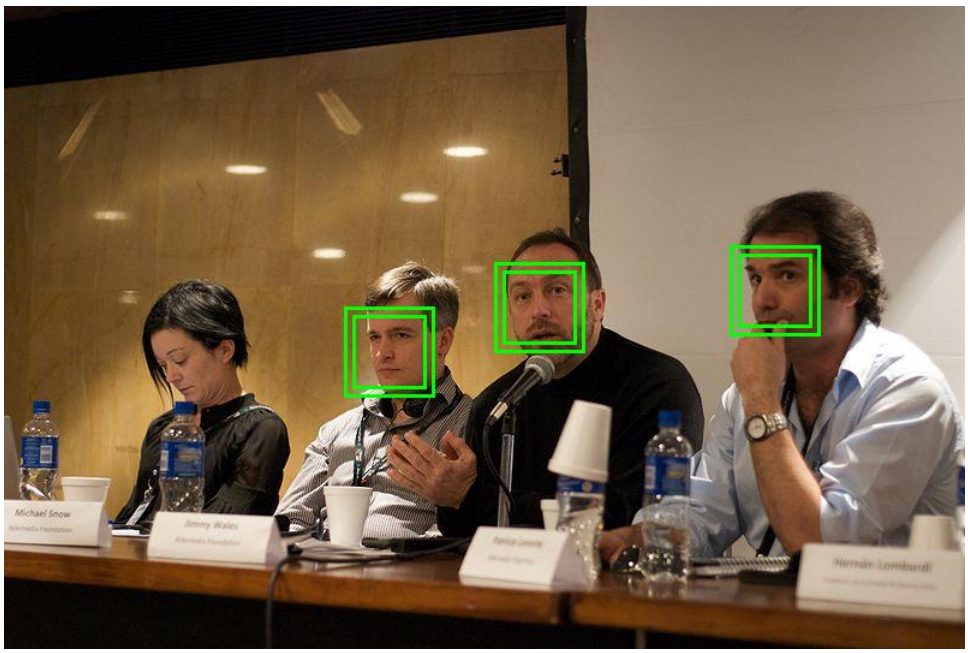

Figura 2: Face detection

Immagine tratta da:

"https://it.wikipedia.org/wiki/File:Face detection"

Il riconoscimento facciale e' basato sull'intelligenza artificiale e può identificare una persona esclusivamente attraverso l'analisi dei modelli e delle caratteristiche facciali dell'individuo. Solitamente il riconoscimento avviene mediante tecniche di elaborazione digitale delle immagini, ignorando tutto quello che non rappresenta una faccia, come edifici, alberi, corpi. che vengono solitamente definiti in un pre-processo.

Per facilitare l'individuazione di una faccia, i primi sistemi tenevano conto del fatto che un viso umano è composto da due occhi, un naso e una bocca. I sistemi più recenti invece riescono a riconoscere una persona anche se questa ha il viso ruotato, o comunque non in visione frontale, oppure indossa occhiali o se le condizioni di illuminazione non sono ottime.

2.1 Principali architetture utilizzate

Con l'evoluzione della ricerca nell'intelligenza artificiale, sono state create varie architetture per il riconoscimento facciale.
DeepFace [16] è stata la prima architettura profonda presentata alla comunità del FR, ha un profondo archivio ed e' realizzata con una rete neurale convolutiva (CNN) con diversi strati collegati localmente. In seguito sono state poi sviluppate FaceNet [13] e VGG-Face [11] basate sul deep learning. Solo successivamente sono state introdotte le reti GoogleNet [15] e VGGNet [14] che, a differenza delle altre, sono in grado di analizzare set di volti su larga scala. In seguito è stata introdotta la SphereFace [9] che è stata proposta secondo un'architettura ResNet [8] la quale utilizza una nuova perdita angolare softmax con caratteristiche che si differenziano dal margine di errore angolare. Simile a questa architettura, sono state poi introdotte CosFace [18] e ArcFace [5] basate rispettivamente sulla funzione di perdita data dal coseno e sul margine di perdita angolare.
Sono state proposte anche reti piu' leggere per sopperire alla mancanza di potenza di GPU e alla dimensione della memoria diventando applicabili a molti cellulari e dispositivi embedded [19].
Queste archietteture vengono chiamate LightCNN [19], ed hanno una funzione di attivazione max-feature-map (MFM) la quale riduce molto il costo computazionale dell'architettura.

