

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FÁBIA ISABELLA PIRES ENEMBRECK

**RE-IDENTIFICAÇÃO DE PESSOAS EM IMAGENS DIGITAIS
UTILIZANDO REDES NEURAS SIAMESAS E *TRIPLET* BASEADAS
EM UMA REDE NEURAL CONVOLUCIONAL E UM *AUTOENCODER***

DISSERTAÇÃO

PONTA GROSSA
2020

FÁBIA ISABELLA PIRES ENEMBRECK

**RE-IDENTIFICAÇÃO DE PESSOAS EM IMAGENS DIGITAIS
UTILIZANDO REDES NEURAS SIAMESAS E *TRIPLET* BASEADAS
EM UMA REDE NEURAL CONVOLUCIONAL E UM *AUTOENCODER***

Dissertação apresentada como requisito parcial
à obtenção do grau de Mestre em Ciência da
Computação, do Departamento Acadêmico
de Informática, da Universidade Tecnológica
Federal do Paraná.

Orientador: Dr. Erikson Freitas de Moraes

PONTA GROSSA

2020

Ficha catalográfica elaborada pelo Departamento de Biblioteca
da Universidade Tecnológica Federal do Paraná, Campus Ponta Grossa
n.61/20

E56 Enembreck, Fábila Isabella Pires

Re-identificação de pessoas em imagens digitais utilizando redes neurais
siamesas e triplet baseadas em uma rede neural convolucional e um autoencoder.
/ Fábila Isabella Pires Enembreck, 2020.

78 f. : il. ; 30 cm.

Orientador: Prof. Dr. Erikson Freitas de Moraes

Dissertação (Mestrado em Ciência da Computação) - Programa de Pós-
Graduação em Ciência da Computação. Universidade Tecnológica Federal do
Paraná, Ponta Grossa, 2020.

1. Pessoas - Identificação. 2. Vigilância eletrônica. 3. Imagens digitais. 4. Redes
neurais (Computação). I. Moraes, Erikson Freitas de. II. Universidade Tecnológica
Federal do Paraná. III. Título.

CDD 004



Ministério da Educação
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ CÂMPUS PONTA GROSSA
Diretoria de Pesquisa e Pós-Graduação
Programa de Pós-Graduação em Ciência da Computação



FOLHA DE APROVAÇÃO

Título de Dissertação 23/2020

RE-IDENTIFICAÇÃO DE PESSOAS EM IMAGENS DIGITAIS UTILIZANDO REDES NEURAI SIAMESAS E TRIPLET BASEADAS EM UMA REDE NEURAL CONVOLUCIONAL E UM AUTOENCODER

Por

Fábia Isabella Pires Enembreck

Esta dissertação foi apresentada às **14:30 horas de 18 de agosto de 2020**, na sala de **videoconferência online** como requisito parcial para a obtenção do título de MESTRE EM CIÊNCIA DA COMPUTAÇÃO, Programa de Pós-Graduação em Ciência da Computação. O candidato foi arguido pela Banca Examinadora, composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho APROVADO.

Prof. Dr. William Robson Schwartz
(UFMG)

Prof. Dr. Luciano José Senger (UEPG)

Prof^a. Dr^a. Marcella Scoczynski Ribeiro
Martins (UTFPR)

Prof. Dr. Erikson Freitas de Moraes
(UTFPR)
Orientador e presidente da banca



Visto do Coordenador:

Prof. Dr. André Koscianski
Coordenador do PPGCC
UTFPR – Câmpus Ponta Grossa

Dedico esse trabalho aos meus pais, Fábio e Maria Eliana.

AGRADECIMENTOS

Agradeço aos meus pais, Fábio e Maria Eliana e à minha irmã Flávia por serem meus maiores incentivadores. Por estarem sempre ao meu lado, acreditando que sou capaz de alcançar meus objetivos.

À Deus, por me dar força e saúde para ir atrás dos meus sonhos e conseguir concluir esse desafio.

Ao Prof. Dr. Erikson pela dedicação com me orientou neste trabalho e pelo aprendizado que me proporcionou. Obrigada também por suas valiosas considerações e sugestões.

Ao meu namorado, Gustavo, por me dar apoio e carinho nos momentos que mais precisei e pela compreensão nos dias que te deixei de lado para focar na dissertação. Obrigada por sempre acreditar em mim.

Aos professores Dr. Luciano, Dra. Marcela e Dr. William, por aceitaram participar da banca de defesa e pelas sugestões dadas na banca de qualificação.

Aos amigos que fiz durante o mestrado. Aleffer, Bauke, Eduardo, Everton, Lin e Luís obrigada pela amizade, diversão e ajuda sempre que precisei durante esses anos. Nunca vou esquecer de vocês.

Para os membros do COVAP, pela troca de conhecimentos e pela oportunidade de conhecer outros trabalhos. Um agradecimento especial ao Felipe, Leonardo e Sérgio, pela ajuda com os experimentos.

Aos professores do PPGCC, pelos ensinamentos oferecidos durante o curso, que com certeza levarei para a minha vida toda.

E também gostaria de agradecer à UTFPR pelo apoio financeiro, que tornou possível o desenvolvimento desta pesquisa.

RESUMO

ENEMBRECK, Fábila Isabella Pires. Re-identificação de pessoas em imagens digitais utilizando Redes Neurais Siamesas e *Triplet* baseadas em uma Rede Neural Convolutiva e um *Autoencoder*. 2020. 78 p. Dissertação (Mestrado em Ciência da Computação), Universidade Tecnológica Federal Do Paraná. Ponta Grossa, 2020.

Em ambientes monitorados por câmeras de segurança, o problema de determinar se uma pessoa que está sendo observada já esteve presente na cena ou não, independente se o sistema utiliza uma ou mais câmeras, é chamado de re-identificação de pessoas. Este problema é considerado desafiador, uma vez que as imagens obtidas por câmeras estão sujeitas a sofrer grandes variações, como iluminação e perspectiva. Além disso, pessoas em imagens podem passar por transformações e oclusões parciais. Com isso, este trabalho tem como objetivo o desenvolvimento de duas abordagens para re-identificação de pessoas que sejam robustas a essas variações, por meio de técnicas de aprendizagem profunda. A primeira abordagem proposta utiliza uma arquitetura de rede neural siamesa, composta por duas sub-redes idênticas, esse modelo recebe duas imagens de entrada que podem ser ou não de uma mesma pessoa. A segunda abordagem consiste em uma rede neural *triplet*, com três sub-redes idênticas e que recebe de entrada uma imagem de referência de uma determinada pessoa, uma segunda imagem da mesma pessoa e outra imagem de uma pessoa diferente. Ambas as redes possuem sub-redes idênticas, formadas por uma rede neural convolutiva que irá extrair características gerais de cada imagem e uma rede *autoencoder*, responsável por tratar as grandes variações que as imagens da entrada podem sofrer. Para analisar e comparar as redes desenvolvidas foram utilizados três *datasets*, sendo que as medidas de avaliação escolhidas para análise foram a acurácia e a curva CMC. Experimentos realizados comprovaram uma melhora de até 71,05% nos resultados com a utilização do *autoencoder* nas sub-redes. Além disso, os experimentos também mostraram uma superioridade da Rede Neural *Triplet* desenvolvida neste trabalho em relação a Rede Neural Siamesa e a outros métodos do estado da arte.

Palavras-chaves: Re-identificação de Pessoas. *Deep learning*. Vigilância. Rede Neural Siamesa. Rede Neural *Triplet*.

ABSTRACT

ENEMBRECK, Fábila Isabella Pires. Person re-identification in digital images using Siamese and Triplet Neural Networks based on a Convolutional Neural Network and an Autoencoder. 2020. 78. Partial Exam (Master in Computer Science) – Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2020.

In environments monitored by security cameras, the problem of identifying if a person being watched has ever been in the scene or not, independent of the system uses one or more cameras, is called person re-identification. This problem is considered challenging, since the images obtained by cameras are subject to many variations, such as lighting and perspective. In addition, people in pictures may undergo transformations and partial occlusions. This work aims to develop two approaches for person re-identification robust to these variations, through deep learning techniques. The first approach proposed uses a Siamese neural network architecture, composed of two identical subnets, this model receives two input images that may or may not be from the same person. The second approach consists of a triplet neural network, with three identical subnets, which receives a reference image from a certain person, a second image from the same person and another image from a different person. Both networks have identical subnets, formed by a convolutional neural network that will extract general characteristics from each image and an autoencoder network, responsible for dealing with the great variations that the input images may undergo. To analyze and compare the developed networks, three datasets were used, and the metrics chosen for analysis were accuracy and the CMC curve. Experiments carried out proved an improvement up to 71.05% in the results with the use of the autoencoder in the subnets. Also, the experiments showed a superiority of the Triplet Neural Network developed in this work to the Siamese Neural Network and other state-of-the-art methods.

Key-words: Person re-identification. Deep learning. Surveillance. Siamese Neural Network, Triplet Neural Network.

LISTA DE FIGURAS

Figura 1	– Exemplo de um sistema de vigilância multi-câmera para re-identificação ...	15
Figura 2	– Neurônio biológico	20
Figura 3	– Neurônio artificial	21
Figura 4	– Rede Neural com duas camadas ocultas	22
Figura 5	– Funções de Ativação	23
Figura 6	– Camadas de uma CNN	24
Figura 7	– Exemplo de convolução.	25
Figura 8	– <i>Max Pooling</i>	26
Figura 9	– Exemplo de <i>Autoencoder</i>	28
Figura 10	– <i>Denosing Autoencoder</i>	29
Figura 11	– Rede Neural Siamesa	30
Figura 12	– <i>Triplet Loss</i>	31
Figura 13	– Principais etapas para a re-identificação de pessoas em imagens	32
Figura 14	– Arquitetura proposta por Li <i>et al.</i> (2014)	34
Figura 15	– Arquitetura proposta por Ahmed, Jones e Marks (2015).	35
Figura 16	– Arquitetura proposta por McLaughlin, Rincon e Miller (2016)	37
Figura 17	– Arquitetura proposta por Cheng <i>et al.</i> (2016)	38
Figura 18	– Sub-rede proposta por Cheng <i>et al.</i> (2016)	39
Figura 19	– Etapas adotadas para o desenvolvimento deste trabalho	42
Figura 20	– Modelo de sub-rede proposto	44
Figura 21	– Modelo de Rede Neural Siamesa proposto	45
Figura 22	– Associação entre pares positivos e pares negativos	46
Figura 23	– Modelo de Rede Neural <i>Triplet</i> proposto	47
Figura 24	– Modelos de Rede Neurais propostas	48
Figura 25	– <i>Dataset VIPeR</i>	51
Figura 26	– <i>Dataset i-LIDSVID</i>	52
Figura 27	– <i>Dataset CUHK03</i>	52
Figura 28	– Curva CMC	55
Figura 29	– Resultado da aplicação da técnica de <i>Data Augmentation</i> no <i>dataset VIPeR</i>	58
Figura 30	– Acurácia da Rede Neural Siamesa, em relação ao N° de Épocas de treinamento, para sub-redes com 4 camadas, utilizando o <i>dataset VIPeR</i>	59
Figura 31	– Acurácia, em relação ao N° de Épocas de treinamento, utilizando o <i>dataset VIPeR</i>	60
Figura 32	– Curva CMC, utilizando o <i>dataset VIPeR</i>	61
Figura 33	– Acurácia, em relação ao N° de Épocas de treinamento, utilizando o <i>dataset i-LIDSVID</i>	62
Figura 34	– Curva CMC, utilizando o <i>dataset i-LIDSVID</i>	63
Figura 35	– Acurácia, em relação ao N° de Épocas de treinamento, utilizando o <i>dataset CUHK03</i>	64
Figura 36	– Curva CMC, utilizando o <i>dataset CUHK03</i>	65
Figura 37	– Acurácia em relação ao N° de Épocas de treinamento	66
Figura 38	– Comparação entre as Curvas CMC	67
Figura 39	– Comparação das Curva CMC com métodos do estado da arte, utilizando o <i>dataset VIPeR</i>	68

Figura 40	–	Comparação das Curva CMC com métodos do estado da arte, utilizando o <i>dataset</i> i-LIDSVID	69
Figura 41	–	Comparação das Curva CMC com métodos do estado da arte, utilizando o <i>dataset</i> CUHK03	71

LISTA DE TABELAS

Tabela 1	– <i>Datasets</i> públicos para re-identificação de pessoas em imagens digitais	41
Tabela 2	– Distribuição dos ângulos do ponto de vista no <i>dataset</i> VIPeR.....	51
Tabela 3	– Ferramentas utilizadas nos experimentos	56
Tabela 4	– Máquina utilizada nos experimentos	56
Tabela 5	– Acurácia da Rede Neural Siamesa, em relação ao N° de Épocas de treinamento, para sub-redes com 4 camadas, utilizando o <i>dataset</i> VIPeR.....	58
Tabela 6	– Acurácia em relação ao N° de Épocas de treinamento, utilizando o <i>dataset</i> VIPeR.....	59
Tabela 7	– Curva CMC, utilizando o <i>dataset</i> VIPeR	60
Tabela 8	– Acurácia em relação ao N° de Épocas de treinamento, utilizando o <i>dataset</i> i-LIDSVID.....	61
Tabela 9	– Curva CMC, utilizando o <i>dataset</i> i-LIDSVID	63
Tabela 10	– Acurácia em relação ao N° de Épocas de treinamento, utilizando o <i>dataset</i> CUHK03.....	64
Tabela 11	– Curva CMC, utilizando o <i>dataset</i> CUHK03.....	65
Tabela 12	– Comparação das Curva CMC com métodos do estado da arte, utilizando o <i>dataset</i> VIPeR	68
Tabela 13	– Comparação das Curva CMC com métodos do estado da arte, utilizando o <i>dataset</i> i-LIDSVID.....	69
Tabela 14	– Comparação das Curva CMC, com métodos do estado da arte,, utilizando o <i>dataset</i> CUHK03	70

LISTA DE QUADROS

Quadro 1	–	Trabalhos Relacionados.....	40
Quadro 2	–	Matriz de Confusão	53

LISTA DE ABREVIATURAS E SIGLAS

AE	Autoencoder
CMC	Cumulative Matching Characteristic
CNN	Convolutional Neural Network
DAE	Denoising Autoencoder
DGD	Domain Guided Dropout
DML	Deep Metric Learning
ELF	Ensemble of Localized Features
HSV	Hue, Saturation and Value
KISSME	Keep It Simple and Straightforward Metric
KNN	K-Nearest Neighbors
ReID	Re-identification
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue
SDALF	Symmetry-Driven Accumulation of Local Features
SIFT	Scale-Invariant Feature Transform

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	16
1.1.1	Objetivos Específicos	16
1.2	JUSTIFICATIVA	16
1.3	ORGANIZAÇÃO DO TRABALHO	17
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	MACHINE LEARNING	19
2.1.1	Redes Neurais Artificiais	20
2.1.2	Deep Learning	22
2.2	REDES NEURAIS CONVOLUCIONAIS	23
2.2.1	Camada Convolutacional	24
2.2.2	Camada de Pooling	25
2.2.3	<i>Batch Normalization</i>	26
2.2.4	<i>Dropout</i>	27
2.3	REDE AUTOENCODER	27
2.4	REDES NEURAIS SIAMESAS	29
2.4.1	Triplets	31
3	TRABALHOS RELACIONADOS	32
3.1	<i>DATASETS</i>	39
4	MATERIAL E MÉTODOS	42
4.1	REDE NEURAIS PROPOSTAS	43
4.1.1	Rede Neural Siamesa	44
4.1.2	<i>Triplet Loss</i>	46
4.1.3	Compilação dos modelos propostos	48
5	EXPERIMENTOS E RESULTADOS	50
5.1	SETUP EXPERIMENTAL	50
5.1.1	Datasets	50
5.1.2	Medidas de Avaliação	53
5.1.2.1	Acurácia	53
5.1.2.2	Curva CMC	54
5.1.3	Ambiente de Trabalho e Implementação	56
5.2	RESULTADOS	57
5.2.1	Experimentos com o <i>Dataset</i> VIPeR	57
5.2.2	Experimentos com o <i>Dataset</i> i-LIDSVID	61
5.2.3	Experimentos com o <i>Dataset</i> CUHK03	62
5.3	DISCUSSÃO DOS RESULTADOS ENCONTRADOS	64
5.4	COMPARAÇÃO COM O ESTADO DA ARTE	67
5.4.1	<i>Dataset</i> VIPeR	67
5.4.2	<i>Dataset</i> i-LIDSVID	68
5.4.3	<i>Dataset</i> CUHK03	70
6	CONCLUSÃO	72
	REFERÊNCIAS	74

1 INTRODUÇÃO

A visão computacional é um ramo da ciência que extrai informações de imagens digitais por meio de algoritmos que buscam identificar o conteúdo de uma ou mais imagens, assim como reconstruir suas propriedades, tais como iluminação, distribuição de cores e formato. Essa área tem como objetivo fazer com que o computador reproduza o funcionamento da visão humana. Krishna (2017) define a visão como um mecanismo que pode ser utilizado na captura e detecção de detalhes em uma cena e também para interpretação das informações disponíveis na cena. De forma análoga, pode-se dizer que as câmeras substituem os olhos na captura de uma imagem de uma determinada cena e o computador representa o cérebro no processo de interpretação e extração de informações em imagens digitais (SZELISKI, 2010; KRISHNA, 2017).

Segundo Szeliski (2010), a visão computacional começou a ser estudada em 1966 por um estudante de graduação do *MIT*, para o desenvolvimento de um projeto de verão que tinha como objetivo fazer com que o computador descrevesse uma cena capturada a partir de uma câmera ligada a ele. No entanto, o problema se mostrou muito mais complexo para se resolver em apenas um verão e desde então passou a ser dividido em diversos problemas, assim como a agregar outras disciplinas para a solução desses problemas, como a neurociência, inteligência artificial e processamento de imagens, tornando a visão computacional um campo interdisciplinar. Ao longo dos anos, o campo de estudo na área de visão computacional cresceu e foi utilizado em diversas aplicações, como por exemplo biometria, reconhecimento de caracteres, detecção de face, diagnóstico médico, *surveillance*, entre outras. Neste trabalho um problema na área de *surveillance* ou vigilância é tratado utilizando técnicas de *deep learning* aplicadas a sistemas de segurança inteligentes monitorados por câmeras para re-identificar pessoas.

Sistemas de segurança monitorados por câmeras têm sido cada vez mais utilizados. Eles podem ser manuais, de forma que seu controle e análise sejam feitos por um operador humano. No entanto, um grande número de dados visuais é gerado, necessitando atenção extrema do operador. Assim, este tipo de monitoramento, muitas vezes, torna-se propenso a erros, uma vez que deve-se ter o controle de uma quantidade muito grande de dados em tempo real. Neste caso, a utilização de um sistema inteligente de vigilância é mais indicado para processar esses dados, já que podem trabalhar com o monitoramento em tempo real de um ambiente, fornecendo uma interpretação automática de cenas para compreensão de atividades e interações entre objetos com base em informações visuais adquiridas. O principal desafio desse tipo de sistema é a resolução de uma série de problemas para tornar possível a análise do que acontece em uma cena (JR; SCHWARTZ, 2016). A utilização de sistemas inteligentes de segurança em ambientes de monitoramento se torna muito mais eficaz e é capaz de detectar movimento, rastrear pessoas e objetos, re-identificar pessoas, identificar atividades suspeitas, entre outras aplicações (TU *et al.*, 2007).