2.2 Principali set di dati utilizzati

Per addestrare i modelli alla base del riconoscimento facciale possono essere usati vari set di dati.
Avere set di dati di allenamento sufficientemente grandi per valutare l'efficacia dei modelli FR profondi ha portato allo sviluppo di set di dati più complessi per facilitare ricerca la del FR.
Sotto vengono riportati alcuni dei piu' utilizzati.
LFW (label faces in the wild) è composto da 3K immagini di volti presi dal web in condizioni di luminosita' diverse. LFW ha cosi' aperto un nuovo percorso per l'utilizzo di altri database di testing per compiti diversi. [17]

I primi modelli FR profondo, come DeepFace, FaceNet e DeepID sono stati addestrati con set di dati di addestramento controllati su piccola scala [17].
Successivamente sono stati introdotti i primi set di dati su larga scala come, CASIA-Webface, una raccolta di 0,5 milioni di immagini di 10K celebrità: è stato presentato come il primo pubblico ampiamente utilizzato come set di dati per l'addestramento [17].
In seguito la ricerca ha portato allo sviluppo di MS-Celeb-1M, VGGface2,e Megaface, raccolte di oltre 1 milione di immagini che venivano utilizzate in molti metodi avanzati di deep learning. [17]

2.3 Principali attacchi di FR

Gli attacchi si dividono in:

- attacchi white-box che presuppogono la completa conoscenza del modello di target, ovvero i suoi parametri, l'architettura, il metodo di formazione e, in alcuni casi, anche i suoi dati di addestramento.

- Attacchi black-box alimentano un modello bersaglio con attacchi avversari(durante il test) creati senza conoscere il modello (ad esempio, la sua procedura di addestramento o la sua architettura).

La specificità dell'avversario è definita come la capacità dell'attacco di consentire una specifica intrusione/disturbo [17].
I modelli di minaccia nei sistemi FR profondi potrebbero essere classificati nelle seguenti tipologie a seconda della specificità dell'attacco.

- L'attacco mirato inganna un modello inducendolo a prevedere erroneamente una specifica etichetta.

- L'attacco non mirato predice gli esempi avversari e le etichette in modo irrilevante, purché i risultati non siano etichette corrette. Un attacco non mirato è più facile da implementare rispetto a un attacco mirato poiché si hanno più scelte e lo spazio per modificare l'output e' maggiore.

Lo scopo degli attacchi e' diminuire le prestazioni su un modello di classificazione. Successivamente verrano riportati alcuni degli attacchi ai sistemi di FR piu' utilizzati.
FGSM("Fast Gradient Method) scoperto nell'anno 2014 e' uno dei primi attacchi scoperti per il face recognition.
Questo attacco calcola il gradiente della funzione di perdita del modello relativo al vettore immagine per ottenere la direzione di cambiamento del pixel.[7]

In seguito (2015) viene implementato DeepFool che è una tecnologia di attacco non mirata.
Calcola iterativamente il rumore minimo in grado di ingannare il sistema. Il rumore minimo è definito come l'orto-distanza più vicina all'iper-piano, assumendo che il classificatore sottostante sia lineare. [10]
Un altro attacco molto efficace e' CARLINI & WAGNER (C&W)L2. [3] Questo attacco è uno dei più forti attacchi avversari attualmente disponibili per il face recognition. Può essere utilizzato in forme sia mirate che non mirate.
Questo metodo tenta di ridurre al minimo la seguente equazione per generare immagini avversarie.

$$\min \|\frac{1}{2}(tanh(w)+1) - x\|_2^2 + c \cdot f(\frac{1}{2}(tanh(w)+1))$$

Dove w e' l'immagine avversaria che minimizza l'espressione.
f(x)=max(max $\{Z(x)i : i = t\} - Z(x)t, -k)$
Z è il livello logit, t è la classe di destinazione, K è il parametro che controlla l'affidabilità della classificazione errata.
Infine viene sviluppato EAD: ("Elastic-Net Attacks to Deep Neural Networks") [4]
L'attacco EAD-EN (Elastic Net) tenta di trovare esempi avversari che possono ingannare la rete neurale riducendo al minimo le metriche e la distanza della perturbazione.
Siano (x_{orig}, t_{orig}) che e' la coppia originale immagine-etichetta e (x_{adv}, t_{adv}) che e' la coppia avversaria immagine-etichetta. Per creare esempi avversari quindi la formula utilizzata da questo metodo di attacco è:

$$\min_{x_{adv}} \alpha \cdot f(x_{adv}, t_{adv}) + \beta \cdot \|x_{adv} - x_{orig}\|_1 + \|x_{adv} - x_{orig}\|_2^2$$

$$\textit{subject to } x_{adv} \in [0,1]^d$$

dove, α e β sono i parametri che regolano la grandezza della rete.
$f(x_{adv}, t_{adv})$ è la funzione di perdita.
Il metodo di attacco consiste nel generare un'immagine attaccante classificata nella classe target t_{adv} andando a minimizzare la funzione di perdita della rete.
EAD puo' basarsi su una funzione di perdita che utilizza una combinazione lineare di penalità L1. In questo caso assume il nome EAD-L1.