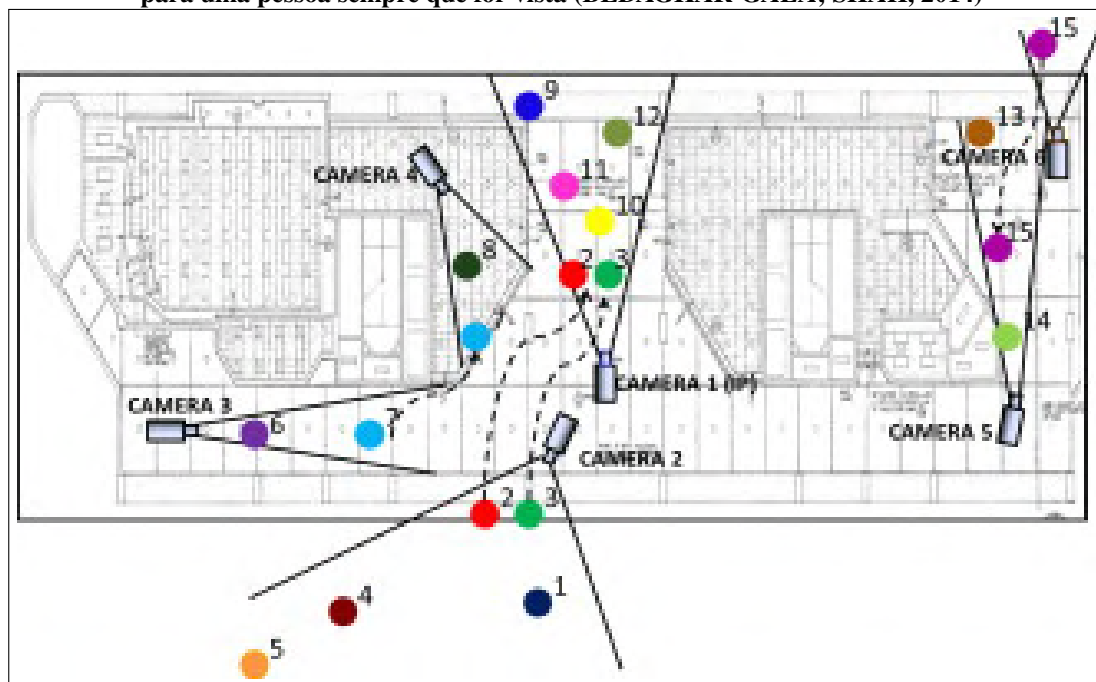
Acompanhar pessoas em um ambiente monitorado por uma ou várias câmeras é essencial para a análise de cenas em áreas amplas e a sua re-identificação é um aspecto fundamental do rastreamento (BEDAGKAR-GALA; SHAH, 2014). De acordo com Dictionary (2014), a palavra "re-identificar" está associada a ação de identificar algo novamente ou por um tempo, como por exemplo, a cada *frame* de um vídeo. A re-identificação de pessoas é um dos problemas relacionados à sistemas inteligentes de vigilância e consiste em identificar pessoas, que já foram identificadas em um outro momento, na mesma câmera ou em imagens obtidas por câmeras diferentes (AHMED; JONES; MARKS, 2015). Com isso, é muito importante que sistemas de segurança possam rastrear pessoas depois delas deixarem o campo de visão de uma câmera e aparecerem no campo de visão de outra, mesmo se esses campos de visão não se sobrepõem. Ou seja, esses sistemas devem ser capazes de re-identificar uma pessoa de qualquer lugar que possa ser observada, obtendo a correspondência das detecções e rastreando apenas uma trajetória (AVRAHAM *et al.*, 2012).

Bedagkar-Gala e Shah (2014) definem o problema da re-identificação de pessoas como um processo em que se pretende estabelecer correspondência entre imagens diferentes de uma mesma pessoa. Em um sistema que utiliza apenas uma câmera a re-identificação de pessoas pode ser válida para detectar se uma pessoa esteve presente em uma mesma localização várias vezes, por exemplo, além disso, também há sistemas de segurança monitorados por mais de uma câmera, nesse caso a re-identificação pode servir para se obter a trajetória de uma pessoa no local monitorado. Esse problema é considerado difícil, uma vez que diversas mudanças na aparência dos indivíduos analisados, causadas por diferentes condições de aquisição de imagens pelas câmeras, tais como: oclusões, variação de iluminações, mudanças de perspectivas, interferência de fundo, entre outros (MCLAUGHLIN; RINCON; MILLER, 2016).

Na Figura 1, retirada de (BEDAGKAR-GALA; SHAH, 2014), temos um exemplo de sistema multi-câmera com imagens não sobrepostas, formado por seis câmeras. Essa imagem representa a vista de cima de uma fábrica, sendo os pontos numerados identificadores das pessoas que estão no local. O movimento de locomoção das pessoas é representado pelas linhas pontilhadas. O re-identificador, neste caso, é utilizado para reconstrução da trajetória de uma pessoa na cena. Quando uma pessoa que está na vista de uma determinada câmera se desloca para a vista de outra câmera, o re-identificador estabelece uma correspondência entre essas imagens e as pessoas recebem sempre o mesmo identificador. Como pode ser observado, as pessoas representadas pelos pontos 2 e 3 estão inicialmente sob o campo de visão da Câmera 2. Ao se deslocar, as pessoas não serão mais monitoradas pela câmera, alcançado a visão da Câmera 1, o sistema reconhece que as pessoas já estiveram presente no ambiente, atribuindo a elas o mesmo rótulo. (BEDAGKAR-GALA; SHAH, 2014).

Considerando o problema de re-identificação de pessoas em imagens digitais, neste trabalho foram desenvolvidas duas abordagens para tratá-lo. A primeira é uma rede neural siamesa, constituída por duas sub-redes idênticas compostas por uma rede neural convolucional e uma

Figura 1 – Exemplo de um sistema de vigilância multi-câmera para re-identificação. Nesse ambiente, monitorado por 6 câmeras, há 15 pessoas andando. Cada pessoa é identificada por um número e à medida que estão se movendo, podem ser observada por uma câmera diferente. Com isso, o sistema deve atribuir a o mesmo número de identificação para uma pessoa sempre que for vista (BEDAGKAR-GALA; SHAH, 2014)



Fonte: (BEDAGKAR-GALA; SHAH, 2014)

rede *autoencoder*. A rede neural siamesa tem como função decidir se duas imagens diferentes referem-se a mesma pessoa, comparando duas imagens de entrada, através de características extraídas de cada uma pelas sub-redes. A outra técnica desenvolvida é um Rede Neural Triplet, que é composta por três sub-redes idênticas, neste método a rede recebe como entrada três imagens, sendo um par positivo e um negativo, que compartilham uma imagem de referência. A rede tem como objetivo aproximar as imagens positivas e aumentar a distância do par negativo.

Vale ressaltar que ambas as redes propostas possuem o mesmo modelo de sub-rede. Nesse caso, cada sub-rede é formada por uma rede neural convolucional para extração de características que produz um vetor de características da imagem de entrada e na sequência uma rede neural *autoencoder* irá reconstruir esse vetor, com objetivo de amenizar os ruídos presentes na imagem que possam comprometer a comparação com a outra imagem, de forma que sejam mantidas características mais relevantes para a re-identificação.

As duas redes foram treinadas e testadas utilizando os *datasets* VIPeR (GRAY; BRENNAN; TAO, 2007), i-LIDSVID (WANG *et al.*, 2014) e CUHK03 (LI *et al.*, 2014). Além disso, para validação e comparação das redes as medidas de avaliação utilizadas foram a acurácia e a curva CMC.

Um diferencial importante do trabalho é o uso do *autoencoder* na saída da subrede proposta eliminando ruídos indesejáveis. O uso do *autoencoder* melhorou os resultados em ambas as redes implementadas em até 71,05% de acurácia.

1.1 OBJETIVOS

O objetivo principal deste trabalho é avaliar duas abordagens para re-identificação de pessoas em imagens digitais utilizando técnicas de aprendizagem profunda. A primeira abordagem será uma rede neural siamesa e a segunda uma rede neural triplet, construídas utilizando uma sub-rede comum baseada em uma rede neural convolucional e uma rede *autoencoder*, que devem realizar a re-identificação de pessoas tendo como entrada recortes de imagens de pessoas já detectadas previamente.

1.1.1 Objetivos Específicos

Para cumprimento do objetivo principal deste trabalho foram estabelecidos os seguintes objetivos específicos:

1. Identificação de conjuntos de dados (*datasets*) que possuem vários recortes de imagens de uma mesma pessoa, obtidas de pontos de vista diferentes;
2. Desenvolvimento de uma sub-rede composta por uma rede neural convolucional e uma rede *autoencoder* capaz de extrair e seletar as características mais relevantes da imagem;
3. Construir uma rede siamesa com base na sub-rede desenvolvida;
4. Construir uma rede triplet com base na sub-rede desenvolvida;
5. Validação dos modelos usando as medidas de avaliação escolhidas para verificação da viabilidade e imagens dos *datasets* selecionados;
6. Comparação dos modelos de rede neural desenvolvidos;
7. Comparação das redes neurais propostas com o mesmo modelo sem o *autoencoder*;
8. Comparação dos modelos com outros disponíveis na literatura.

1.2 JUSTIFICATIVA

De acordo com Bedagkar-Gala e Shah (2014), o problema de re-identificar pessoas em imagens digitais é considerado desafiador. Esse problema pode ser aplicado em diversas áreas como segurança, rastreamento, recuperação de trajetória de uma pessoa em um determinado ambiente monitorado por várias câmeras, entre outras. O problema é considerado ainda sem

solução, apesar de existirem diversas técnicas para a re-identificação de pessoas.(BEDAGKAR-GALA; SHAH, 2014).

Um dos métodos comumente utilizado para re-identificar pessoas é o manual. Neste caso, depende de um operador humano para examinar imagens e atestar se pertencem a uma mesma pessoa. Considerando um sistema de vigilância, esse operador deve identificar cada pessoa em um vídeo e associar a uma pessoa que já foi vista anteriormente ou não. Se esse ambiente monitorado possui um número limitado de pessoas que se movimentam com pouca frequência, essa não é uma tarefa tão complexa. No entanto, em um ambiente real podem transitar diversas pessoas diferentes ao mesmo tempo, tornando essa tarefa se torna mais difícil e, conseqüentemente, forçando o operador a dedicar extrema atenção as imagens, de forma que nenhuma pessoa seja perdida de vista. Nesse contexto, um pequeno momento de cansaço ou distração pode comprometer uma re-identificação. (JR; SCHWARTZ, 2016).

Além disso, a re-identificação também pode ser automática através de um sistema computacional. Esse tipo de re-identificação não exige um operador humano para analisar imagens, evitando erros por cansaço ou falta de atenção, por exemplo. No entanto, nesses sistemas a re-identificação também é muito complexa, uma vez que as imagens podem apresentar inúmeras variações de iluminação, pose do pedestre, ponto de vista, oclusões parciais, variações de aparência e baixas resoluções. Os métodos já existentes no estado-da-arte (Capítulo 4) têm dificuldade para a combinação correta de imagens da mesma pessoa sob variações bruscas. Visto que essas variações nas imagens podem fazer com que uma mesma pessoa pareça diferente de uma imagem à outra para o re-identificador, assim como pessoas diferentes podem ficar muito parecidas, fazendo com que o sistema atribua a elas o mesmo rótulo (CHENG *et al.*, 2016).

Nesse contexto, o presente trabalho apresenta duas abordagens para a re-identificação de pessoas em imagens. As abordagens utilizam técnicas de aprendizagem profunda para extrair características de imagens de pessoas já detectadas e manter as características mais relevantes para o processo. Com isso, cada imagem de entrada possui um vetor próprio de características para ser comparado com o vetor de outra imagem de entrada e o re-identificador deve analisar se pertencem ou não a mesma pessoa.

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho é composto por seis capítulos, entre os quais o primeiro apresenta uma introdução a respeito do estudo proposto, bem como os objetivos deste trabalho. O segundo capítulo apresenta os temas e conceitos abordados na pesquisa, constituindo uma fundamentação teórica importante. No quarto capítulo é retratada a metodologia empregada neste trabalho, as técnicas proposta e os *datasets* utilizados. No terceiro capítulo são apresentados outros trabalhos que trataram sobre o problema de re-identificação de pessoas em imagens digitais, bem como o

método proposto em cada um. No penúltimo capítulo são descritos os experimentos realizados, os resultados obtidos e uma comparação entre os métodos desenvolvidos. Por fim, são descritas as conclusões e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apontar os principais conceitos para o desenvolvimento da proposta deste trabalho, apresentando uma introdução a respeito de aprendizagem de máquina e redes neurais, com destaque para as redes neurais convolucionais, redes neurais *autoencoders* e redes neurais siamesas.

2.1 MACHINE LEARNING

Machine Learning ou aprendizagem de máquina é uma área da inteligência artificial que tem crescido muito nos últimos anos para a resolução de problemas em diversos campos. Segundo Shalev-Shwartz e Ben-David (2014), *machine learning* é o processo no qual permite à máquina identificar padrões em dados. Em outras palavras, concede ao computador a capacidade de aprender a partir de uma entrada, criando uma certa experiência que se transformará em conhecimento posteriormente, por meio do treinamento de dados.

De acordo com Mitchell (1997), o aprendizado é adquirido pelo computador através da experiência, em relação a alguma classe de tarefas e medida de desempenho. Assim, quando se objetiva solucionar um problema de aprendizado, deve-se definir qual é a classe de tarefas, a medida de desempenho e como será adquirida a experiência. Como exemplo, Mitchell (1997) citou um programa de computador que aprender a jogar damas, neste caso, a classe de tarefas é jogar damas, a medida de desempenho é a porcentagem de jogos ganhos contra oponentes e a experiência se dá jogando contra si mesmo.

Goodfellow, Bengio e Courville (2016) definiram tarefas como sendo o método utilizado pelo computador para o aprendizado, assim as tarefas são modos de como o computador deve processar uma determinada amostra de dados em *machine learning*. Um exemplo é a classificação, em que o computador deve especificar uma categoria ou classe a que uma entrada pertence. Com isso, uma medida de desempenho deve ser avaliada para a tarefa executada pela máquina. Uma medida de desempenho utilizada para a classificação é a acurácia, que é a proporção entre as amostras de dados em que o computador produz a saída correta. Outro conceito importante é a experiência que é adquirida pelo computador pode ser obtida por meio de algoritmos de aprendizado supervisionado ou algoritmos de aprendizado não supervisionado e deve ser determinado antes do processo de aprendizagem. Com isso define-se aprendizado supervisionado e aprendizado não supervisionado como:

- **Aprendizado Supervisionado:** os experimentos são realizados com amostras de dados recebidas na entrada associadas a uma classe ou rótulo, com isso o computador deve aprender uma forma de se chegar a classe correta por meio desses dados.

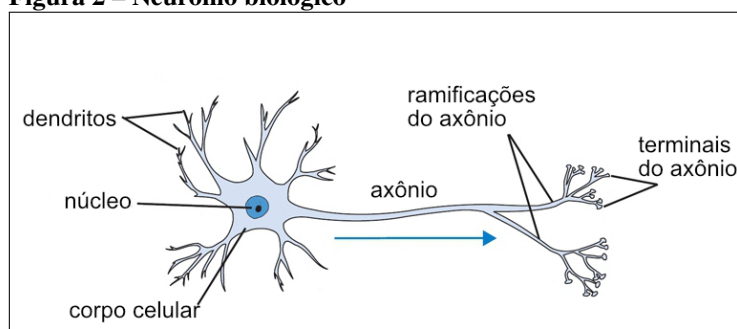
- **Aprendizado Não Supervisionado:** os experimentos são feitos com base em um conjunto de dados sem classes, nesse caso os algoritmos devem aprender características úteis que identificam cada conjunto de dados. Um exemplo é a clusterização que deve identificar as amostras mais semelhantes entre si por meio de suas características.

2.1.1 Redes Neurais Artificiais

Uma área de aprendizagem de máquina é aquela definida pelas redes neurais artificiais. Essas redes são inspiradas no comportamento do cérebro biológico, mais especificamente nos neurônios do sistema nervoso central, e tentam reproduzir o seu funcionamento. A motivação para o desenvolvimento das redes neurais artificiais veio do fato de que o cérebro tem a capacidade de organizar seus neurônios para que cálculos complexos seja processados de maneira muito mais rápida do que um computador para atividades como reconhecimento de padrões, percepção e controle motor, por exemplo (GRAUPE, 2007; HAYKIN S., 2009).

Uma rede neural biológica é formada por células nervosas, também conhecidas como neurônios, ilustrado na Figura 2. A passagem de informação pelos neurônios se dá por meio de sinais elétricos que passam de uma célula a outra pelo axônio, que são ligadas aos outros neurônios por meio de terminais. Após isso, a informação é recebida pelos dendritos e processadas no corpo celular para que sejam enviadas novamente a outros neurônios. Esse processo de comunicação entre as células nervosas é conhecido como sinapses Graupe (2007), Bezerra (2016b).

Figura 2 – Neurônio biológico

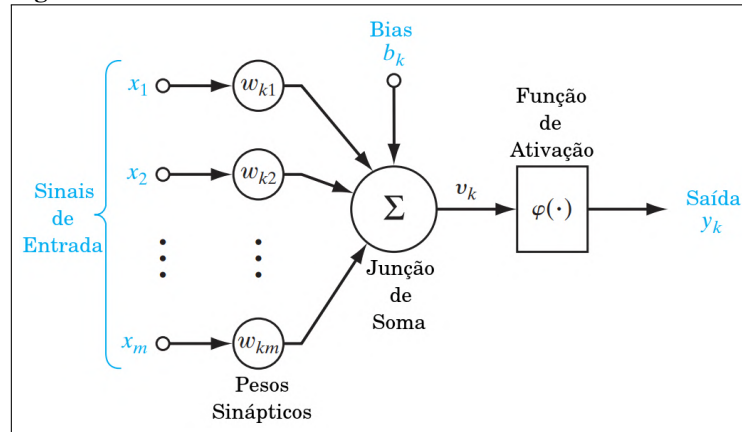


Fonte: (BEZERRA, 2016b)

De acordo com Haykin S. (2009), uma rede neural artificial é semelhante ao cérebro humano, pois é capaz de adquirir conhecimento por meio de um processo de aprendizagem. Além disso, há conexões internas, conhecidas como pesos sinápticos, que são aprendidos durante a fase de treinamento. De forma simplificada, uma rede é formada por neurônios que estão conectados uns aos outros na rede e possuem pesos nas conexões.

A Figura 3 apresenta um modelo de neurônio artificial, que recebe vários sinais de entrada e retorna apenas um sinal na saída. Nesse neurônio, as sinapses são caracterizadas por

Figura 3 – Neurônio artificial



Fonte: Adaptado de (HAYKIN S., 2009)

pesos sinápticos. Assim, para cada sinal de entrada x_j de uma sinapse j conectada a um neurônio k , um peso sináptico w_{kj} é multiplicado. Os sinais de entrada calculados são somados para produzir uma única saída. Este modelo também apresenta uma polarização ou *bias* de entrada b_k , que é uma variável incluída no somatório para aumentar ou diminuir a entrada da função de ativação. O *bias* é aprendido durante o treinamento da rede e pode ter seus valores alterados durante esse processo. Com isso, o neurônio artificial k pode ser descrito pelas Equações 2.1 e 2.2,

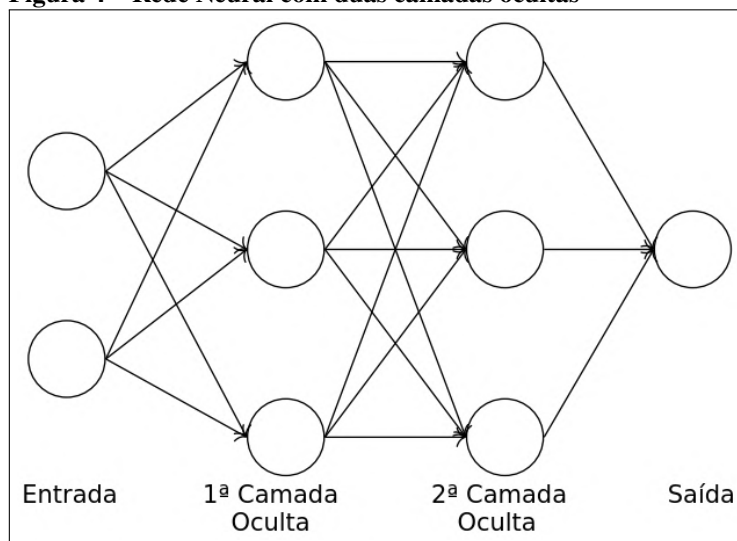
$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (2.1)$$

$$y_k = \varphi(u_k + b_k); \quad (2.2)$$

em que x_j representa os sinais de entrada e w_{kj} representa os pesos sinápticos, u_k é a saída do somatório dos sinais de entrada, φ é a função de ativação e y_k a saída do neurônio k . A função de ativação é utilizada para limitar o valor da saída do neurônio a partir de um intervalo, por exemplo, mapeando os valores de saída para 0 ou 1, a Figura 5 apresenta as principais funções de ativação que podem ser utilizadas Haykin S. (2009), Shalev-Shwartz e Ben-David (2014).

Uma rede neural é formada por diversos neurônios que podem ser organizados em diferentes modelos, entre eles há a arquitetura de Rede *Feedforward*. Kriesel (2007) define uma rede neural do tipo *Feedforward* como uma rede com camadas claramente separadas entre camada de entrada, camada de saída e camadas ocultas, sendo que cada neurônio só pode direcionar suas conexões para neurônios da camada seguinte. A forma mais simples de uma rede *Feedforward* é conhecida como Rede *Single-Layer Feedforward* ou Rede de Camada Única, uma vez que os nós de entrada são projetados diretamente nos nós saída. Além disso, também existem as Redes *Multi-Layer Feedforward* ou Redes de Múltiplas Camadas, que contêm camadas ocultas, com a função de extrair características de ordem superior da sua entrada, intervindo entre a camada de entrada e saída (HAYKIN S., 2009; DENG; YU, 2014). Na Figura 4 um modelo de rede do tipo *Multi-Layer Feedforward* é apresentado, contendo duas camadas ocultas.