2.4 Principali strategie di difesa

Poiché i nuovi approcci per creare attacchi sono molto efficaci, fra gli obiettivi dei ricercatori in ambito adversarial machine learning c'e' anche quello di creare strategie di difesa sufficientemente efficaci in modo da poterli contrastare. [17]. Sono state definite quindi diverse strategie di difesa per aumentare la sicurezza dei sistemi di FR, e potrebbero essere organizzate nel modo seguente.
La conservazione dell'architettura del modello è una considerazione primaria quando si costruiscono tecniche di difesa contro attacchi avversari, [6] l'obiettivo e' alterare il meno possibile l'architettura di un modello.
Poi abbiamo il mantenimento dell'accuratezza che è un fattore considerato fondamentale per mantenere quasi inalterati i risultati della classificazione.
La conservazione della velocità del modello è un altro fattore che non dovrebbe essere influenzato durante la fase di test. Le metodologie descritte precedentemente sono strategie generali per difendersi dai principali attacchi avversari.
I metodi di difesa piu' specifici possono essere suddivise in tre categorie:

- Alterazione dell' addestramento durante l'apprendimento.
  Ad esempio, iniettando esempi avversari in dati di addestramento o incorporando input alterati durante i test. [17].
- Cambiando le architetture di rete.
  Ad es. numero di livelli, sotto-reti, funzione di perdita e attivazione [17].
- Integrando il modello primario con reti esterne durante il processo di classificazione. [17]

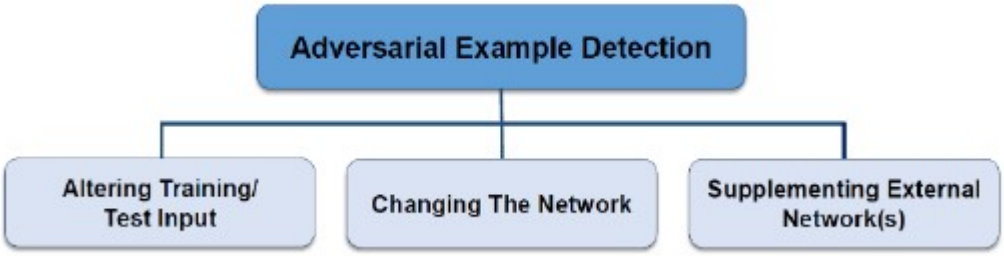

Immagine tratta da:

"Adversarial Attacks Against Face Recognition: A Comprehensive Study"

Figura 3: Una categorizzazione generale dei metodi di rilevamento finalizzati a difendere i sistemi FR da attacchi avversari [17]

3 Progetto Smart Box

SmartBox è un toolbox basato su Python che fornisce un'implementazione open-source di algoritmi di attacco, rilevamento, e mitigazione di attacchi avversari.
In questa ricerca il dataset utilizzato e' Yale Face Database. Il dataset è stato utilizzato per testare vari algoritmi di attacco come DeepFool, FGSM, e C&WL2. SmartBox fornisce una piattaforma per valutare i nuovi attacchi, modalità di rilevamento e approcci di mitigazione su un riconoscimento del volto.
Il codice del progetto e' disponibile su Github `https://github.com/akhil15126/SmartBox`. SmartBox è composto da tre grandi moduli: generazione di attacchi, rilevamento e mitigazione.[6]