Figura 4 – Rede Neural com duas camadas ocultas



Fonte: Adaptado de (DENG; YU, 2014)



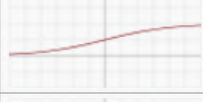
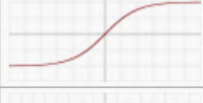
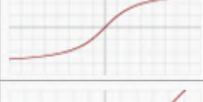



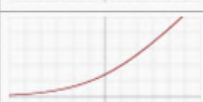
2.1.2 Deep Learning

Outro conceito importante de *machine learning* é chamado de *deep learning* ou aprendizado profundo, que deve ser crescimento através da melhoria de capacidade de *hardware* e no aumento da quantidade de dados disponíveis. Que se baseia nas redes neurais, com muitas camadas e neurônios e treinadas com um grande número de dados. Essa técnica poder ser muito útil para prever estruturas complexas em dados de alta dimensão, podendo ser aplicada a diversas aplicações como reconhecimento de fala, reconhecimento de objetos visuais, detecção de objetos, entre outras (LECUN; BENGIO; HINTON, 2015).

Deng e Yu (2014) definiram *Deep Learning* como uma classe de técnicas de *machine learning* que explora várias camadas de processamento de informações para extração e transformação de recursos supervisionados ou não supervisionados, e utilizados na análise e classificação de padrões. Essas técnicas permitem que modelos computacionais com múltiplas camadas aprendam uma representação de dados em camadas mais abstratas. LeCun, Bengio e Hinton (2015) dizem que métodos de *Deep Learning* aprendem por meio de múltiplos níveis de representação, que transformam a representação da entrada de dados em uma representação em um nível mais abstrato, possibilitando o aprendizado de funções muito complexas.

O conceito por trás de *Deep Learning* inclui as áreas de pesquisa de redes neurais, inteligência artificial, otimização e reconhecimento de padrões, com isso o avanço dessa área se deu devido a melhoria da capacidade de processamento computacional, possibilidade de aumentar a quantidade de dados usados para treinamento e avanços em *machine learning*. Exemplos de técnicas de *Deep Learning* são as redes neurais convolucionais e redes *autoencoder* que são descritas nas Seções 2.2 e 2.3

Figura 5 – Funções de Ativação

Nome	Gráfico	Equação
Identity		$f(x) = x$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Logistic (a.k.a. Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$
Tanh		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$
ArcTan		$f(x) = \tan^{-1}(x)$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU)		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU)		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$

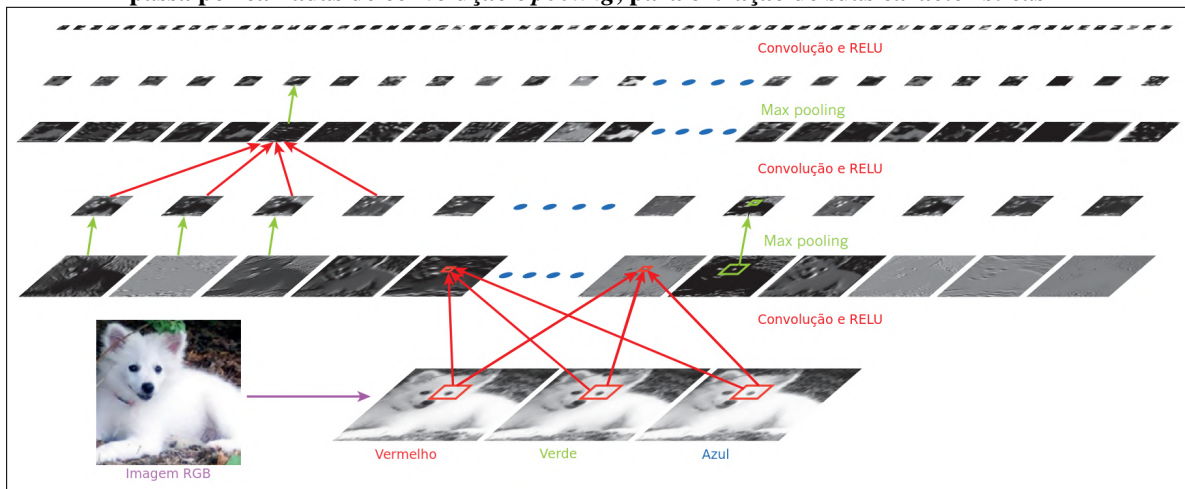
Fonte: Adaptado de (SHARMA, 2017)

2.2 REDES NEURAIIS CONVOLUCIONAIS

Uma Rede Neural Convolutiva ou *Convolutional Neural Network* (CNN) é caracterizada pela utilização de operações de convolução em pelo menos uma de suas camadas, com objetivo de aprender padrões de um determinado conjunto de dados. Uma CNN trabalha com dados no formato de múltiplos *arrays* e distribuídos em forma de uma estrutura de grade tridimensional, representadas por largura l , altura a e profundidade p , ou seja, cada neurônio possui três dimensões $l * a * p$. Nesse caso, a profundidade não é da rede toda, mas de apenas uma camada convolutiva. Por exemplo, para dados de uma imagem RGB, a profundidade da camada de entrada será o número de canais de cores dessa imagem, no caso 3, com isso pode-se dizer que a imagem é analisada como sendo três camadas separadas de cores empilhadas umas sobre as outras. Conforme os filtros de convolução vão sendo aplicados nas camadas de uma CNN, as

suas dimensões sofrem alterações, para N filtros aplicados em uma camada as novas dimensões serão $l' * a' * N$ (LECUN; BENGIO, 1995; GOODFELLOW; BENGIO; COURVILLE, 2016).

Figura 6 – Camadas de uma CNN. Neste exemplo, a rede recebe como entrada uma imagem RGB, que passa por camadas de convolução e *pooling*, para extração de suas características



Fonte: Adaptado de (LECUN; BENGIO; HINTON, 2015)

Na Figura 6 é apresentado um exemplo de CNN, adaptado de (LECUN; BENGIO; HINTON, 2015). Nesta figura são ilustradas as saídas de dois tipos de camada da CNN: convolução e *pooling*. Geralmente uma camada de uma CNN é composta por três estágios, sendo que no primeiro operações de convolução são executadas para produzir um conjunto de ativações lineares, que são executadas por meio de uma função de ativação não linear, como a função de ativação ReLU. No terceiro estágio, uma operação conhecida como *pooling* é utilizada para modificar a saída da camada (GOODFELLOW; BENGIO; COURVILLE, 2016). Nas próximas seções serão descritas os tipos de camadas de uma CNN utilizadas neste trabalho.

2.2.1 Camada Convolutacional

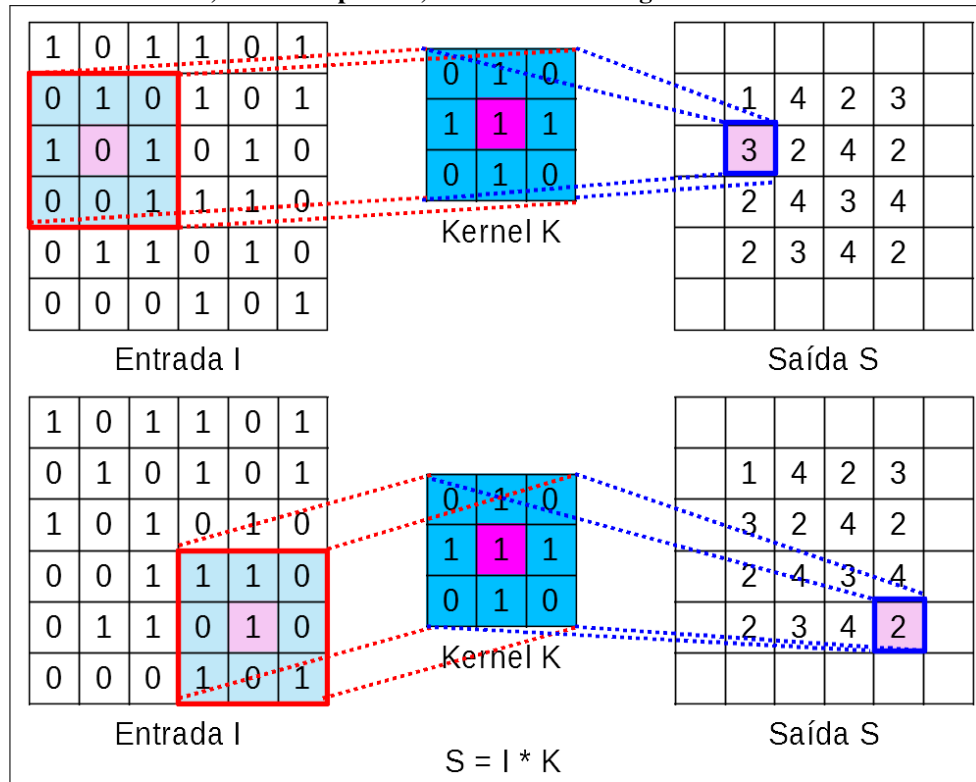
Uma CNN utiliza uma operação de convolução em uma ou mais de suas camadas, de forma geral, uma operação de convolução pode ser denotada na Equação 2.3:

$$s(x) = (f * w) = \sum_{a=-\infty}^{\infty} f(a)w(t - a); \quad (2.3)$$

onde o primeiro argumento, representado por x , é a entrada e w é o *kernel*, que representa um filtro a ser aplicado na imagem (GOODFELLOW; BENGIO; COURVILLE, 2016).

Quando se trabalha com imagens, uma camada de convolução é formada um conjunto de filtros. Na Figura 7 um filtro é aplicado na imagem I , representado por um *kernel* K de tamanho 3×3 . Esse filtro passa por cada *pixel* de I produzido uma nova representação da imagem na saída S , vale ressaltar que na Figura 7 o filtro aplicado não tem tratamento para

Figura 7 – Exemplo de convolução. Em uma imagem representada por I é aplicado um filtro de convolução 3×3 K . Para cada *pixel* da imagem, com exceção das bordas, o filtro é aplicado, resultando na imagem S



Fonte: Autoria própria

bordas, logo os *pixels* da borda são desconsiderados, gerando uma saída com altura e largura menor do que a da entrada. Dessa forma, para cada filtro utilizado na imagem haverá um mapa de características, resultante da convolução da imagem e do *kernel* (DENG; YU, 2014). Assim, a operação de convolução para entradas multidimensionais é indicada na Equação 2.4, em que f é a imagem e w o *kernel* (GOODFELLOW; BENGIO; COURVILLE, 2016).

$$s(x, y) = (f * w)(x, y) = \sum_m \sum_n f(m, n)w(x - m, y - n) \quad (2.4)$$

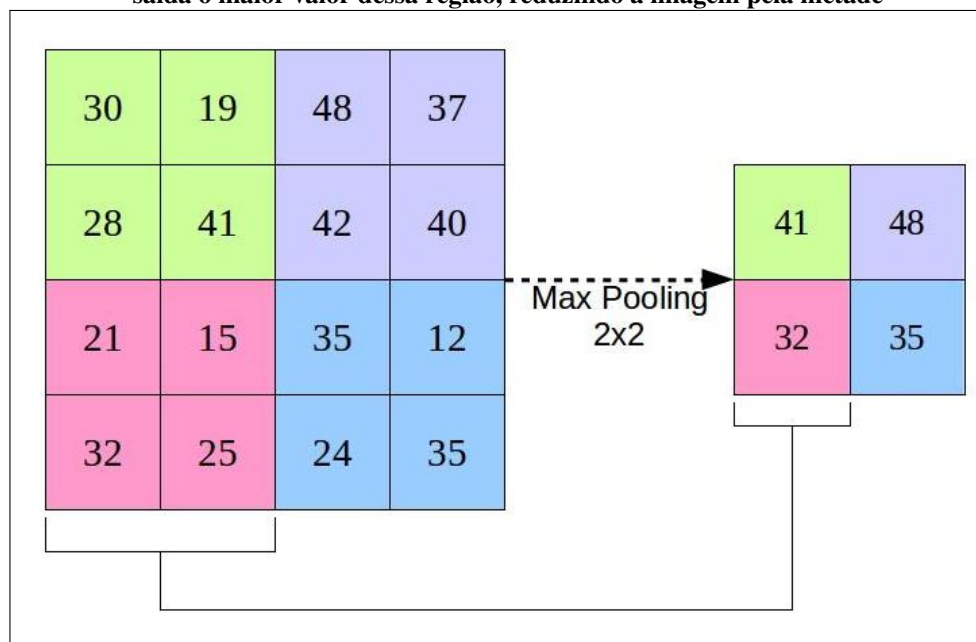
2.2.2 Camada de Pooling

A camada de *pooling* ou agrupamento é utilizada para reduzir o tamanho do mapa de características produzido na camada anterior, de forma que o número de parâmetros na rede seja menor e que o custo computacional diminua. Vale ressaltar, que essa operação é realizada em cada mapa de características presente na profundidade da entrada, por exemplo, em cada canal de cor de uma imagem. Com isso, apenas a largura e altura no mapa de características é afetada Goodfellow, Bengio e Courville (2016), LeCun, Bengio e Hinton (2015).

Uma operação bastante comum é o *pooling* máximo ou *max pooling*, que substitui

na saída o valor máximo dentro de uma região da imagem. Na Figura 8 há um exemplo de como funciona essa operação, para um filtro de tamanho 2×2 . Assim, para cada região 2×2 , o *pixel* equivalente na imagem de saída recebe o maior dos quatro valores da região. Neste caso, a saída tem a metade do tamanho espacial da entrada. Além disso, também existem outras formas de se realizar o *pooling*, como através da média de uma vizinhança retangular, a norma de uma vizinhança retangular ou uma média ponderada baseada na distância do *pixel* central. (GOODFELLOW; BENGIO; COURVILLE, 2016).

Figura 8 – Max Pooling. Para cada região 2×2 da imagem de entrada, é substituído na saída o maior valor dessa região, reduzindo a imagem pela metade



Fonte: Autoria própria

2.2.3 Batch Normalization

A normalização de amostras é uma maneira de torná-las mais semelhantes entre si, para que possa auxiliar um determinado modelo de aprendizado de máquina a aprender e generalizar melhor para novos dados. Isso é realizado por meio de alterações nos valores dos dados para uma escala em comum, sem que sejam distorcidas diferenças nos intervalos dos valores e sem perder informações. Um exemplo de normalização utilizada em redes neurais é a normalização em lotes ou *Batch Normalization*. (CHOLLET, 2018).

A técnica conhecida como *Batch Normalization* foi desenvolvida por (IOFFE; SZEGEDY, 2015) para o tratamento de problemas de inicialização da rede. Esses problemas podem ocorrer durante o treinamento, visto que ocorrem ajustes nos pesos e parâmetros aplicados ao conjunto de dados, assim, para cada iteração, a distribuição das entradas nas camadas mudam durante o treinamento, exigindo que taxas menores de aprendizagem sejam utilizadas, bem

como reduzindo a velocidade do treinamento. Para isso, Ioffe e Szegedy (2015) propuseram normalizar as saídas de uma camada de ativação em uma distribuição Gaussiana. Essa técnica normaliza adaptativamente os dados, mesmo que a média e a variação se alterem ao longo do tempo durante o treinamento, mantendo uma média móvel exponencial da média em lotes e variação dos dados vistos durante o treinamento. O que auxilia na propagação do gradiente permitindo redes mais profundas (CHOLLET, 2018).

2.2.4 Dropout

Durante o treinamento a rede neural realiza os processo de otimização e generalização. O primeiro consiste em ajustar a rede para obter o melhor desempenho possível com os dados de treinamento. Já a generalização está relacionada com o desempenho da rede treinada para dados nunca antes vistos. Ou seja, o treinamento tem como objetivo obter uma boa generalização, com base na otimização, ajustando a rede a partir dos dados de treinamento (CHOLLET, 2018). No entanto, se forem realizados ajustes demais na rede para os dados de treinamento, existe a possibilidade de ajustar o ruído nos dados, aprendendo padrões específicos das amostras, em vez de encontrar uma regra preditiva geral. Esse problema é conhecido como *overfitting* (DIETTERICH, 1995).

Pensando no *overfitting*, Srivastava *et al.* (2014) desenvolveram um método de regularização chamado de *Dropout*. Essa técnica consiste em descartar aleatoriamente dados de uma rede neural durante o treinamento, evitando que a rede se adapte demais. Por exemplo, se uma camada de uma rede neural retornar, para uma determinada amostra de entrada do treinamento, o vetor $[0, 2; 0, 5; 1, 3; 0, 8; 1, 1]$, aplicando o *Dropout* o novo vetor poderia ser $[0; 0, 5; 1, 3; 0; 1, 1]$, uma vez que terá dados nulos distribuídos de forma aleatória. (CHOLLET, 2018; SRIVASTAVA *et al.*, 2014). De acordo com Srivastava *et al.* (2014), a remoção aleatória de um subconjunto diferente de neurônios reduz o *overfitting*, quebrando padrões que não são significativos para o aprendizado da rede.

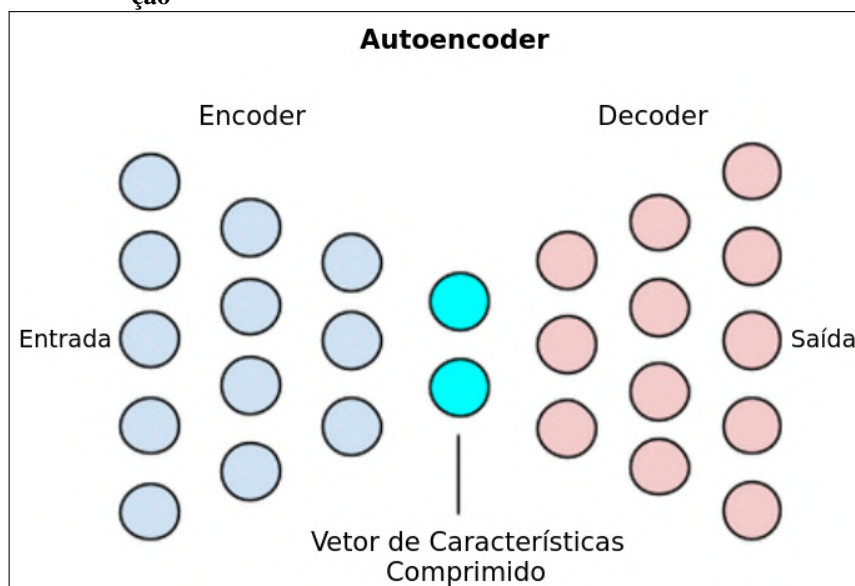
2.3 REDE AUTOENCODER

Uma rede neural do tipo *autoencoder* ou autocodificadora (AE) utiliza aprendizagem não-supervisionada para aprender por meio de um conjunto de dados. Esse tipo de rede, basicamente, reproduz, de forma mais próxima possível, os dados de entrada na sua saída, por meio de uma função codificadora e decodificadora, que é responsável por reproduzir a entrada. A princípio isso pode parecer desnecessário, no entanto, um *autoencoder* não aprende a copiar a entrada de forma perfeita, mas sim aproximada. Dessa forma, a rede acaba aprendendo a

priorizar as propriedades mais importantes dos dados de entrada (GOODFELLOW; BENGIO; COURVILLE, 2016; BEZERRA, 2016a).

A arquitetura de um *autoencoder* consiste em no mínimo 3 camadas: entrada, saída e camada oculta, sendo que a entrada e a saída são de mesmo tamanho, conforme apresentado na Figura 9. Esse tipo de rede tem a capacidade de reconstruir a sua entrada na saída por meio de duas funções: codificadora e decodificadora. A função codificadora ou *encoder* é responsável por extrair as características dos dados, produzindo um vetor comprimido de características. Esse vetor será a entrada da função decodificadora ou *decoder* que traduz esse vetor de forma a produzir a entrada, de forma que o erro de reconstrução seja minimizado, uma vez que os dados são traduzidos de forma que se assemelhem aos dados de treinamento (BEZERRA, 2016a; NETO *et al.*,).

Figura 9 – Exemplo de Autoencoder. Composto por uma função codificadora, que comprime o vetor de entrada. Em seguida a função decodificadora irá reconstruir esse vetor, minimizando erros de reconstrução



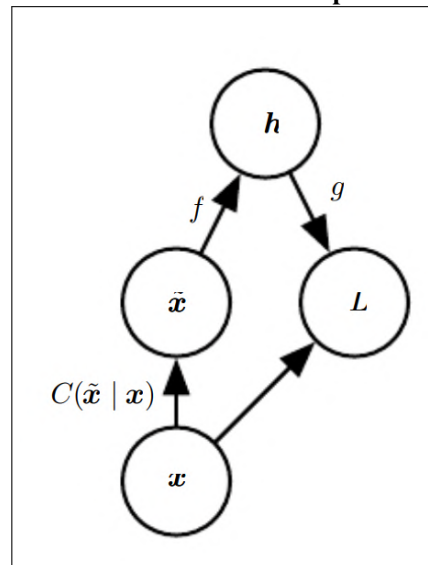
Fonte: Adaptado de (NETO *et al.*,)

Há diferentes variações de uma rede *autoencoder*, como 1) *Denosing Autoencoder*, que produz uma representação robusto a ruídos; 2) *Contractive Autoencoder*, que tem como objetivo evitar representações indesejadas, adicionando um termo a função de perda; e 3) *Sparse Autoencoder*, que fazem com que uma pequena quantidade de unidades da camada oculta seja ativada em cada padrão de entrada (BEZERRA, 2016a).

Uma rede do tipo *Denosing Autoencoder* ou DAE é uma variação de *autoencoder* tradicional, que recebe dados corrompidos como entrada e é treinado para prever os mesmos dados não corrompidos na sua saída. Na Figura 10 é apresentado um modelo simples do processo de treinamento de um DAE, retirado de (GOODFELLOW; BENGIO; COURVILLE, 2016). Considerando que a a função codificadora é representada por $h = f(x)$ e a decodificadora é representada por $r = g(h)$, a rede recebe como entrada um conjunto de dados representado por

x , o qual será introduzido um processo de corrupção $C(\tilde{x}|x)$. Dessa forma uma estrutura de reconstrução é aprendida em pares $(x|\tilde{x})$, ou seja, por meio de um exemplo de x e uma amostra da versão corrompida \tilde{x} , sendo $L(x, g(f(\tilde{x})))$ uma função de perda minimizadora (GOODFELLOW; BENGIO; COURVILLE, 2016).

Figura 10 – Denoising Autoencoder. Esse tipo de AE corrompe os vetor de entrada e produz um vetor decorrumpido



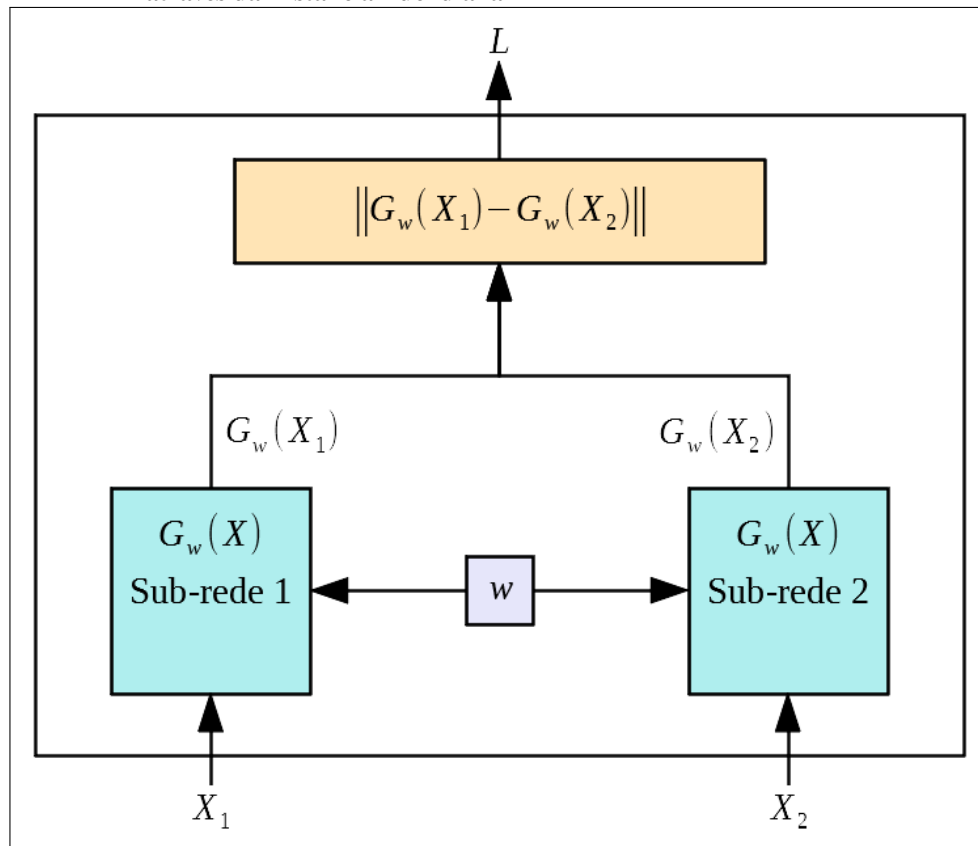
Fonte: (GOODFELLOW; BENGIO; COURVILLE, 2016)

2.4 REDES NEURAI SIAMESAS

Segundo Bromley *et al.* (1994), uma rede siamesa é formada por duas sub-redes idênticas que são unidas em suas saídas e foi inicialmente apresentada para buscar correspondências em imagens contendo assinaturas. Assim, cada sub-rede recebe uma entrada diferente que é mapeada para um descritor de característica, com isso dois descritores são obtidos e comparados para verificar a semelhança entre eles, resultando na saída da rede. Vale ressaltar que é necessário que as sub-redes compartilhem os mesmos parâmetros e pesos para tornar a saída de cada uma delas comparável.

Na Figura 11 é apresentado um exemplo de rede siamesa, retirado de Koch, Zemel e Salakhutdinov (2015). Sendo X_1 e X_2 um par de imagens de entrada, que recebem um rótulo binário Y para identificar se são semelhantes ou não, o conjunto de funções $G(X)$, um vetor de parâmetros W compartilhado e $G_w(X_1)$ e $G_w(X_2)$ o mapeamento das entradas dadas X_1 e X_2 , respectivamente. Com isso, a rede neural siamesa durante o seu treinamento busca por um valor do parâmetro W que encontre uma menor diferença entre X_1 e X_2 . Para isso, uma função para

Figura 11 – Rede Neural Siamesa. Esse modelo de rede neural é formado por duas sub-redes que compartilham parâmetros w e são unidas em suas saídas. Cada sub-rede recebe uma imagem de entrada, que é mapeada para um vetor de características. Em seguida compara-se os vetores de saída de cada sub-rede através da Distância Euclidiana



Fonte: Adaptado de Chopra *et al.* (2005)

calcular essa diferença é utilizada, dada pela Equação 2.5, que é passada para uma função de Perda Contrastiva L (BROMLEY *et al.*, 1994; KOCH; ZEMEL; SALAKHUTDINOV, 2015).

$$E_W = |G_w(X_1) - G_w(X_2)| \quad (2.5)$$

A função de perda *Contrastive Loss* é utilizada para medir a capacidade da rede de encontrar as semelhanças entre as imagens. Assim, essa função deve aprender os parâmetros W , de forma que os exemplos mais semelhantes fiquem mais próximos e os mais diferentes sejam separados. Para garantir que isso ocorra, durante o treinamento cada par de imagens recebem um rótulo Y , sendo $Y = 0$ quando são semelhantes, ou seja, de um mesmo grupo e $Y = 1$ quando são diferentes. A Perda Contrastiva é dada pela Equação 2.6

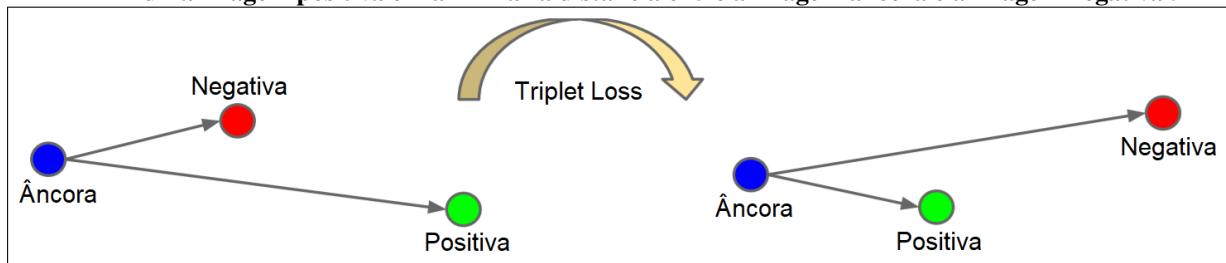
$$L(W, Y, X_1, X_2) = (1 - Y) \frac{1}{2} (E_W)^2 + (Y) \frac{1}{2} \{ \max(0, m - E_W) \}^2 \quad (2.6)$$

onde $m > 0$ é a margem, tal que (X_1, X_2) seja um par de entradas positivos e (X_1, X_2') seja um par negativo, temos que $E_W(X_1, X_2) + m < E_W(X_1, X_2')$ (HADSELL; CHOPRA; LECUN, 2006).

2.4.1 Triplets

Um outro modelo de Rede Neural Siamesa é composto por Redes Neurais *Triplets*, que utilizam como alternativa ao *Contrastive Loss*, a função de perda *Triplet Loss*. A principal diferença entre elas é que a Rede *Triplet* possui três sub-redes idênticas, sendo assim, recebe três entradas, sendo elas: imagem âncora x_i^a , imagem positiva x_i^p e imagem negativa x_i^n , sendo que x_i^a e x_i^p possuem a mesma identidade e x_i^n é de uma identidade diferente. A Rede *Triplet* deve determinar a relação de similaridade relativa para as três imagens, através da ordenação de similaridade dos *triplets* (WANG *et al.*, 2014; SCHROFF; KALENICHENKO; PHILBIN, 2015).

Figura 12 – *Triplet Loss* deve minimizar a distância, representada pelas setas, entre uma imagem âncora e uma imagem positiva e maximizar a distância entre a imagem âncora e a imagem negativa .



Fonte: Adaptado de Schroff, Kalenichenko e Philbin (2015)

De acordo com Hoffer e Ailon (2015), a partir de três entradas, a rede deve gerar dois valores intermediários de distância entre representações de duas de suas entradas e a representação da terceira. Em outras palavras, a rede deve estabelecer as distâncias entre as imagens âncora e positiva, assim como, entre as imagens âncora e negativa. Como pode ser observado na Figura 12, a função de perda *Triplet Loss* é motivada pela classificação do n -vizinho mais próximo e tem como objetivo tornar a distância entre a âncora e a imagem positiva menor, bem como aumentar a distância com a imagem negativa. Com isso, considerando que $f(x) \in \mathbb{R}$ seja a representação incorporada de cada sub-rede, temos que

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2, \forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in \tau \quad (2.7)$$

para α sendo uma margem e τ um conjunto de todos os *triplets*.

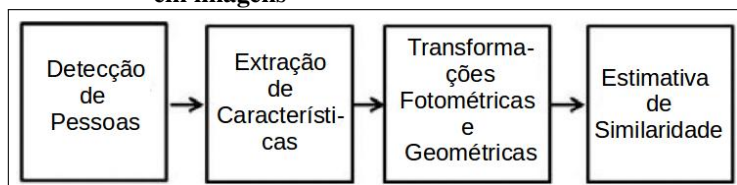
Neste trabalho, foi utilizada uma Rede Neural Siamesa, bem como uma Rede Neural *Triplet*, com o mesmo modelo de sub-rede, para realizar a re-identificação de pessoas em imagens. Além disso, foi realizada uma comparação entre essas duas abordagens de *deep learning*, utilizando três *datasets* diferentes.

3 TRABALHOS RELACIONADOS

Neste capítulo serão apresentados alguns trabalhos disponíveis na literatura que tratam do problema de re-identificar pessoas em imagens digitais, utilizando descritores de características e aprendizagem de máquina. Considerando que o processo para re-identificar pessoas em imagens tem como tarefa determinar se uma pessoa em uma imagem é a mesma pessoa que aparece em outra imagem diferente, podendo ser da mesma câmera ou não e com diversas variações de luminosidade, perspectiva e até mesmo oclusões, um passo inicial importante é de obter informações de característica de cada indivíduo, em seguida deve-se verificar as correspondências entre as características de cada pessoa (AHMED; JONES; MARKS, 2015).

A Figura 13 apresenta as etapas básicas no processo de re-identificação de pessoas. Inicialmente deve-se detectar automaticamente os pedestres, para segmentar as pessoas de uma determinada cena, seguida da extração de características que servem para identificar e diferenciar cada pessoa nas imagens. No entanto, essas características podem sofrer diversas transformações devido à condições de iluminação e configuração das câmeras. Além disso, transformações geométricas, caracterizadas por mudanças de pontos de vista e posição dos pedestres, também podem ocorrer. Isso pode tornar imagens de uma mesma pessoa muito diferentes, assim, medidas de similaridade podem ser aprendidas para encontrar pessoas correspondentes em imagens diferentes (LI *et al.*, 2014).

Figura 13 – Principais etapas para a re-identificação de pessoas em imagens



Fonte: Adaptado de (LI *et al.*, 2014)

Gray e Tao (2008) desenvolveram um método, chamado de ELF, que trata as transformações geométricas, considerando a invariância de pontos de vista para reconhecimento de pedestres. O método se baseia na combinação de histogramas e características da imagem e utilização de aprendizagem de máquina para aprender um modelo de características mais representativo nos dados de treinamento. Isso pode ser feito combinando os canais de cores RGB, YCbCr e HSV1 com características de textura da imagem. Considerando essas informações, aplica-se o algoritmo AdaBoost que aprende um modelo contendo o conjunto de características mais significativas, durante o treinamento. Com isso, uma função de similaridade é aprendida dos dados de treinamento, para ser aplicada a re-identificação de pedestres, com variação no ponto de vista das imagens. Utilizando o conjunto de imagens ViPER (GRAY; BRENNAN; TAO, 2007), Gray e Tao (2008) concluíram que as características mais significativas foram a matiz e saturação, devido as variações de luminosidade entre as imagens de duas câmeras. Esse

método não resolve o problema da re-identificação automática de pessoas, no entanto, por meio da função de similaridade auxilia um operador humano no reconhecimento de pedestres em imagens de câmeras diferentes, reduzindo cerca de 82% no tempo de pesquisa pelo operador.

Prosser *et al.* (2010) também consideraram as características mais representativas de cada imagem. No entanto, o problema da re-identificação foi reformulado para ser um problema de classificação, assim, não há a necessidade de utilizar uma medida de distância entre os vetores de características de cada imagem, nem a limiarização entre essas distâncias entre imagens com correspondências positivas ou negativas. Neste método, um vetor contendo as características de cada imagem é obtido e durante o treinamento é associado a um conjunto de características relevantes e um vetor com características irrelevantes, ou seja, por meio de correspondências positivas e negativas. Cada imagem foi dividida em 6 retângulos, de forma a representar a cabeça, o tronco superior e inferior e as pernas superiores e inferiores de cada indivíduo, as características analisadas foram 8 canais de cores (RGB, HSV1 e YCbCr) e 21 filtros de textura aplicados ao canal de luminância. Para a classificação foi utilizado o método de ranqueamento RankSVM, que atribui às correspondências positivas pontuações entre os vetores. Foram realizados testes utilizando os *datasets* VIPeR (GRAY; BRENNAN; TAO, 2007) e i-LIDSVID (WANG *et al.*, 2014). Apesar de se obter melhores resultados do que o proposto por Gray e Tao (2008), este método pode se tornar muito custoso computacionalmente, uma vez que pode gerar grande número de amostras negativas. Além disso, o método não trata correlações entre características diferentes, nem grandes dispersões entre os dados.

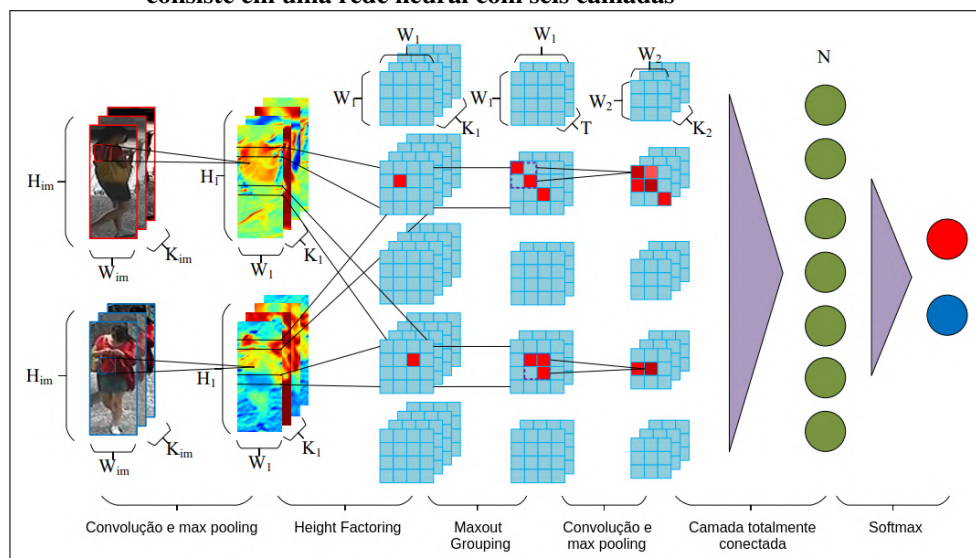
Outro método para re-identificar pessoas é baseado na extração de características e foi desenvolvido por Zhao, Ouyang e Wang (2013). Neste caso, são consideradas características extraídas de regiões salientes, descritas pelos autores como discriminativas e confiáveis, uma vez que são regiões que tornam uma pessoa distinta da outra e que devem estar presentes nas imagens mesmo com variações no ponto de vista. Uma saliência pode ser a cor da bolsa que uma pessoa está carregando, por exemplo. Com isso, essas informações são extraídas por histograma de cores LAB e descritor SIFT de cada pedaço de 10×10 *pixels* na imagem. A correspondência entre esses *pedaços* é determinada com base na distância Euclidiana, por meio da aprendizagem não supervisionada, utilizando o algoritmo do K vizinho mais próximo (KNN). Assim, o algoritmo busca encontrar em um conjunto de imagens aquela com a menor distância. Os testes foram realizados utilizando os *datasets* VIPeR (GRAY; BRENNAN; TAO, 2007) e ETHZ. Com os testes foi observado que o método empregado pode auxiliar no processo de re-identificação de pessoas com variações no ponto de vista, no entanto, em alguns casos as regiões de saliência podem ser tratadas como *outliers* pelo algoritmo.

Dentre as técnicas para a re-identificação de pessoas destaca-se o *Deep Learning*. Nesta seção serão apresentados alguns trabalhos que utilizam métodos de *Deep Learning* para tratar deste problema.

Wu *et al.* (2016) propuseram um método para extração de características imagens, com-

binando uma rede neural convolucional com o método proposto por Gray, Brennan e Tao (2007) chamado de Feature Fusion Network. O método consiste em uma rede neural convolucional formada por duas partes, sendo a primeira contendo camadas de convolução e agrupamento para extrair características de uma imagem. A segunda etapa utiliza uma alteração do método ELF desenvolvido por (GRAY; TAO, 2008) para extrair características com base nos canais de cores RGB, YCbCr, HSV1 e textura da imagem. A alteração do método consiste em dividir a imagem de entrada em 16 faixas para extração das características serem realizadas em cada faixa, produzindo, assim, um histograma com 16 dimensões para cada canal de cor, que será concatenado no final para formar um único vetor. Wu *et al.* (2016) chamaram o método modificado de ELF16. Depois disso, os vetores extraídos em cada uma das etapas devem ser combinados para se obter uma representação mais completa da imagem. Com isso, foi observado que o Feature Fusion Network torna a rede neural convolucional mais completa na extração de características do que o método desenvolvido por Gray, Brennan e Tao (2007).