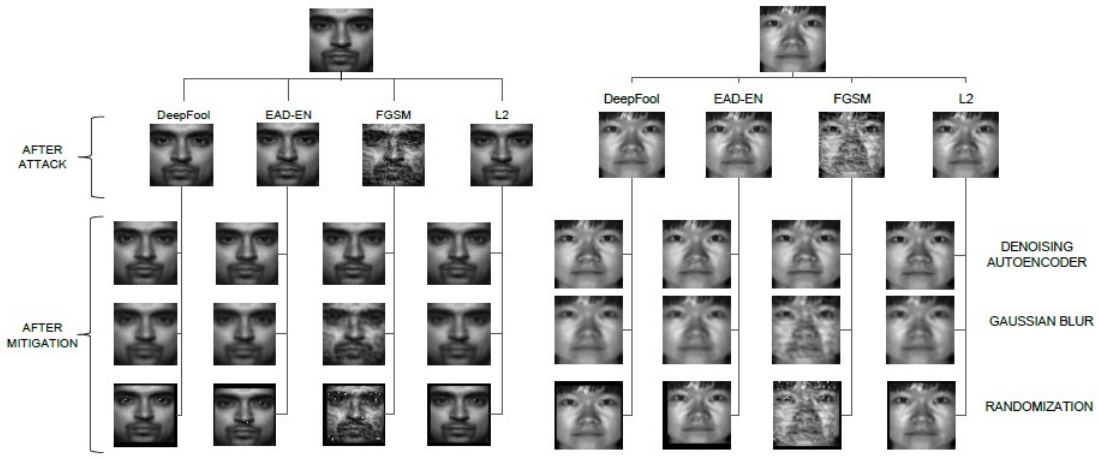

Immagine tratta da:

"https://ieeexplore.ieee.org/document/Smart-Box"

Figura 4: Effetti della generazione attacchi avversari e degli algoritmi di mitigazione su due esempi facciali (a) e (b). [6]

3.1 Modulo di generazione degli attacchi

SmartBox include l'implementazione di algoritmi di attacco mirati e non mirati per generare immagini avversarie da un insieme di immagini di input.
Lo scopo è quello di diminuire le prestazioni di un modello di classificazione. Gli attacchi che vengono presentati all'interno del modulo sono quelli visti in precedenza.

- FGSM("Fast Gradient Method") [7].
- DeepFool [10].
- CARLINI & WAGNER (C&W)L2 [3].
- EAD-EN("Elastic-Net Attacks to Deep Neural Networks"). [4]

3.2 Modulo di mitigazione

Il modulo di mitigazione nello SmartBox ha metodi che cercano di individuare le perturbazioni nelle immagini fornite.
Di seguito abbiamo alcune modalità implementate all'interno del progetto.
Il primo sistema presente all'interno della sezione e' la formazione "avversaria".[6] Nella formazione avversaria un nuovo modello viene addestrato utilizzando il set di dati originale ed esempi avversari con le etichette corrette. Dietro a questa tecnica otteniamo che il nuovo modello è robusto per le immagini avversarie.
Successivamente abbiamo la Randomizzazione.[6] In questo approccio le immagini vengono prima campionate e poi sotto-campionate con una dimensione casuale. Le immagini sotto-campionate sono riempite con degli zeri. I fotogrammi sono riempiti in modo tale che la dimensione di un'immagine "imbottita" (di zeri) sia uguale alla dimensione originale dell'immagine. Dalle immagini ottenute dopo il riempimento, ne viene presa una in modo randomico, selezionata e passata al modello. L'interpolazione utilizzata in questo approccio rimuove il rumore creato dall'immagine avversaria.
Un altro metodo di mitigazione e' la sfocatura gaussiana [6]. Questa è una tecnica efficace di elaborazione di un'immagine che offusca e riduce i dettagli complessi di una fotografia in input, concatenando l'immagine con il kernel i cui pesi sono derivati dalla distribuzione gaussiana.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Qui, μ è la media.
σ è la deviazione standard di distribuzione.
L'intuizione dietro la sfocatura gaussiana, proposta come tecnica di mitigazione, e' data dal fatto che la convoluzione con un kernel con una deviazione ragionevole lungo gli assi puo' comportare la diminuzione dell'effetto delle perturbazioni [6].
Infine abbiamo il Denoising Autoencoder[6]. Questo metodo apprende i pesi di un autoencoder denoising tramite esempi di formazione avversaria. Durante l'allenamento, l'autoencoder impara a ricreare le immagini originali dalle immagini perturbate.