Figura 14 – Arquitetura proposta por Li *et al.* (2014). O método chamado de *DeepReID* consiste em uma rede neural com seis camadas



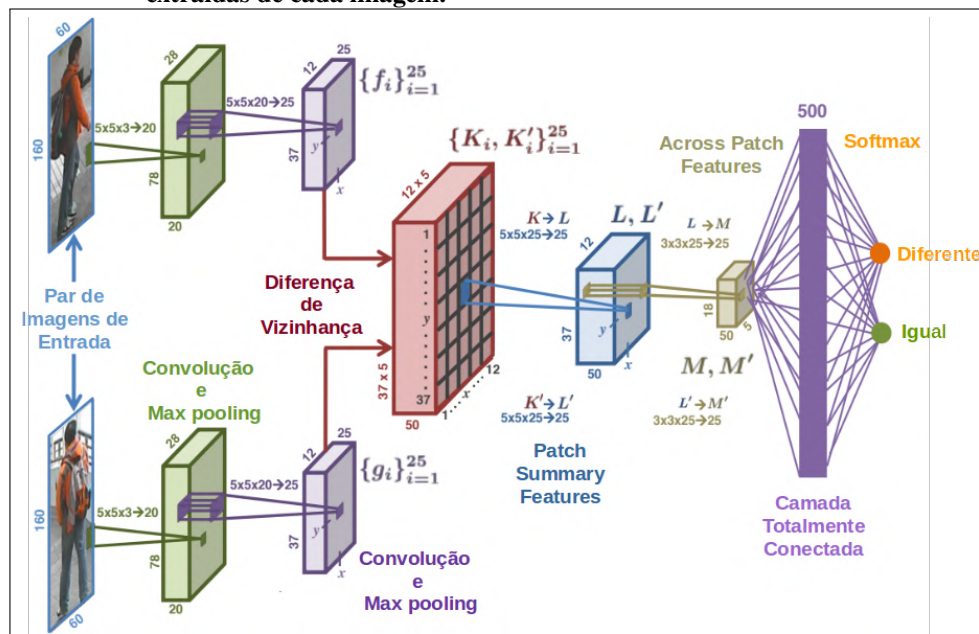
Fonte: Adaptado de (LI *et al.*, 2014)

Um dos primeiros trabalhos a utilizar *Deep Learning* para re-identificar pessoas em imagens digitais foi proposto por Li *et al.* (2014). O método chamado de *DeepReID: Deep Filter Pairing Neural Network* consiste em uma rede que aprende automaticamente recursos para identificação de pessoas, levando em consideração possíveis transformações que podem ocorrer nas imagens e interferências do *background*. A rede proposta possui seis camadas, tal como mostrado na Figura 14. Sendo a primeira formada por duas sub-redes compostas por uma camada de convolução com *max pooling*, em que cada sub-rede processa uma imagem diferente, a dimensão das imagens de entrada é representada por $W_{im} * H_{im} * K_{im}$. A segunda camada divide cada mapa de característica produzido na etapa anterior em M faixas horizontais para realizar a correspondência entre dois mapas de características nas faixas correspondentes, essa etapa é indicada na Figura 14 por *Height Factoring*. A próxima camada é conhecida como

maxout-grouping para tornar a rede mais robusta a variações de iluminação. Após essa etapa, os mapas de características passam novamente por uma camada de convolução de *max pooling* e uma camada totalmente conectada. Por fim, mais uma camada totalmente conectada com a função de ativação *softmax*. Além disso, Li *et al.* (2014), para o treinamento de seu método, criou um conjunto de imagens, chamado de CUHK03, com 13.164 imagens de 1.360 pedestres diferentes, obtidas por 6 câmeras, até então o maior conjunto de dados para re-identificação de pessoas da literatura.

Ahmed, Jones e Marks (2015) utilizaram uma rede neural siamesa para aprender medidas de similaridade entre duas imagens. A rede desenvolvida é apresentada na Figura 15. Como pode ser observado, a rede é composta por duas sub-redes, em que cada uma recebe uma imagem de entrada. Vale ressaltar que essas sub-redes contêm parâmetros idênticos, visto que as características obtidas de cada imagem serão comparadas posteriormente. Uma sub-rede, contém duas camadas de convolução, que são utilizadas para obter as características da imagem. Como ilustrado na Figura 15, dada uma imagem RGB de tamanho 160×60 como entrada, a primeira camada de convolução aplica 20 filtros de tamanho 5×5 , em seguida, utiliza a operação de *Max-pooling*, reduzindo o tamanho da imagem que será a entrada da segunda camada. Essa camada aplica 25 filtros de tamanho 5×5 , seguida da operação de *Max-pooling*. Com isso, resultou-se em 25 mapas de características de tamanho 12×37 .

Figura 15 – Arquitetura proposta por Ahmed, Jones e Marks (2015). As camadas iniciais são responsáveis pela extração de características das imagens individualmente. As camadas conectadas calculam as relações entre as características extraídas de cada imagem.



Fonte: Adaptado de (AHMED; JONES; MARKS, 2015)

Após se obter os mapas de características de ambas as imagens, a próxima camada calcula as diferenças de vizinhança entre os mapas de características correspondente das imagens. Essa operação resultará em 25 mapas de diferença de vizinhança. A camada seguinte, identifi-

cada por *Patch Summary Features* produz uma representação local de alto nível para cada um dos mapas de diferenças de vizinhança, calculando uma relação entre as características das duas imagens, produzindo uma representação de cada bloco 5×5 . Recebendo um mapa de tamanho $12 \times 37 \times 5 \times 5$ e resultando em um mapa de tamanho 12×37 . Considerando que os mapas de características obtidos fazem parte de um conjunto L de tamanho $12 \times 37 \times 25$, a próxima camada, realiza a convolução de L com 25 filtros de tamanho 3×3 , com objetivo de aprender relações espaciais entre as diferenças de vizinhança, resultando em um conjunto M , de tamanho $5 \times 18 \times 25$. Aplica-se uma camada totalmente conectada, resultando em um vetor de características de tamanho 500. Por fim, outra camada totalmente conectada é utilizada com a função *softmax* para calcular a probabilidade do par de imagens serem da mesma pessoa.

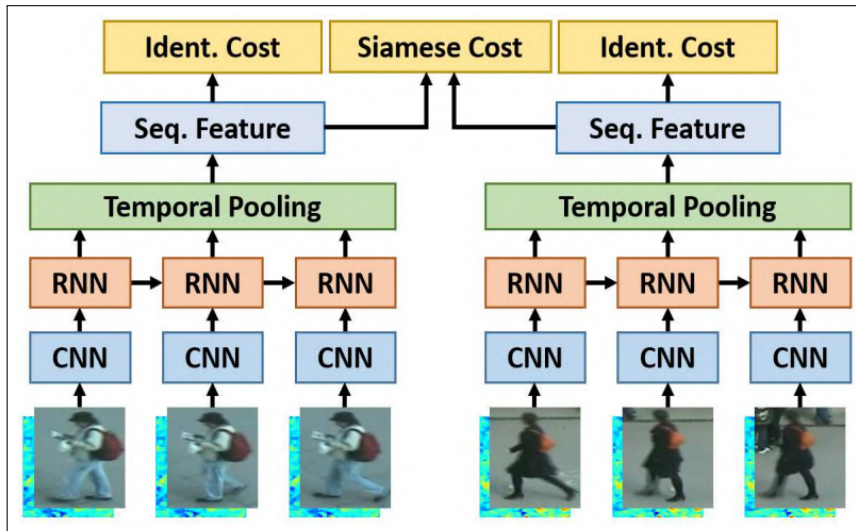
Ahmed, Jones e Marks (2015) testaram a rede proposta em 3 *datasets*: VIPeR (GRAY; BRENNAN; TAO, 2007), CUHK03 (LI *et al.*, 2014) e CUHK01 (LI; ZHAO; WANG, 2012). No entanto, não foi utilizado nenhum *dataset* do tipo *multi-shot*, que contém imagens sequências de cada pedestre. Além disso, 5 redes também foram treinadas, considerando 5 partes do corpo dos pedestres nas imagens, para tratamento de oclusões. A região da cabeça foi a que obteve melhores resultados, porém não foi desenvolvido nenhum método para se verificar a re-identificação utilizando algum tipo de relação entre os resultados obtidos com a análise das 5 partes do corpo.

Uma rede neural siamesa foi utilizada por Yi *et al.* (2014) para encontrar a similaridade em duas imagens. A rede proposta possui duas redes neurais convolucionais. Cada rede neural convolucional, neste caso, é formada por 2 camadas de convolução ativadas pela função RELU, cada uma seguida por uma camada de normalização e *max pooling* e por uma camada totalmente conectada. Cada sub-rede tem como saída um vetor de tamanho 500. Para o cálculo da similaridade das imagens foi utilizada a função cosseno. Os testes foram realizados para os *datasets* CUHK03, VIPER e i-LIDS. O método obteve resultados superiores aos métodos desenvolvidos por Gray, Brennan e Tao (2007) e Zhao, Ouyang e Wang (2013). Entretanto as sub-redes implementadas não compartilham os mesmos parâmetros, tornando a comparação entre os vetores de características imprecisos.

McLaughlin, Rincon e Miller (2016) projetaram uma rede neural siamesa composta por uma rede neural convolucional e uma rede neural recorrente para re-identificar pessoas com base em vídeo, tal como mostra a Figura 16. A rede neural convolucional produz um mapa de características de cada *frame* e envia como entrada para a rede neural recorrente. A principal particularidade de uma rede recorrente é que ela é capaz de armazenar informações temporais em uma memória durante o seu processamento. Com base nisso, as informações de cada *frame* são mantidas e processadas pela rede neural recorrente a medida que novos *mapas* de características são obtidos pela rede. Uma camada de *temporal pooling* é utilizada para combinar todas essas características. A função de similaridade utilizada neste método é a Distância Euclidiana. Os *datasets* utilizados para treinamento foram iLIDS-VID (WANG *et al.*, 2014) e PRID2011

(HIRZER *et al.*, 2011).

Figura 16 – Arquitetura proposta por McLaughlin, Rincon e Miller (2016), que consiste em uma rede neural siamesa composta por CNN e RNN. Sendo que cada *frame* é processado por uma CNN que produz um vetor de característica. Em seguida esse vetor passa por uma RNN, que possibilita armazenar informações de tempo. Isso permite que a rede resuma informações de uma sequência de vídeo.



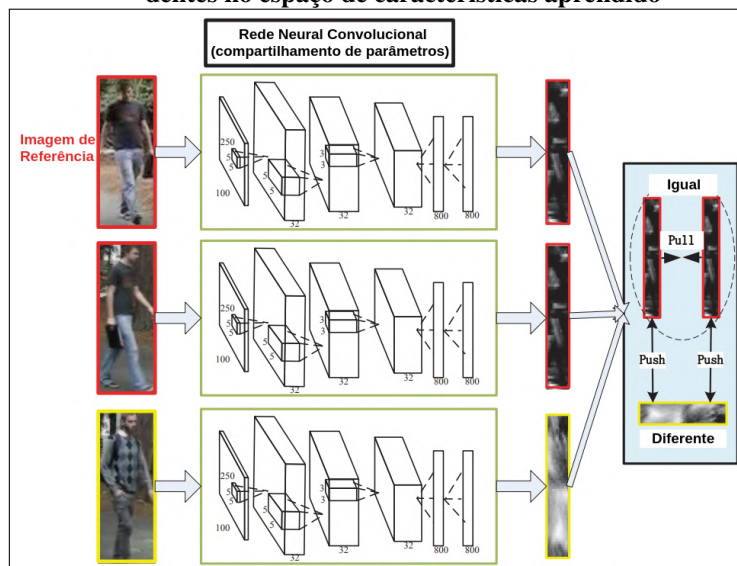
Fonte: (MCLAUGHLIN; RINCON; MILLER, 2016)

Um outro modelo de rede siamesa foi utilizado por Cheng *et al.* (2016) para a re-identificação de pessoas. Neste caso, ao invés de receber duas imagens de entrada, a rede siamesa recebe três. Assim, a rede é composta por 3 sub-redes idênticas. A estrutura dessa rede pode ser observada na Figura 17. As entradas da rede consistem em uma imagem de referência ou âncora, uma positiva da mesma pessoa e outra negativa de uma pessoa diferente.

Cada sub-rede proposta por Cheng *et al.* (2016) é formada por uma rede neural convolucional que gera mapas de características globais e de partes do corpo de uma pessoa, como pode ser observado na Figura 18. Com isso, a sub-rede contém uma camada de convolução global, uma camada de convolução de corpo inteiro e quatro camadas de convolução de partes do corpo. As camadas representadas por $P_i - conv_1$ e $P_i - conv_2$ representam as camadas referentes as partes do corpo, enquanto as camadas representadas por B são indicam a imagem de corpo inteiro. Essas cinco camadas são treinadas separadamente umas das outras e depois as suas saídas são concatenadas em um único vetor $N - fc$. Vale ressaltar que os vetores concatenados para formar o vetor $N - fc$ são 4 vetores de partes do corpo $P_i - fc$ e 1 vetor da imagem de corpo inteiro $B - fc$. Durante o treinamento, a rede calcula a perda entre os vetores de cada sub-rede, por meio da função *Triplet Loss Function*. Com isso, os pares positivos podem ser representados mais próximos, enquanto os pares negativos são mais distantes.

Wang *et al.* (2018) propôs um método para re-identificação de pessoas utilizando aprendizado não supervisionado. Nesta técnica, o compartilhamento do conhecimento de informações de um determinado indivíduo de origem é feito por meio de atributos aprendidos com dados de

Figura 17 – Arquitetura proposta por Cheng *et al.* (2016). A rede recebe como entrada três imagens de entrada: âncora, positiva e negativa. A função de perda *Triplet Loss* é usada para treinar os modelos de rede, diminuindo a distância entre os pares correspondentes menos e aumentando a distância entre os pares não correspondentes no espaço de características aprendido

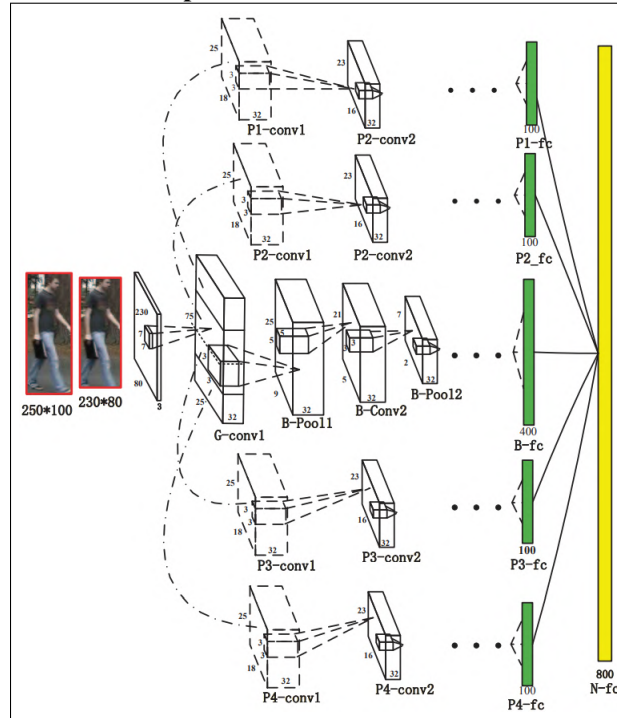


Fonte: Adaptado de (CHENG *et al.*, 2016)

origem rotulados, dois desses dados são transferidos para dados não rotulados, através de aprendizado conjunto da transferência de identidade entre domínios. Para isso, cada figura de um determinado indivíduo passa por dois extratores de características, o primeiro deles com o objetivo de extrair informações sensíveis para re-identificação e o segundo com o objetivo de extrair o conhecimento semântico dos rótulos dos atributos. Depois disso, um canal de fusão de aprendizado é usado para integrar as informações obtidas pelos dois canais. Neste trabalho, Wang *et al.* (2018) utilizou um *autoencoder* no canal de fusão, com a justificativa de que ele possui uma grande capacidade de capturar as informações mais importantes das entradas e também porque é uma representação mais concisa dos recursos, facilitando a transferência de informações no canal de fusão.

O Quadro 1 apresenta um breve resumo de cada trabalho descrito neste capítulo. O resumo expõe o método empregado para extrair características das imagens, tanto utilizando *Deep Learning*, tanto para os que utilizaram algoritmos para extração de características. Também é indicado a forma que foi verificada a correspondência entre duas imagens nos trabalhos descritos. Além disso, o Quadro 1 também mostra os *datasets* utilizados para validação dos métodos propostos. Em relação as medidas de avaliação utilizadas pelos autores para análise da técnica desenvolvida, todos os trabalhos descritos utilizaram a Curva CMC.

Figura 18 – Sub-rede proposta por Cheng *et al.* (2016). O modelo de sub-rede proposto consiste nas seguintes camadas: uma camada de convolução global, uma camada de convolução de corpo inteiro, quatro camadas de convolução de parte do corpo e uma camada de corpo inteiro



Fonte: (CHENG *et al.*, 2016)

3.1 DATASETS

Há diversos *datasets* públicos para a re-identificação de pessoas em imagens digitais disponíveis na literatura. Esses *datasets* variam em número de câmeras, condições de iluminação e perspectiva, bem como total de identidades, que corresponde a cada indivíduo que aparece nas imagens. Além disso, as imagens podem ser classificadas em dois tipos: *single shot* e *multishot shot*. Em que o primeiro refere-se a imagens obtidas de forma não sequenciais por câmeras diferentes. Já no caso das imagens *multi shot*, as imagens são obtidas seguindo a trajetória de uma pessoa, capturada por meio de uma sequencia de imagens.

A Tabela 1 apresenta uma seleção dos *datasets* públicos utilizados nos trabalhos relacionados na área de re-identificação de imagens descritos neste capítulo. São destacados os pontos mais relevantes de cada um, como: número de imagens, identidades, e câmeras, bem como se as imagens foram obtidas de forma *multi shot* ou *single shot*.

Dentre os trabalhos relacionados citados, os *datasets* mais utilizados foram VIPeR (GRAY; BRENNAN; TAO, 2007), i-LIDSVID(WANG *et al.*, 2014), CUHK01 (LI; ZHAO; WANG, 2012) e CUHK03 (LI *et al.*, 2014), utilizados em 7, 4, 3 e 3 trabalhos, respectivamente.

Quadro 1 – Trabalhos Relacionados

Referência	Método	Dataset
(GRAY; TAO, 2008)	Extração de características por histogramas de cores e texturas + Algoritmo <i>Ada-Boost</i>	VIPeR, i-LIDSVID
(PROSSER <i>et al.</i> , 2010)	Extração de características por histogramas de cores e texturas + <i>RankSVM</i>	VIPeR, i-LIDSVID
(ZHAO; OUYANG; WANG, 2013)	Extração de características salientes, por histograma LAB e descritor SIFT + KNN	VIPeR, ETHZ
(WU <i>et al.</i> , 2016)	CNN + extração de características por histogramas de cores e texturas de forma manual	ViPER, CUHK01
(LI <i>et al.</i> , 2014)	<i>Deep Filter Pairing Neural Network</i>	CUHK03, CUHK01, CUHK02
(AHMED; JONES; MARKS, 2015)	Rede Neural Siamesa formada por duas sub-redes compostas por duas camadas convolucionais conectadas por uma camada de diferença de vizinhança	CUHK03, CUHK01, VIPER
(YI <i>et al.</i> , 2014)	Rede neural siamesa formada por duas sub-redes neurais convolucionais unidas pela função cosseno	CUHK03, VIPER
(MCLAUGHLIN; RINCON; MILLER, 2016)	Rede Neural Siamesa formada por duas sub-redes compostas por uma rede neural convolucional e uma rede neural recorrente	i-LIDSVID, PRID2011
(CHENG <i>et al.</i> , 2016)	Rede Neural Siamesa formada por três sub-redes neurais convolucionais	i-LIDSVID, PRID2011, VIPeR, CUHK01
(WANG <i>et al.</i> , 2018)	Método não supervisionado que compartilha o conhecimento do domínio de origem por meio de características aprendidos com dados de origem rotulados, sendo transferidos para dados de destino não rotulados. Utiliza uma rede AE	VIPeR, PRID, Market-1501, DukeMTMC-ReID

Fonte: Autoria Própria

Tabela 1 – Datasets públicos para re-identificação de pessoas em imagens digitais

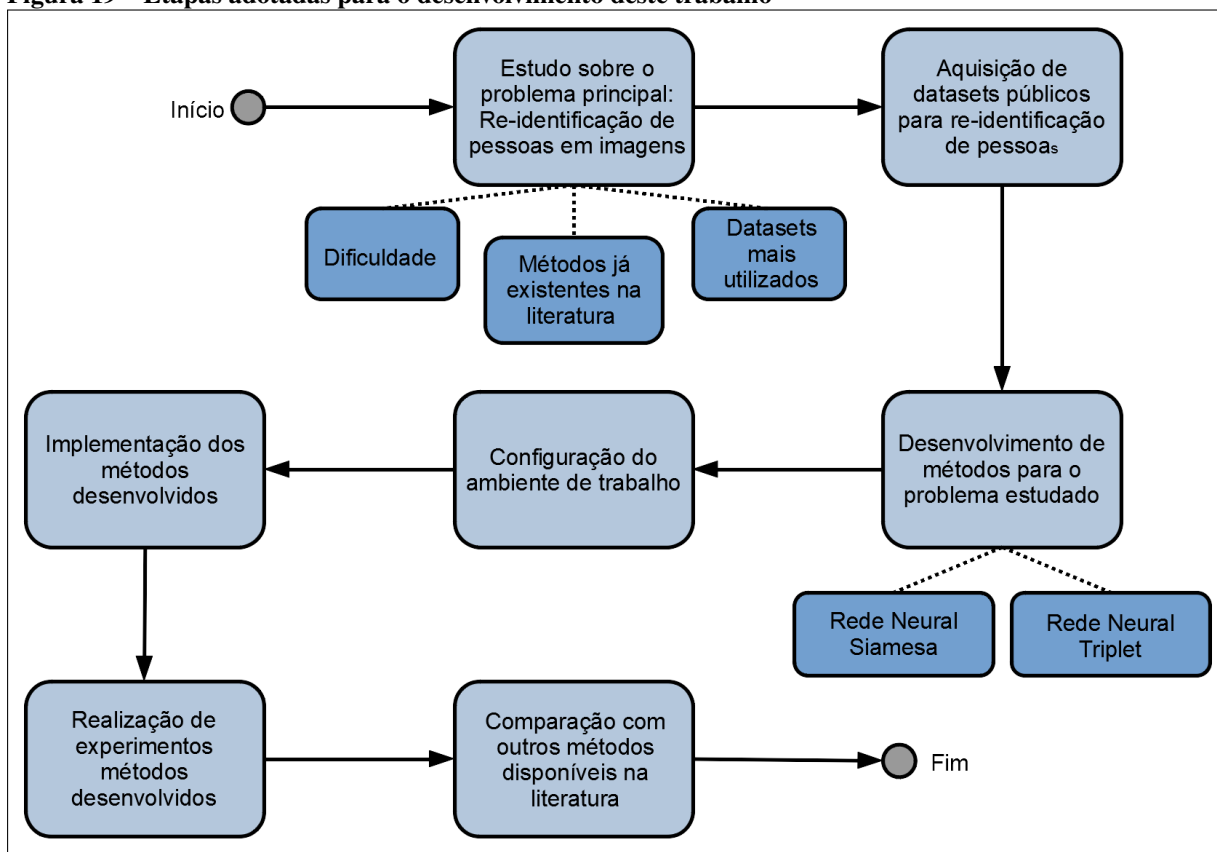
Referência	Imagens	Identities	Câmeras	Tipo
VIPeR (GRAY; BRENNAN; TAO, 2007)	1264	632	2	<i>single shot</i>
CUHK03 (LI <i>et al.</i> , 2014)	13164	1360	6	<i>single shot</i>
CUHK01 (LI; ZHAO; WANG, 2012)	3884	971	2	<i>single shot</i>
i-LIDSVID (WANG <i>et al.</i> , 2014)	42495	300	2	<i>multi shot</i>
PRID2011 (HIRZER <i>et al.</i> , 2011)	24541	934	2	<i>multi shot</i>
Market-1501 (ZHENG <i>et al.</i> , 2015)	32217	1501	6	<i>multi shot</i>
CAVIAR4REID (CHENG <i>et al.</i> , 2011)	1220	72	2	<i>multi shot</i>
3DPeS (BALTIERI; VEZZANI; CUC-CHIARA, 2011)	1011	192	8	<i>multi shot</i>
ETHZ (SCHWARTZ; DAVIS, 2009)	8555	83	2	<i>multi shot</i>
DukeMTMC-ReID Zheng, Zheng e Yang (2017)	36441	1812	8	<i>multi shot</i>

Fonte: Autoria Própria

4 MATERIAL E MÉTODOS

Este capítulo apresenta a metodologia adotada neste trabalho para re-identificação de pessoas em imagens digitais utilizando técnicas de *Deep Learning*, sendo dividida em etapas, conforme descrito na Figura 19.