3.3 Modulo di rilevamento

Questo modulo ha implementazioni di vari metodi di rilevamento delle perturbazioni che mira a rilevare le immagini false prima di passarle alla rete.
A seguire abbiamo i metodi di rilevamento inclusi in SmartBox:
rilevamento che utilizza le statistiche del filtro di convoluzione [6]. Con questo metodo estraiamo le caratteristiche che includono i coefficienti PCA ("Principal Component Analysis") normalizzati e i valori di minimo e massimo dall'output degli strati della convoluzione, alimentando un classificatore a cascata per il rilevamento avversario.
Il classificatore a cascata è costituito da una sequenza di classificatori di base che vengono poi utilizzati per rilevare immagini avversarie da un insieme di immagini di input.
Oltre a questa tecnica abbiamo il rilevamento basato su PCA [6]: questo metodo, calcola prima la matrice di proiezione dove applica l'analisi delle componenti principali sui dati di allenamento.
La matrice di proiezione viene quindi utilizzata per proiettare le immagini sullo spazio lineare. Le caratteristiche ottenute attraverso questa tecnica sono utilizzate per addestrare un classificatore SVM ("Support Vector Machine").
Oltre a questi abbiamo il metodo di apprendimento degli artefatti [6]. Questo metodo utilizza le funzionalità apprese dal modello per distinguere le immagini originali e avversarie. Nello specifico questo metodo riesce a distinguere la formazione originale e la formazione "avversaria" delle immagini passate attraverso la rete addestrata. Dopo di che le caratteristiche corrispondenti ai livelli desiderati vengono recuperate e viene addestrato un nuovo classificatore binario.
Un'altra strategia utilizzata e' la riduzione adattiva del rumore [6]: a differenza del metodo precedente, questa tecnica, è indipendente dalla natura del rumore aggiunto. E' basata sull'entropia dell'immagine (ovvero la quantità di informazioni contenuta in un'immagine) e le tecniche applicate per modificare l'immagine sono la quantizzazione scalare e i filtri spaziali "smoothing" (di livellavemento).
Un'immagine è identificata come avversario se la modifica cambia la sua classificazione.

4 Risultati ottenuti

Gli esperimenti sono stati condotti sullo Yale Face Database. Il set di dati è costituito da immagini ritagliate con volti di persone di età diversa [6].
I soggetti sono 38 individui rappresentati in diverse condizioni di illuminazione.
Il set di dati è suddiviso in training (80%), validazione (10%) e test (10%).
Ad ogni "attacco" e' assegnato randomicamente un bersaglio ed e' classificato casualmente come "originale" o impostore. Gli esperimenti sono stati condotti con una rete con le seguenti specifiche:
2 livelli di convoluzione ciascuno con 16 filtri e un kernel 3x3 e il livello maxpool con passi 2 impostati . Inoltre abbiamo un livello convuluzionale con 32 filtri e kernel 3x3, un livello di maxpool con passi altri 2 passi. In seguito un livello convoluzionale con 32 filtri e kernel 3x3, un livello maxpool con passi impostati come 2. Abbiamo poi uno strato denso con 100 unità, il Dropout con tasso di caduta del 50%; uno strato denso con 100 unità e un altro livello denso con 1024 unità; infine un livello di logits [6].
Il modello è stato addestrato utilizzando la discesa del gradiente stocastico.
La discesa stocastica del gradiente ("Stochastic Gradient Descent", SGD) è un metodo iterativo per l'ottimizzazione di funzioni differenziabili, ad ogni iterazione, sostituisce il valore esatto della funzione di costo del gradiente con una stima data dalla valutazione del gradiente solo su un sotto-insieme di addendi. È ampiamente usato per l'allenamento di una varietà di modelli probabilistici e modelli di apprendimento automatico.

Nella tabella sottostante [6] vengono rappresentati i risultati degli effetti della generazione di attacchi e degli algoritmi di mitigazione sulla verifica del volto e sull'accuratezza dell'identificazione.

Attacks	Mitigation Algorithm	Verification			Identification		
		Before Attack	After Attack	After Mitigation	Before Attack	After Attack	After Mitigation
DeepFool	Adversarial Training	96.96%	50%	**98.48%**	95.07%	3.4%	**97.34%**
	Gaussian Blur			93.56%			86.74%
	Randomization			73.48%			45.07%
	Denoising AutoEncoder			93.18%			87.50%
EAD-L1	Adversarial Training		0%	**98.86%**		0%	**97.72%**
	Gaussian Blur			92.80%			85.22%
	Randomization			68.56%			38.25%
	Denoising AutoEncoder			93.93%			85.98%
EAD-EN	Adversarial Training		0%	**99.24%**		0%	**99.24%**
	Gaussian Blur			92.04%			83.33%
	Randomization			68.56%			36.36%
	Denoising AutoEncoder			93.18%			85.60%
FGSM	Adversarial Training		67.04%	79.92%		29.92%	57.19%
	Gaussian Blur			67.42%			32.57%
	Randomization			57.95%			18.18%
	Denoising AutoEncoder			**88.25%**			**75.37%**
L2	Adversarial Training		0%	**99.62%**		0%	**99.62%**
	Gaussian Blur			92.80%			85.60%
	Randomization			70.07%			39.39%
	Denoising AutoEncoder			93.18%			85.98%

Immagine tratta da:

"https://ieeexplore.ieee.org/document/Smart-Box"

I risultati riportati sono stati ricavati da test eseguiti sul modello descritto in precedenza.
La tabella riportata sotto descrive invece le prestazioni degli algoritmi di rilevamento degli avversari disponibili in SmartBox, in termini di precisione, recall e accuratezza (%) [6].

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Accuracy = \frac{CorrectPrediction}{TotalNumberOfExample} = \frac{TruePositive + TrueNegative}{TotalNumberOfExample}$$

Attacks	Algorithms	Precision	Recall	Accuracy
DeepFool	Adapt. Noise Reduction	77.48%	93.22%	83.06%
	Artifact Learning	**84.33%**	**95.83%**	**89.01%**
	Conv Filter	52.13%	64.77%	52.65%
	PCA	58.84%	61.74%	59.28%
EAD-L_1	Adapt. Noise Reduction	78.41%	**98.40%**	85.65%
	Artifact Learning	**89.27%**	97.72%	**92.99%**
	Conv Filter	56.77%	74.62%	58.90%
	PCA	61.06%	56.43%	60.22%
EAD-EN	Adapt. Noise Reduction	78.48%	**98.80%**	85.85%
	Artifact Learning	**86.44%**	96.59%	**90.71%**
	Conv Filter	53.79%	64.39%	54.54%
	PCA	56.86%	54.92%	56.62%
FGSM	Adapt. Noise Reduction	55.17%	46.24%	54.33%
	Artifact Learning	89.41%	**86.36%**	**88.06%**
	Conv Filter	**95.54%**	73.10%	84.84%
	PCA	95.42%	63.25%	80.11%
L_2	Adapt. Noise Reduction	78.54%	**99.20%**	86.05%
	Artifact Learning	**85.04%**	96.96%	**89.96%**
	Conv Filter	54.19%	63.63%	54.92%
	PCA	56.55%	57.19%	56.62%

Immagine tratta da:

"https://ieeexplore.ieee.org/document/Smart-Box"

Dalla tabella [6], è evidente che la formazione avversaria si comporta costantemente meglio degli algoritmi rimanenti, essendo stato esposto sia ai dati originali che ai dati di addestramento avversari.
La sfocatura gaussiana attenua efficacemente le perturbazioni prodotte da CARLINI & WAGNER (C&W)L2, EAD-EN e Deep-Fool e fallisce negli attacchi basati sulla discesa del gradiente.
Il Denoising autoencoder invece è effettivamente in grado di mitigare le perturbazioni prodotte da tutti gli algoritmi di generazione di attacchi testati.
L'algoritmo di randomizzazione, d'altra parte, lavora in modo scadente rispetto agli altri algoritmi di attacco [6].
Questa analisi è supportata dalla curva caratteristica di funzionamento del ricevitore e dalla curva caratteristica della corrispondenza cumulativa.

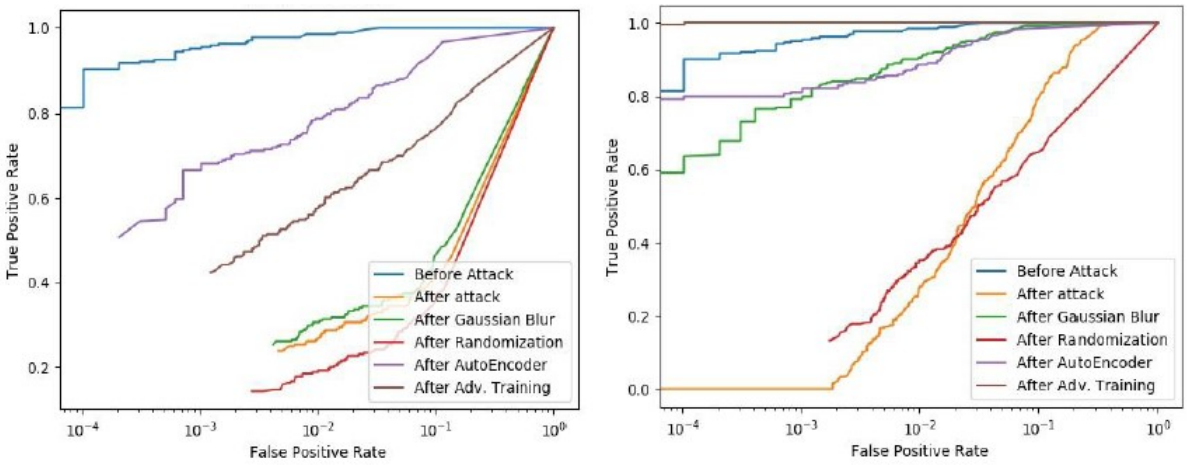

Immagine tratta da:

"https://ieeexplore.ieee.org/document/Smart-Box"