Figura 19 – Etapas adotadas para o desenvolvimento deste trabalho



Fonte: Autoria própria

Na primeira etapa foi realizado um estudo a cerca de re-identificação de pessoas, levando em consideração quais são as principais dificuldades, técnicas desenvolvidas por outros autores e os *datasets* mais utilizados, no Capítulo é descrito sobre alguns trabalhos mais relevantes na literatura, sendo que no Quadro 1 do Capítulo 4 é apresentado um breve resumo a respeito de cada um. Com isso, foi observado que, apesar de existirem métodos que o tratem, as diferentes condições de imagens fazem com que o problema seja um desafio e permaneça sem uma solução definitiva.

A segunda etapa consiste na aquisição de *datasets* públicos para o problema de re-identificação de pessoas, que serão utilizados posteriormente no treinamento da rede e realização de testes. Na Seção 5.1 do Capítulo 5 são descritos os *datasets* públicos selecionados para realização dos experimentos.

As etapas seguintes são descritas nas próximas seções. Na Seção 4.1 são apresentadas

as duas redes neurais para re-identificação de pessoas propostas neste trabalho. Os experimentos realizados e medidas de avaliação utilizadas para verificação dos métodos são descritos no Capítulo 5.

4.1 REDE NEURAIAS PROPOSTAS

Seguindo a Figura 19, a próxima etapa consiste no desenvolvimento de métodos para re-identificar pessoas em imagens. Este trabalho propõe a implementação de dois modelos de rede neural, para a re-identificação de pessoas em imagens, obtidas de pontos de vista e câmeras diferentes. A primeira é uma Rede Neural Siamesa, composta por duas sub-redes idênticas. A segunda proposta é uma Rede Neural *Triplet*, formada por três sub-redes idênticas. As sub-redes de cada modelo são as mesmas. Cada sub-rede recebe uma imagem de entrada, que passa por um processo de extração de características e posteriormente a verificação de similaridade entre elas, com base nas características encontradas.

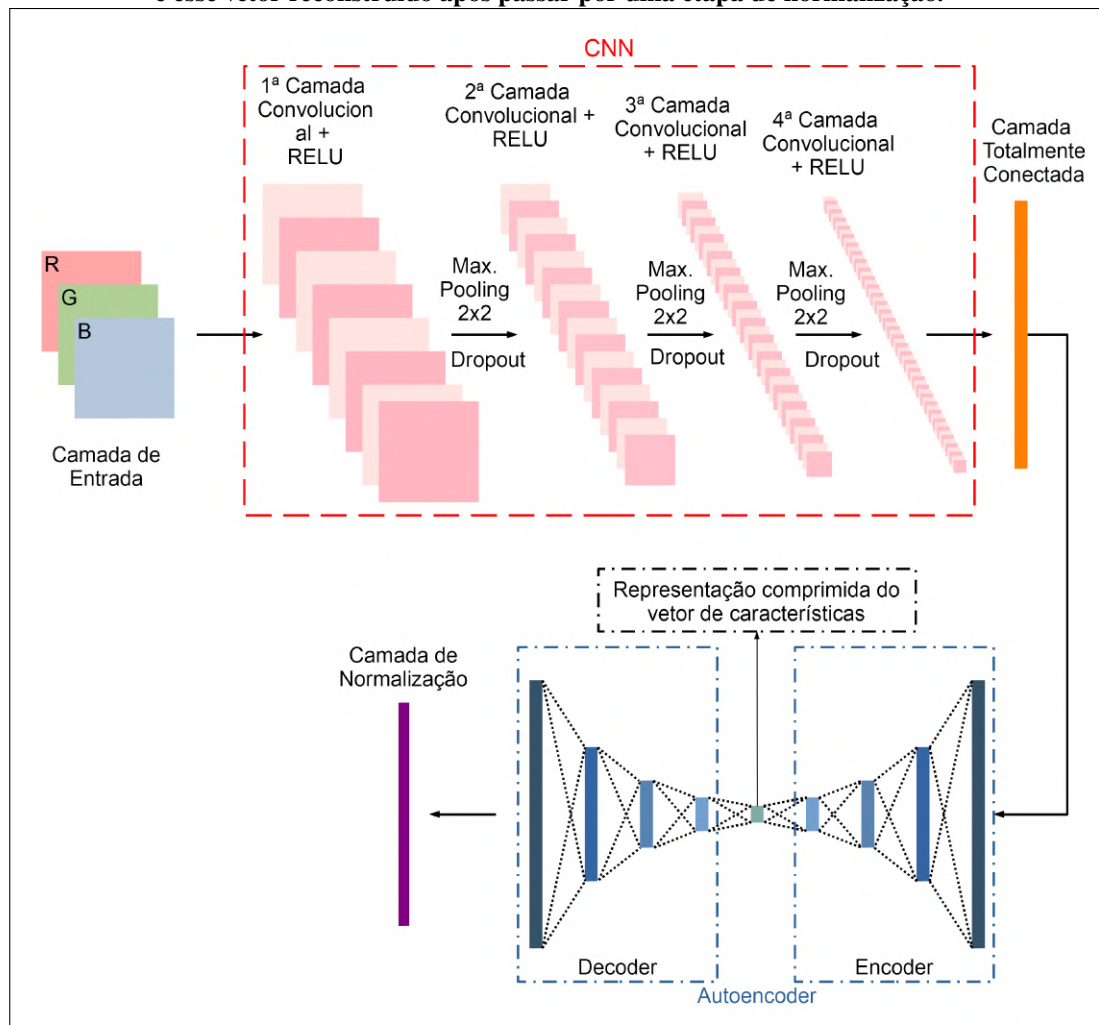
O modelo de sub-rede proposta é apresentado na Figura 20. Como pode ser observado, inicialmente cada imagem passa separadamente por uma rede neural convolucional com parâmetros iguais, isso é realizado para que a rede aprenda características gerais de cada imagem, resultando em um vetor de características. As camadas de convolução são intercaladas por camadas de *max pooling*, que reduz o tamanho dos mapas de características e também por camadas de *Dropout* para evitar *overfitting*.

Levando em consideração que cada imagem possui diversos tipos de variações, como iluminação e perspectiva, uma vez que não foram obtidas por uma mesma câmera e pela movimentação do pedestre. Com isso, com base no método de Wang *et al.* (2018), cada vetor de características produzido pela CNN passa por uma rede neural AE, que tem como objetivo reconstituir esses dados de forma que possa eliminar esses ruídos, tornando a rede menos propensa a erros. Após isso, o vetor de características produzido passa por uma camada de normalização, resultando na saída de cada sub-rede.

Como descrito no Capítulo 4, no método de Wang e Yeung (2013) a re-identificação de pessoas é feita de uma forma diferente, uma vez que cada imagem passa por duas redes neurais convolucionais, uma com dados rotulados e outra não. Os dois vetores produzidos são unidos para passar pelo AE, gerando um vetor de característica da imagem e depois uma medida de distância é aplicada para comparar com outras imagens. Wang *et al.* (2018) sugeriu que a utilização de uma rede neural AE após a extração de características de uma imagem serviu para gerar um vetor de características mais conciso, sem perder as informações importantes do vetor de características, uma vez que o AE aprendeu as informações mais relevantes de uma determinada representação de entrada.

Por fim, para cada imagem de entrada um vetor de características reconstruído e norma-

Figura 20 – Modelo de sub-rede proposto: a sub-rede recebe uma imagem RGB como entrada e é formada por uma CNN com 4 camadas de convolução, que produz um vetor de características. Esse vetor passa por um AE para reconstrução. A saída da sub-rede é esse vetor reconstruído após passar por uma etapa de normalização.



Fonte: Autoria própria

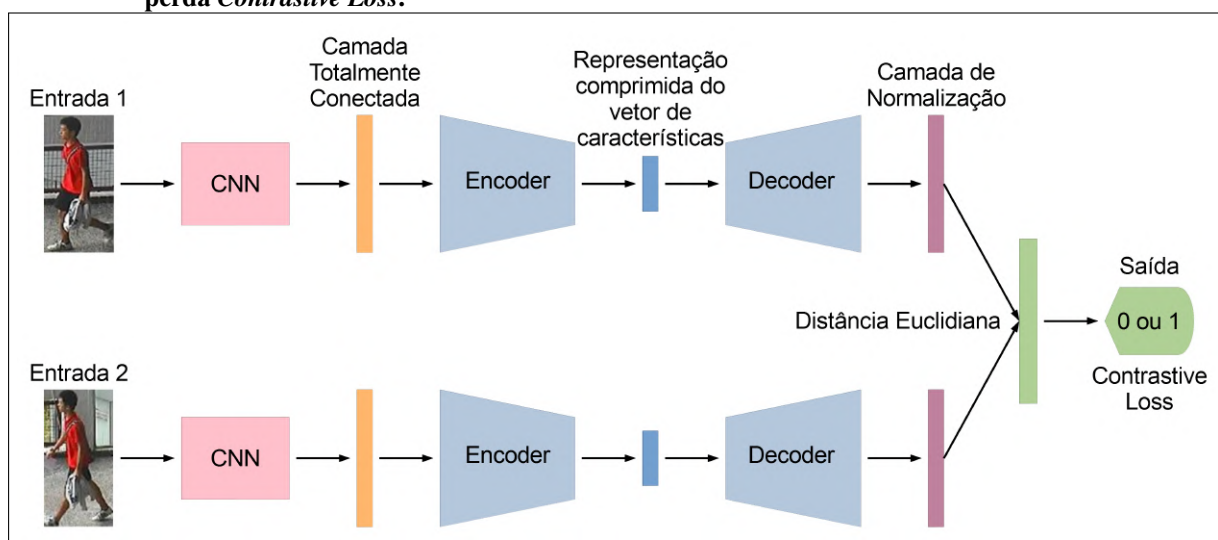
lizado será produzido, que devem ser mais semelhantes aos de uma imagem da mesma pessoa. Para realizar associação entre eles, ou seja, verificar se esses vetores corresponde a imagem de uma mesma pessoa. Para isso, neste trabalho é proposto dois tipos de abordagem: Rede Neural Siamesa e Rede Neural *Triplet*.

4.1.1 Rede Neural Siamesa

A Figura 21 mostra o modelo de Rede Neural Siamesa proposto. Este tipo de rede possui duas sub-redes idênticas que são unidas em suas saídas. Considerando que cada sub-rede produz um vetor de características para sua entrada, esses vetores comparados para estimar uma semelhança entre eles, resultando na saída da rede.

A utilização de Redes Neurais Siamesas é muito comum para estimar similaridade entre duas entradas. Com isso, considerando que uma solução para o problema de re-identificar de pessoas pode ser a estimação da similaridade entre duas imagens de uma mesma pessoa ou de pessoas diferentes, esse modelo de rede também já foi utilizado em diversos trabalhos que tratam deste problema como no caso dos trabalhos de Li *et al.* (2014), Ahmed, Jones e Marks (2015), Yi *et al.* (2014) e (MCLAUGHLIN; RINCON; MILLER, 2016). Em todos esses trabalhos, foram obtidos resultados superiores em relação aos métodos utilizados anteriormente, como o método de Gray e Tao (2008), por exemplo. Vale ressaltar que, apesar de utilizar a mesma arquitetura de rede, os trabalhos mencionados deferem no modelo de sub-rede e também na função de perda.

Figura 21 – Modelo de Rede Neural Siamesa proposto: a rede neural proposta é uma rede neural siamesa, formada por duas sub-redes idênticas que recebem uma imagem diferente para verificação da similaridade entre elas. Para isso, cada imagem passa por uma CNN e por um AE, que formam a sub-rede, resultando em um vetor de características para cada imagem. Para a verificação da similaridade a Distância Euclidiana entre os vetores é calculada, passando por uma função de perda *Contrastive Loss*.

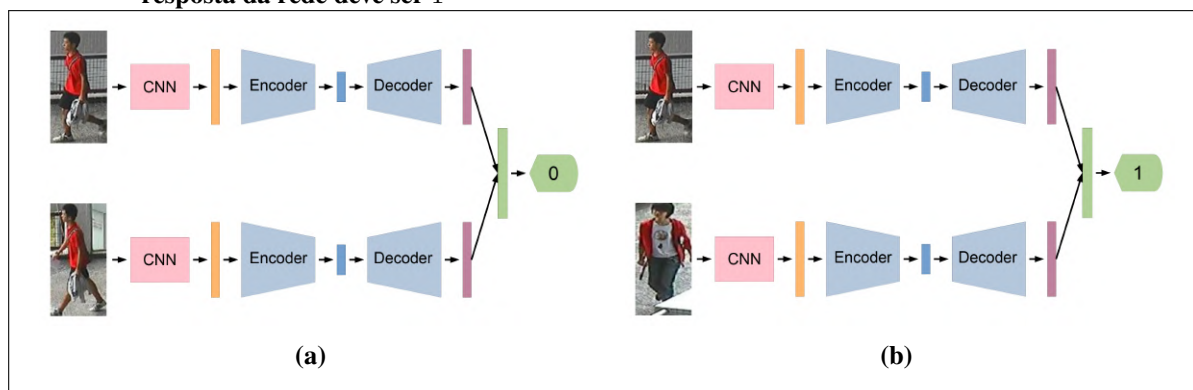


Fonte: Autoria própria

A Rede Neural Siamesa durante o treinamento ajusta seus pesos, para atingir um determinado valor, de acordo com os pares de imagens de entrada, usando a função de perda *Contrastive Loss*, que tem como objetivo separar com maior distância as entradas pertencentes a pessoas diferentes usando um valor de margem predefinido e diminuir a distância entre imagens de entrada de pessoas com as mesmas identidades.

Para executar o treinamento da Rede Neural Siamesa, foi considerado $\frac{2}{3}$ do *dataset* utilizado, restando $\frac{1}{3}$ para os testes realizados. Vale ressaltar que essa divisão foi efetuada com base no número de indivíduos que o *dataset* contém. Ou seja, se houver 300 indivíduos diferentes, as imagens de 200 serão destinadas ao treinamento, enquanto o restante será para teste. Isso é necessário para que não ocorram pares de imagens de uma mesma pessoa nas duas etapas, contendo pessoas nunca vistas pela rede na fase de teste, não comprometendo os resultados obtidos.

Figura 22 – Associação entre pares positivos e pares negativos. A imagem (a) apresenta pares positivos, em que a rede deve retornar 0 como resposta. Já a imagem (b) mostra pares negativos, nesse caso a resposta da rede deve ser 1



Fonte: Autoria própria.

Além das duas imagens de entrada, a rede também recebe para o treinamento uma variável, que pode ser 0, se as imagens correspondem a uma mesma pessoa ou 1 se forem de pessoas diferentes. Dessa forma, a entrada da rede contém duas imagens e um valor binário. Com isso, a rede irá ajustar seus pesos durante o treinamento para atingir um determinado valor, de acordo com os pares de imagens da entrada, utilizando uma função de perda para otimização. A função de perda utilizada é a Contrastive Loss, que busca separar com uma maior distância entradas pertencentes a pessoas diferentes utilizando um valor de margem pré-definido. Da mesma forma, essa função procura reduzir a distância entre entradas de uma mesma pessoa. A distância utilizada neste trabalho é a Euclidiana.

A Figura 22 apresenta dois exemplos de entradas e saída da rede, em que em (a) temos um par de imagens positivo, ou seja, quando as duas imagens são de uma mesma pessoa, neste caso a rede deve retornar 0 na sua saída. Além disso, no exemplo (b) há um par de imagens negativos, visto que as duas imagens são de pessoas diferentes, assim a saída da rede deve ser igual a 1.

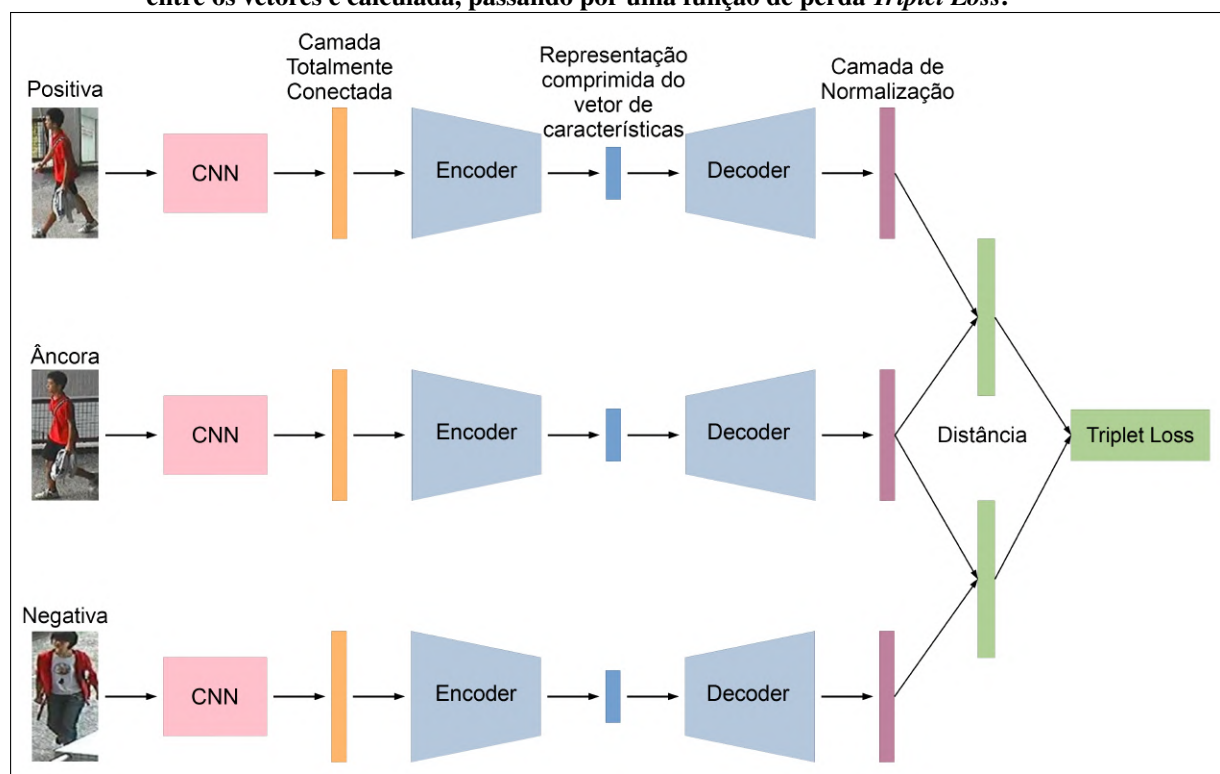
4.1.2 Triplet Loss

O outro modelo proposto é uma Rede Neural *Triplet*, apresentado na Figura 23. Esse tipo de rede foi utilizado no método proposto por Cheng *et al.* (2016) e gerou melhores resultados do que outros métodos que utilizaram Redes Neurais Siamesas, por exemplo. No entanto, a rede de Cheng *et al.* (2016) se difere do modelo proposto neste trabalho por utilizar apenas redes neurais convolucionais nas suas sub-redes e também por extrair características de 4 regiões diferentes da imagem de entrada.