Figura 5: Curve ROC che riassumono le prestazioni con altri algoritmi di attacco sulla classificazione. [6]

Le curve ROC (Receiver Operating Characteristic, anche note come Relative Operating Characteristic) sono degli schemi grafici per un classificatore binario. Lungo i due assi si possono rappresentare rispettivamente i True Positive Rate (TPR, frazione di veri positivi) e False Positive Rate (FPR, frazione di falsi positivi). In altre parole, si studiano i rapporti fra "allarmi" veri (hit rate) e "falsi" allarmi.
La curva ROC viene creata tracciando il valore del True Positive Rate (TPR, frazione di veri positivi) rispetto al False Positive Rate (FPR, frazione di falsi positivi) con l'aggiunta di varie impostazioni di soglia [2].
Gli algoritmi di attacco utilizzati nella figura sovrastante, da sinistra a destra: FGSM, CARLINI & WAGNER (C&W)L2.

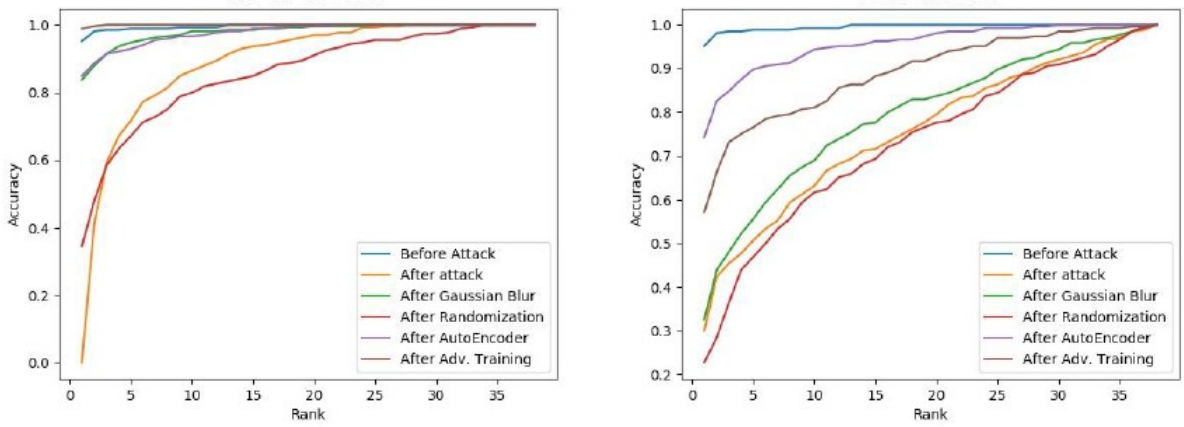

Immagine tratta da:

"https://ieeexplore.ieee.org/document/Smart-Box"

Figura 6: Curve CMC ("Comulative Match Curve") che riassumono la misura della prestazione del sistema di identificazione dopo diversi algoritmi di attacco. [6]

Nella figura sovrastante invece abbiamo le Curve CMC ("Comulative Match Curve") che riassumono il riconoscimento facciale con diversi algoritmi di attacco. La curva (CMC) di corrispondenza cumulativa viene utilizzata come misura delle prestazioni del sistema di identificazione [2]. Giudica le capacità di classificazione di un sistema di identificazione [12]. Gli algoritmi di attacco utilizzati da sinistra a destra: EAD-EN, FGSM.
Gli esempi avversari hanno dimostrato quindi di avere successo "nell'imbrogliare" l'apprendimento profondo e gli algoritmi di riconoscimento facciale.

5 Conclusioni

L'obiettivo di questa tesi e' esporre le fragilita' dell'apprendimento automatico che si basa sul riconoscimento facciale.
Per parlare delle vulnerabilita' che interessano il riconoscimento facciale, prima vengono presentate tutte le principali tecniche per il "face recognition", con i vari modelli ed i dataset piu' utilizzati e poi i vari metodi di attacco ai sistemi di apprendimento automatico.
L'efficacia delle varie tecniche inizialmente analizzate solo teoricamente, e' verificata all'interno del progetto SmartBox in cui vengono riportati i dati sperimentali che mostrano gli effetti delle varie tecniche.
Lo SmartBox può essere usato come punto di riferimento per analizzare l'effetto di un avversario sui sistemi di riconoscimento facciale. In piu' puo' anche essere utilizzato come "base" per creare nuovi algoritmi di attacco avversari o tecniche di difesa. I risultati dei nuovi sistemi di attacco o difesa creati possono essere poi confrontati con quelli gia' esistenti in modo da valutarne l'efficacia.
Inoltre e' stata vista la possibilita' di utilizzare tecniche gia' esistenti per e valutarne l'efficacia su altri set di dati.
In futuro, il programma puo' rappresentare quindi il punto di partenza per aggiungere algoritmi che possono ingannare altre modalità biometriche, come l'iride degli occhi.