Uma Rede Neural Triplet é composta por três sub-redes idênticas, em que cada uma é responsável por extrair características de uma imagem de entrada. Vale ressaltar que a entrada

recebe três tipos de imagens: âncora, positiva negativa. A imagem chamada de âncora representa uma referência para as outras imagens, uma vez que a imagem positiva deve ser do mesmo indivíduo presente na âncora e a imagem negativa de um indivíduo com uma identidade diferente. Na Figura 23 é possível observar que, apesar da pessoas presentes nas três imagens de entrada estarem com a mesma cor de roupa, apenas a imagem âncora e imagem positiva são da mesma pessoa.

Figura 23 – Modelo de Rede Neural *Triplet* proposto: a rede neural *triplet* é formada por três sub-redes idênticas que recebem uma imagem diferente para verificação da similaridade entre elas, sendo que a primeira recebe uma imagem âncora, a segunda uma imagem positiva e a terceira uma imagem negativa. Cada imagem passa por uma CNN e por um AE, resultando em um vetor de características para cada imagem. Para a verificação da similaridade a Distância Euclidiana entre os vetores é calculada, passando por uma função de perda *Triplet Loss*.



Fonte: Autoria própria

A partir das três imagens de entrada, serão produzidos três vetores de características pelas sub-redes. Com isso, a Rede Neural *Triplet* irá gerar dois valores de distância, sendo a distância entre a imagem âncora e a imagem positiva e distância entre a imagem âncora e a imagem negativa. Neste trabalho, assim como para a Rede Neural Siamesa, a distância utilizada foi a Euclidiana (Equação 2.5).

Em uma Rede Neural *Triplet*, o treinamento é realizado considerando que se trata de um problema de classificação, em que o objetivo é classificar qual das imagens (positiva ou negativa) é da mesma classe que uma imagem de referência, neste caso, chamada de âncora. Isso é realizado através do cálculo entre as distâncias, determinando que as imagens mais próximas possuem a mesma identidade. Para isso uma função SoftMax é utilizada nas saídas, de forma que sejam calculadas probabilidades de uma imagem pertencer a determinada classe ou não.

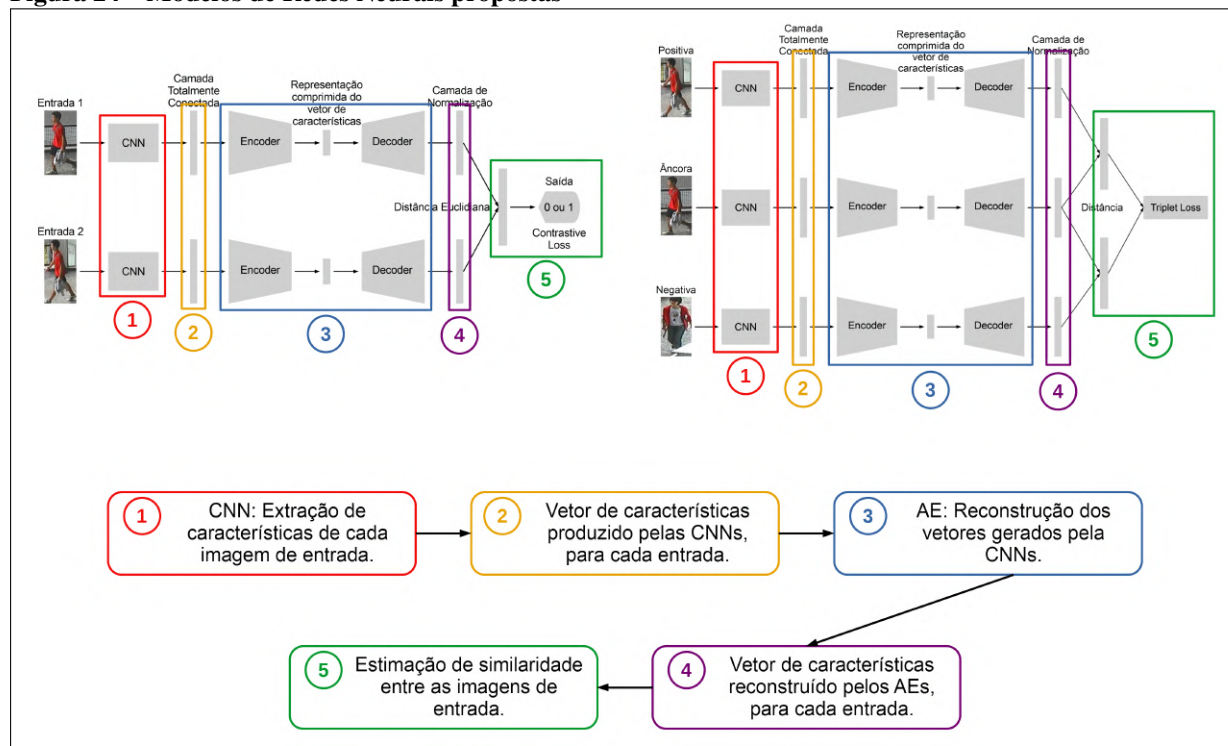
Durante o treinamento a rede irá atualizando seus pesos, utilizando a função de perda *Triplet Loss*.

Neste trabalho, foi destinado $\frac{2}{3}$ do *dataset* para treinamento e $\frac{1}{3}$ para os testes realizados, com base no número de indivíduos, da mesma forma que para os experimentos com a Rede Neural Siamesa. A criação dos triplos para a entrada foi realizada de forma aleatória. Para imagens de uma determinada câmera todas as imagens foram atribuídas como âncora em um triplo diferente. Em seguida, foram selecionadas todas as imagens com a mesma identidade obtidas por uma câmera diferente e atribuídas como entradas positivas para todas as âncoras com a mesma identidade, restando apenas entradas negativas para cada triplo, que foram selecionadas de forma aleatória, com a preocupação de não selecionar imagens com a mesma identidade da âncora.

4.1.3 Compilação dos modelos propostos

A Figura 24 apresenta uma descrição resumida de cada etapa das redes desenvolvidas neste trabalho. Como pode ser observado, as etapas são basicamente as mesmas entre cada rede, uma vez que as sub-redes possuem a mesma arquitetura em todos os casos. A diferença principal é o número de sub-redes, duas na Rede Neural Siamesa e três na Rede Neural *Triplet*, bem como, a função de perda utilizada para estimar a similaridade entre as imagens.

Figura 24 – Modelos de Redes Neurais propostas



Fonte: Autoria própria

A primeira etapa indicada pelo número 1 na Figura 24 consiste na CNN que será responsável pela extração de características de cada imagem de entrada, que é representado por 2. A próxima etapa representa o AE que reconstrói o vetor gerado pela CNN e deve manter as informações mais importantes para a re-identificação. A saída do AE passa por uma camada de normalização, na etapa 4 da Figura 24, que realiza uma conversão nos dados para uma mesma escala.

A última etapa consiste na estimação de similaridade entre as imagens de entrada. Na Rede Neural Siamesa isso ocorre através da função de perda *Contrastive Loss* e na Rede Neural *Triplet* pela função *Triplet Loss*.

5 EXPERIMENTOS E RESULTADOS

Neste capítulo serão apresentados os experimentos e resultados obtidos utilizando as técnicas para re-identificação de pessoas em imagens digitais proposta neste trabalho. Na Seção 5.1.2 são apresentadas as medidas de avaliação utilizadas para analisar os resultados obtidos. Na Seção 5.2 os resultados dos experimentos realizados são relatados. Na Seção apresentam os resultados obtidos com cada um dos *datasets* utilizados. A discussão dos resultados é feita nas Seções 5.3 e 5.4, sendo que a primeira aborda uma comparação entre as duas redes propostas e a segunda realiza uma comparação dos resultados com outros métodos disponíveis no estado da arte.

5.1 SETUP EXPERIMENTAL

Nesta seção serão apresentados os *datasets* selecionados para a realização dos experimentos e validação dos modelos de redes propostas. A escolha dos *datasets* foi realizada de acordo com os mais utilizados, sendo selecionados os *datasets* VIPeR (GRAY; BRENNAN; TAO, 2007) e CUHK03 (LI *et al.*, 2014). Além disso, o *dataset* i-LIDSVID (WANG *et al.*, 2014) também foi escolhido por ser do tipo *multi shot*. Também serão descritas as medidas de avaliação das redes: acurácia e curva CMC. Detalhes sobre a implementação das redes propostas e configurações do ambiente de trabalho e máquina utilizada nos experimentos são detalhados na Seção 5.1.3.

5.1.1 Datasets

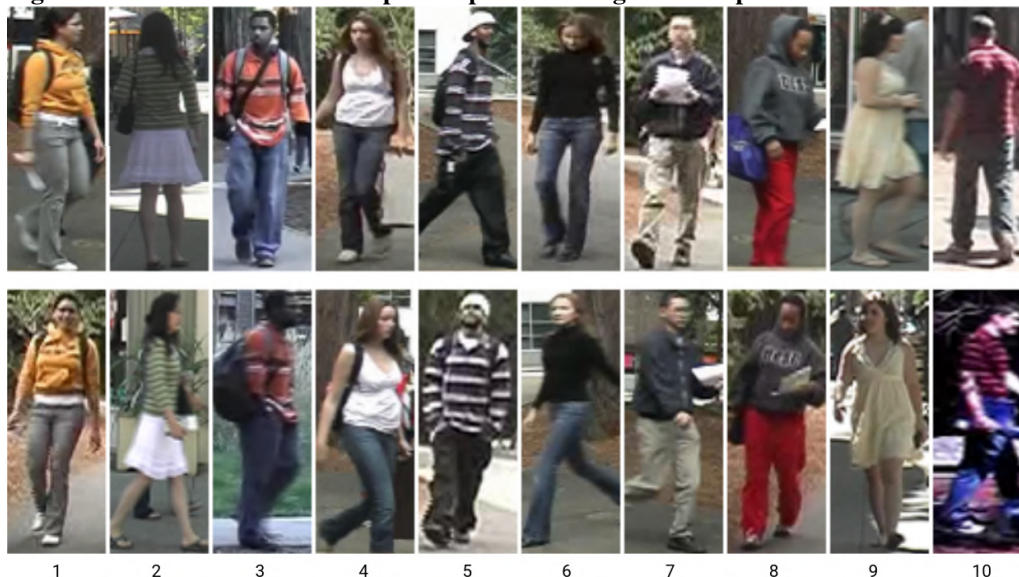
O *dataset Viewpoint Invariant Pedestrian Recognition*, conhecido também por VIPeR, foi desenvolvido por Gray, Brennan e Tao (2007), devido a necessidade de um conjunto de dados que pudesse ser aplicado a um método para reconhecimento de pedestres com variações no ponto de vista. Segundo Gray, Brennan e Tao (2007), para simular ambientes reais de segurança, nos quais pedestres aparecem em grandes ambientes abertos, podendo ser vistos de qualquer ângulo. Este *dataset* contém um total de 1264 imagens de 632 pedestres, em que para cada indivíduo há duas imagens capturadas por câmeras diferentes. As imagens foram obtidas de quatro ângulos diferentes, apresentados na Tabela 2, que contém para cada ângulo o número de indivíduos, consistindo em um único par de pontos de vista para indivíduos diferentes, capturadas por duas câmeras. A Figura 25 apresenta 10 exemplos deste *dataset*, podendo ser observada a variação entre pontos de vista e também mudanças na iluminação e poses dos indivíduos.

Tabela 2 – Distribuição dos ângulos do ponto de vista no *dataset* VIPeR

Ângulo	Número de Exemplos
45	70
90	363
135	96
180	103

Fonte: (GRAY; BRENNAN; TAO, 2007)

Figura 25 – *Dataset* VIPeR: exemplos de pares de imagens de 10 pedestres diferentes



Fonte: (GRAY; BRENNAN; TAO, 2007)

O conjunto de dados *iLIDS Video Re-Identification* (i-LIDSVID) de Wang *et al.* (2014) contém imagens de 300 pedestres diferentes obtidas através de duas câmeras não sobrepostas. Há dois conjuntos de imagens para cada indivíduo: um de imagens estáticas e outro de imagens sequenciais, ou seja, *single shot* e *multi shot*, respectivamente. No caso das sequenciais, as imagens foram obtidas por meio do rastreamento dos pedestres, contendo de 23 a 192 *frames* para cada pessoa diferente do *dataset*.

A utilização do conjunto de dados i-LIDSVID é desafiadora, uma vez que há muitas variações entre as imagens de diferentes câmeras, como iluminação, variações de ponto de vista e oclusões. Além disso, também existem casos de pedestres com roupas muito parecidas, tornando o processo para re-identificação mais complexo. A Figura 26 apresenta um exemplo de imagens sequenciais de um mesmo pedestre, no entanto capturadas por câmeras diferentes. Vale ressaltar, que nesse conjunto de dados, nem sempre haverá a paridade *frame-a-frame*, visto que uma pessoa pode não aparecer nas vistas de ambas as câmeras em um mesmo momento.

O *dataset* CUHK03 foi desenvolvido por Li *et al.* (2014) levando em consideração que para treinar redes de *Deep Learning* deve-se utilizar um grande número de imagens. Com isso, o conjunto de dados CUHK03 foi construído com 13164 imagens de 1360 pessoas diferentes,

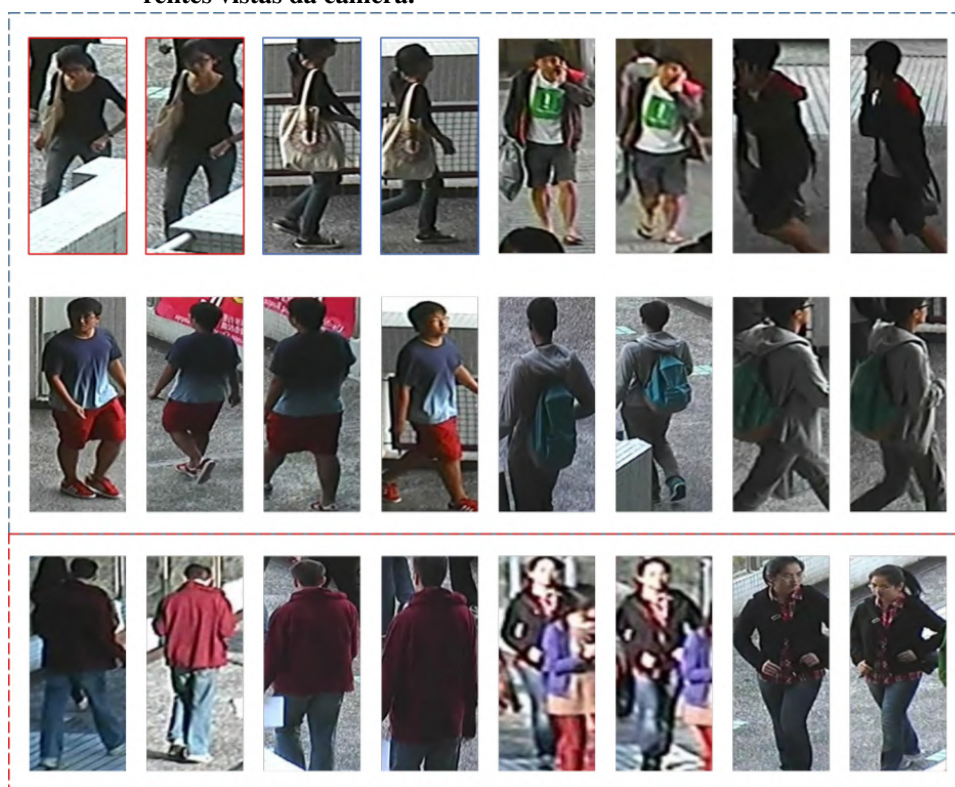
Figura 26 – Dataset i-LIDSVID: Imagens de um mesmo pedestre, obtidas por meio de duas câmeras não sobrepostas



Fonte: (WANG *et al.*, 2014)

obtidas através de seis câmeras de vigilância. Exemplos de imagens desse conjunto de dados podem ser vistos na Figura 27. Imagens de pessoas cortadas manualmente também estão disponíveis e também são detectadas usando um detector de pedestres, além dos frames originais.

Figura 27 – Dataset CUHK03: exemplos de imagens de pedestres observadas em diferentes vistas da câmera.



Fonte: (LI *et al.*, 2014)

Este *dataset* é bastante utilizado por apresentar vários problemas, como desalinhamento, oclusões, falta de partes do corpo, entre outros. Outro problema relevante para a reidentificação de pessoas presente neste *dataset* é o monitoramento pelas câmeras de vigilância em uma área aberta. Este fator torna as imagens sujeitas a variações na iluminação, causadas por vários fatores climáticos em uma única visualização da câmera. Outras transformações também podem ocorrer devido às várias direções que os pedestres se movimentam (LI *et al.*, 2014).

5.1.2 Medidas de Avaliação

Esta Seção aborda as medidas de avaliação escolhidas para analisar os resultados obtidos através dos experimentos realizados. Dentre elas, uma foi a Acurácia, selecionada para medir a eficácia das redes e por ser uma medida disponível na biblioteca utilizada para implementação. Outra medida é a Curva CMC, que é baseada no conceito de ranks. Essa medida foi selecionada por ser mais utilizada em trabalhos disponíveis na literatura, na área de re-identificação de pessoas.

5.1.2.1 Acurácia

A acurácia é uma das medidas de avaliação mais utilizadas em aprendizagem de máquina quando se trata de medir o desempenho de um classificador. Para compreender melhor como é realizado o cálculo nessa medida, deve-se considerar a matriz de confusão apresentada no Quadro 2, que aponta os exemplos reconhecidos correta e incorretamente para cada classe (SOKOLOVA; JAPKOWICZ; SZPAKOWICZ, 2006).

Quadro 2 – Matriz de Confusão

	Positivo	Negativo
Positivo	vp	fn
Negativo	fp	vn

Fonte: Adaptado de Sokolova, Japkowicz e Szpakowicz (2006)

Onde temos os seguintes termos:

- vp: verdadeiro positivo, quando uma determinada classe que está sendo buscada é prevista corretamente;
- vn: verdadeiro negativo, quando uma determinada classe que não está sendo buscada é prevista corretamente;
- fp: falso positivo, quando uma determinada classe que está sendo buscada é prevista incorretamente;
- fn: falso negativo, quando uma determinada classe que não está sendo buscada é prevista incorretamente.

Considerando a matriz de confusão, a acurácia A pode ser calculada como sendo a soma das classificações corretas dividido pelo total de classificações, conforme a Equação 5.1.

$$A = \frac{vp + vn}{vp + fp + vn + fn} \quad (5.1)$$

Em outras palavras a acurácia é a fração de previsões corretas de um determinado modelo em relação ao total de previsões, podendo ser calculada pela Equação 5.2.

$$A = \frac{\text{total de acertos}}{\text{total}} \quad (5.2)$$

Vale ressaltar que a acurácia não realiza uma distinção entre o número de rótulos corretos de diferentes classes. Com isso, se houver um desequilíbrio entre os dados, como, por exemplo, se tiver muitos dados de uma determinada classe, pode-se obter uma taxa alta de acurácia se na maioria das vezes forem previstos dados dessa determinada classe (SOKOLOVA; JAPKOWICZ; SZPAKOWICZ, 2006; JUBA; LE, 2019).

5.1.2.2 Curva CMC

Para a análise dos resultados obtidos no experimentos, foi utilizada uma medida conhecida como curva CMC, que é considerada uma medida de classificação com base no conceito de escores ou ranks. Neste caso, o desempenho de um modelo é analisado com base em cada amostra existente, determinando um conjunto de possíveis candidatos correspondentes para cada uma, sendo que esses candidatos devem pertencer a amostras com a mesma identidade que a amostra avaliada, no melhor caso (DECANN; ROSS, 2013).