Riferimenti bibliografici

[1] Di Luca Sambucci - et al. *La Difesa della Privacy ai Tempi dell'intelligenza artificiale*. Mar. 2021. URL: https://www.notizie.ai/la-difesa-della-privacy-ai-tempi-dellintelligenza-artificiale.

[2] R.M. Bolle et al. "The relation between the ROC curve and the CMC". In: *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05)*. 2005, pp. 15–20. DOI: 10.1109/AUTOID.2005.48.

[3] Nicholas Carlini e David A. Wagner. "Towards Evaluating the Robustness of Neural Networks". In: *CoRR* abs/1608.04644 (2016). arXiv: 1608.04644. URL: http://arxiv.org/abs/1608.04644.

[4] Pin-Yu Chen et al. *EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples*. 2018. arXiv: 1709.04114 [stat.ML].

[5] Jiankang Deng et al. "ArcFace: Additive Angular Margin Loss for Deep Face Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Giu. 2019.

[6] Akhil Goel et al. "SmartBox: Benchmarking Adversarial Detection and Mitigation Algorithms for Face Recognition". In: *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. 2018, pp. 1–7. DOI: 10.1109/BTAS.2018.8698567.

[7] Ian J. Goodfellow, Jonathon Shlens e Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. 2015. arXiv: 1412.6572 [stat.ML].

[8] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[9] Weiyang Liu et al. "SphereFace: Deep Hypersphere Embedding for Face Recognition". In: *CoRR* abs/1704.08063 (2017). arXiv: 1704.08063. URL: http://arxiv.org/abs/1704.08063.

[10] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi e Pascal Frossard. "DeepFool: a simple and accurate method to fool deep neural networks". In: *CoRR* abs/1511.04599 (2015). arXiv: 1511.04599. URL: http://arxiv.org/abs/1511.04599.

[11] Omkar M. Parkhi, Andrea Vedaldi e Andrew Zisserman. “Deep Face Recognition”. In: *Proceedings of the British Machine Vision Conference (BMVC)*. A cura di Mark W. Jones Xianghua Xie e Gary K. L. Tam. BMVA Press, set. 2015, pp. 41.1–41.12. ISBN: 1-901725-53-7. DOI: 10.5244/C.29.41. URL: https://dx.doi.org/10.5244/C.29.4.

[12] N. K. Ratha et al. “The Relation between the ROC Curve and the CMC”. In: *Proceedings. Fourth IEEE Workshop on Automatic Identification Advanced Technologies*. Los Alamitos, CA, USA: IEEE Computer Society, ott. 2005, pp. 15–20. DOI: 10.1109/AUTOID.2005.48. URL: https://doi.ieeecomputersociety.org/10.1109/AUTOID.2005.48.

[13] Florian Schroff, Dmitry Kalenichenko e James Philbin. “FaceNet: A Unified Embedding for Face Recognition and Clustering”. In: *CoRR* abs/1503.03832 (2015). arXiv: 1503.03832. URL: http://arxiv.org/abs/1503.03832.

[14] K. Simonyan e A. Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: (2014). arXiv: 1409.1556.. URL: Available:%20http://arxiv.org/abs/1409.1556.

[15] Christian Szegedy et al. *Going Deeper with Convolutions*. 2014. arXiv: 1409.4842 [cs.CV].

[16] Yaniv Taigman et al. “DeepFace: Closing the Gap to Human-Level Performance in Face Verification”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1701–1708. DOI: 10.1109/CVPR.2014.220.

[17] Fatemeh Vakhshiteh, Ahmad Nickabadi e Raghavendra Ramachandra. “Adversarial Attacks Against Face Recognition: A Comprehensive Study”. In: *IEEE Access* 9 (2021), pp. 92735–92756. DOI: 10.1109/ACCESS.2021.3092646.

[18] Hao Wang et al. “CosFace: Large Margin Cosine Loss for Deep Face Recognition”. In: *CoRR* abs/1801.09414 (2018). arXiv: 1801.09414. URL: http://arxiv.org/abs/1801.09414.

[19] Xiang Wu, Ran He e Zhenan Sun. “A Lightened CNN for Deep Face Representation”. In: *CoRR* abs/1511.02683 (2015). arXiv: 1511.02683. URL: http://arxiv.org/abs/1511.02683.