De acordo com Bolle *et al.* (2005) para calcular uma curva CMC de um determinado conjunto de amostras B_i , devem ser considerados dois subconjuntos, sendo eles:

- Conjunto de Identificadores Biométricos: $G = \{B_1, B_2, \dots, B_m\}$, para m diferentes identidades;
- Conjunto de Exemplos: $Q = \{B'_1, B'_2, \dots, B'_n\}$ ou $Q = \{B'_l, l = 1, \dots, n\}$, com identidades pertencentes ao conjunto G , sendo que pode conter várias amostras com a mesma identidade e não necessariamente precisa conter todas as possíveis identidades do conjunto G .

Com isso, temos que m é o número total de identidades diferentes de um conjunto de amostras e n é o número de exemplos para cada identidade e o número total de amostras é dado por $N = n * m$. Para o cálculo dos ranks cada amostra pode ser comparada com as amostras restantes $N - 1$, dando origem a dois tipos de conjuntos de correspondências:

- Correspondência Genuína: as duas amostras comparadas pertencem a uma mesma identidade;
- Correspondência Impostora: as duas amostras comparadas pertencem a identidades diferentes.

Ao comparar as amostras do conjunto Q com o conjunto de identidades G , é originado um conjunto de pontuações de similaridade S'_i , que são a base para os ranks.

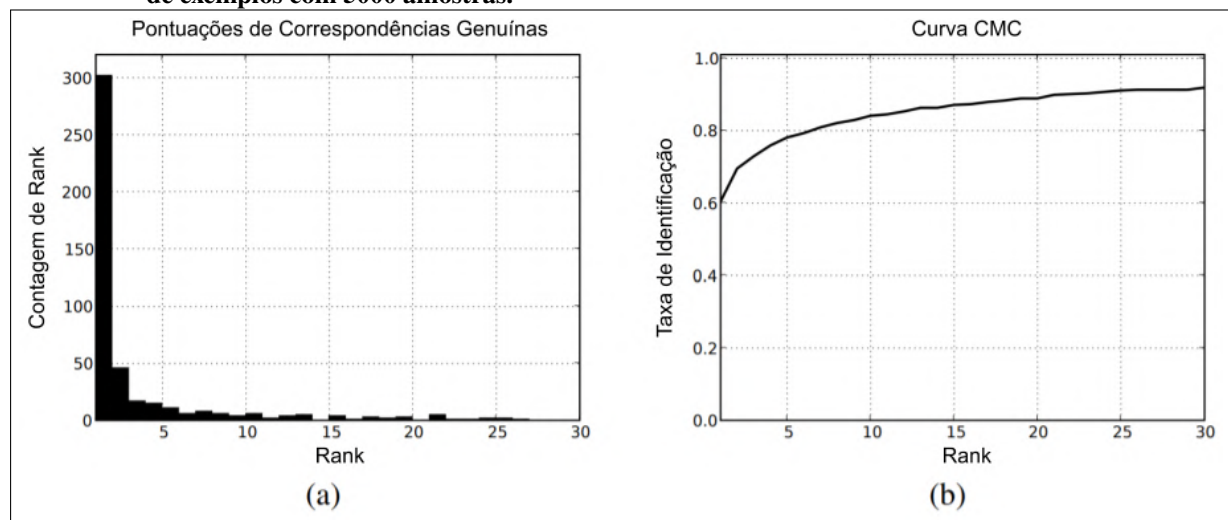
$$S'_i = \{s(B'_i, B_1), s(B'_i, B_2), \dots, s(B'_i, B_m)\} \quad (5.3)$$

Para cada exemplo B'_i , os ranks s são ordenados de acordo com as maiores pontuações de similaridades, da mais alta para a mais baixa, gerando uma lista C :

$$Cm(B'_i; G) = \{B(1), B(2), \dots, B(k), \dots, B(m)\}. \quad (5.4)$$

A curva CMC deve representar a probabilidade de uma correspondência genuína, ou seja, uma correspondência correta estar presente no $rank_k$ da lista C . Por exemplo, para a probabilidade de uma identificação correta no $rank_1$, as pontuações de similaridade s de correspondências genuínas devem ser maiores que todas as pontuações de correspondências impostoras $(m - 1)$ (BOLLE *et al.*, 2005; DECANN; ROSS, 2013).

Figura 28 – Curva CMC para um conjunto de identificadores biométricos com 500 identidades um conjunto de exemplos com 5000 amostras.



Fonte: Adaptado de Dunstone e Yager (2008)

A Figura 28, adaptada de Dunstone e Yager (2008), mostra um exemplo de cálculo da curva CMC. Na Figura 28(a) temos a distribuição de pontuações de correspondências genuínas para um conjunto de identificadores G com 500 identidades diferentes e um conjunto de exemplos Q com 5000 amostras. Observa-se no gráfico que para o $rank_1$ foram feitas 300 correspondências corretas. Já no $rank_2$ foram 50 correspondências corretas, considerando que os ranks são cumulativos, o total de correspondências corretas no $rank_2$ na verdade é 350, já que $rank_1 + rank_2 = 300 + 50$ e assim por diante. No gráfico apresentado pela Figura 28(b) temos a curva CMC, sendo que cada ponto plotado representa a taxa de identificação para cada rank, calculados por meio do número de pontuações genuínas em relação ao número total de identidades. Por exemplo, a taxa de acerto no $rank_2$ é de $350/500 = 70\%$ e no $rank_5$ é de $395/500 = 79\%$

Segundo Dunstone e Yager (2008), em um sistema ideal, a curva CMC começa alta e deve convergir para 100% rapidamente. Vale ressaltar que a curva CMC é cumulativa, ou seja, pode ser crescente ou permanecer constante, mas nunca irá decrescer. Além disso, para um conjunto G com tamanho m , ou seja, com m identidades diferentes, a curva para ranks de 1 a m irá convergir para 100%.

5.1.3 Ambiente de Trabalho e Implementação

A implementação das redes neurais desenvolvidas neste trabalho foi realizada utilizando o *framework Keras* (CHOLLET *et al.*, 2015), por meio de linguagem de programação *Python* e da biblioteca *tensorflow* (ABADI *et al.*, 2015), para aprendizagem de máquina e *deep learning*. Na Tabela 3 são apresentadas as versões utilizadas de cada ferramenta. A execução dos códigos foi realizada na máquina detalhada na Tabela 4, em ambiente *Linux*.

Tabela 3 – Ferramentas utilizadas nos experimentos

Ferramenta	Versão
Python	3.5.2
Keras	2.3.1
TensorFlow-gpu	2.0.0
Cuda	9.0.176
CuDNN	7.6.5
Sublime Text	3.2.1

Fonte: Autoria Própria

Tabela 4 – Máquina utilizada nos experimentos

Ferramenta	Versão
Processador	Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz
Memória RAM	16 GB
Arquitetura	64 bits
Placa de vídeo	NVIDIA Quadro P2000 (GP106GL) 5 GB

Fonte: Autoria Própria

Keras foi o *framework* escolhido por ter a capacidade de implementar e treinar aplicações que utilizam *deep learning*, utilizando CPU ou GPU. Além disso, essa ferramenta possui suporte para redes neurais convolucionais e redes neurais recorrentes, bem como a combinação das duas ou outras arquiteturas de redes arbitrárias, como modelos com múltiplas entradas e saídas, por exemplo. *Keras* é conhecido por possuir uma *interface* de programação simples para facilitar a programação dos modelos, uma vez que atua com operações de alto nível, não trabalhando diretamente com tensores. Para isso, deve ser utilizada uma biblioteca de tensores para servir como mecanismo de *back-end*, como por exemplo o *Tensorflow* (CHOLLET *et al.*, 2015; CHOLLET, 2018).

O *TensorFlow* é uma biblioteca de código aberto desenvolvido pelo Google para a construção e treinamento de modelos de aprendizagem de máquina e lida diretamente com operações de baixo nível, como manipulação e diferenciação de tensores. Essa biblioteca está disponível para CPU e GPU, sendo que para GPU é necessária a utilização da biblioteca *NVIDIA CUDA Deep Neural Network* (cuDNN) (ABADI *et al.*, 2015; CHOLLET, 2018).

5.2 RESULTADOS

Nesta Seção são apresentados os resultados obtidos através do treinamento e teste das redes implementadas. Os experimentos foram realizados utilizando três *datasets* diferentes: VIPeR (GRAY; BRENNAN; TAO, 2007), i-LIDSVID (WANG *et al.*, 2014) e CUHK03 (LI *et al.*, 2014).

5.2.1 Experimentos com o *Dataset* VIPeR

Os primeiros experimentos foram realizados utilizando o *dataset* público VIPeR (GRAY; BRENNAN; TAO, 2007). Para aumentar o número de imagens de treinamento, foi realizado a técnica de *data augmentation* nas imagens deste *dataset*. A Figura 29 apresenta o resultado da aplicação dessa técnica que realizou 11 transformações na imagem original indicada por *a*, sendo que todas as imagens do *dataset* passaram pelas mesmas transformações. Com isso, foi obtido um total de 15144 imagens. Para realização dos testes, foram separadas em $\frac{2}{3}$ das imagens para o treinamento e o restante para teste, com base no número de pessoas diferentes contidas na base de dados. Como há imagens de duas câmeras para 631 pessoas diferentes, imagens de 421 pessoas foi reservada para treinamento e 210 pessoas para teste. Assim, o total de imagens para treinamento foi 10104 e para teste 5040.

Para analisar os resultados produzidos com o treinamento da rede e quão bem ela está re-identificando pessoas em imagens diferentes, foi aplicada inicialmente uma medida de desempenho conhecida como acurácia ou precisão. A acurácia é dada pela proporção de exemplos para os quais a rede irá produzir a saída correta (GOODFELLOW; BENGIO; COURVILLE, 2016).

No primeiro experimento o modelo de sub-rede utilizado não continha a normalização em sua última camada. Neste teste foi realizado o treinamento variando o número de épocas na Rede Neural Siamesa. Os resultados atingidos podem ser observados da Tabela 5 e na Figura 30. Neste teste, a maior acurácia encontrada foi para um treinamento com 1200 épocas, chegando a um percentual de 87,93%. O treinamento com 600 épocas alcançou o pior resultado, com uma acurácia de 62,2%. Neste caso, a variação entre a maior e a menor acurácia foi igual à 25,73%.

Figura 29 – Resultado da aplicação da técnica de *Data Augmentation* no *dataset VIPeR*. A imagem *a* é a imagem original, enquanto o restante são resultados das transformações aplicadas na imagem original



Fonte: Autoria própria

Tabela 5 – Acurácia da Rede Neural Siamesa, em relação ao N° de Épocas de treinamento, para sub-redes com 4 camadas, utilizando o *dataset VIPeR*

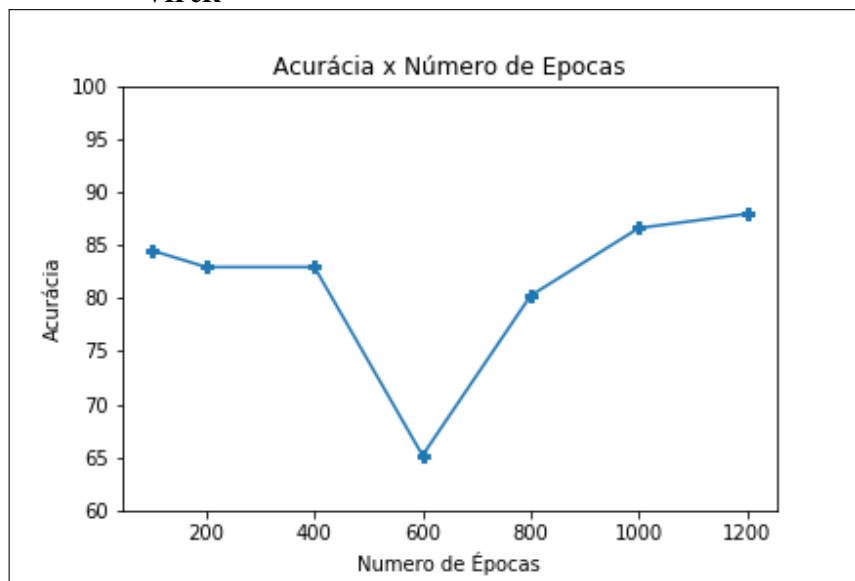
N° de Épocas	Acurácia
100	84,47%
200	82,92%
400	82,92%
600	65,2%
800	80,23%
1000	86,59%
1200	87,93%

Fonte: Autoria Própria

No entanto, a acurácia obtida com 600 épocas de treinamento não estava seguindo o resultado desejado, uma vez que aumentando o número de épocas se esperava que a acurácia aumentasse e o gráfico ficasse crescente, até um momento em que chegasse a um limite. Percebeu-se que os dados de saída precisavam ser normalizados uma vez que a normalização pode facilitar o aprendizado de uma rede, através da transformação dos dados para uma mesma escala (CHOLLET, 2018). Com isso, foi adicionada uma camada de normalização, por meio da função *Batch Normalization*, do *Keras*.

O experimento com a camada de normalização adicionada, gerou um resultado mais consistente não contento trechos com mudanças bruscas na acurácia, e com isso também foi realizado com a Rede Neural *Triplet*, como pode ser observado nos próximos experimentos

Figura 30 – Acurácia da Rede Neural Siamesa, em relação ao N° de Épocas de treinamento, para sub-redes com 4 camadas, utilizando o dataset VIPeR



Fonte: Autoria própria

realizados.

Com objetivo de realizar uma validação da utilização do AE nas sub-redes, foram realizados experimentos sem o AE, ou seja, após a camada totalmente conectada, o vetor produzido passa diretamente para a camada de normalização. Os resultados do treinamento realizado com as sub-redes sem e com o AE podem ser observados na Tabela 6 e Figura 31.

Como pode ser observado, os testes sem o AE obtiveram as piores taxas de acurácia, em relação aos testes com o AE. Em todos os casos, os modelos que utilizaram esse tipo de rede nas suas sub-redes obtiveram melhores resultados.

Além disso, ao comparar as duas redes com o AE, a Rede Neural *Triplet* obteve em maiores acurácias em todos os casos, em relação a Rede Neural Siamesa, obtendo a maior taxa, de 96,29% para 800 épocas de treinamento, sendo que a maior taxa da Siamesa foi de 89,54% também para 800 épocas.

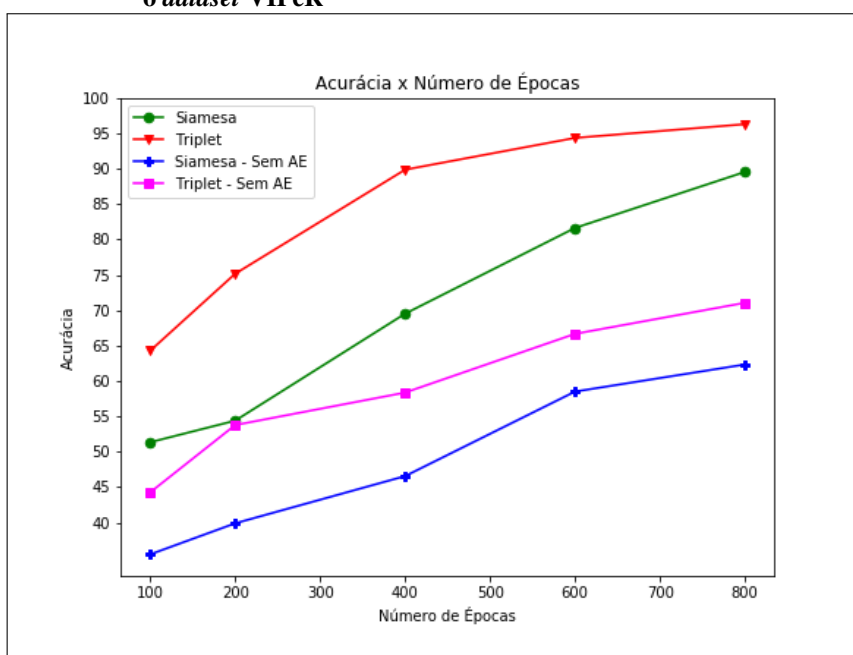
Tabela 6 – Acurácia em relação ao N° de Épocas de treinamento, utilizando o dataset VIPeR

N° de Épocas	Siamesa	<i>Triplet</i>	Siamesa-Sem AE	<i>Triplet-Sem AE</i>
100	51,33%	64,27%	35,48%	44,25%
200	54,38%	75,15%	39,87%	53,76%
400	69,5%	89,86%	46,54%	58,35%
600	81,62%	94,35%	58,49%	66,65%
800	89,54%	96,29%	62,32%	71,04%

Fonte: Autoria Própria

Além de utilizar a acurácia para análise das redes, também foi utilizada a Curva CMC, descrita na Seção 5.1.2.2, uma vez que é uma medida de avaliação bastante utilizada na litera-

Figura 31 – Acurácia, em relação ao N° de Épocas de treinamento, utilizando o dataset VIPeR



Fonte: Autoria própria

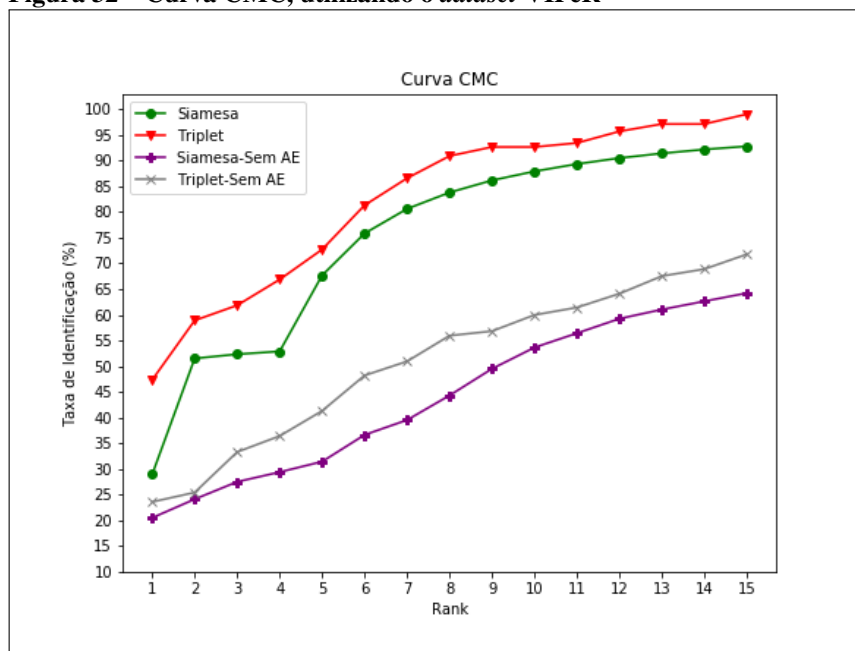
tura quando se pretende analisar métodos para re-identificação de pessoas. Neste trabalho, foi utilizado *ranks* de 1 à 15 para observar o comportamento da rede para re-identificar pessoas. As taxas de identificação de cada rede neural, utilizando o *dataset* VIPeR podem ser observadas na Figura 32 e Tabela 7. Vale ressaltar, que para este teste apenas foi considerado $\frac{1}{3}$ das identidades presentes no *dataset*, sendo que nenhuma identidade testada foi utilizada no treinamento da rede.

Tabela 7 – Curva CMC, utilizando o dataset VIPeR

Rank	Siamesa	Triplet	Siamesa-Sem AE	Triplet-Sem AE
1	28,98%	47,36%	20,52%	23,61%
2	51,52%	58,91%	24,14%	25,45%
3	52,34%	61,85	27,52%	33,32%
4	52,89%	66,87%	29,41%	36,43%
5	67,58%	72,72%	31,45%	41,32%
6	75,84%	81,32%	36,65%	48,21%
7	80,62%	86,58%	39,54%	50,95%
8	83,80%	90,90%	44,32%	55,95%
9	86,16%	92,65%	49,51%	56,83%
10	87,91%	92,65%	53,65%	59,98%
11	89,34%	93,43%	56,43%	61,43%
12	90,50%	95,69%	59,25%	64,12%
13	91,41%	97,12%	61,04%	67,54%
14	92,17%	97,12%	62,64%	68,90%
15	92,79%	99,04%	64,23%	71,78%

Fonte: Autoria Própria

Figura 32 – Curva CMC, utilizando o *dataset* VIPeR



Fonte: Autoria própria

Através dos resultados obtidos com cada uma das redes neurais implementadas, pode-se observar que assim como nos experimentos da acurácia, os resultados com as redes que utilizaram o AE na sub-rede obtiveram os melhores resultados em todos os casos. Dentre as duas melhores abordagens, a Rede Neural *Triplet* foi a que obteve um melhor desempenho nesses experimentos.

5.2.2 Experimentos com o *Dataset* i-LIDSVID

Além disso, também foi realizado um experimento para o *dataset* i-LIDSVID (WANG *et al.*, 2014). Um total de 21969 imagens foi utilizado de 319 pessoas diferentes obtidas por duas câmeras, sendo 14751 imagens para treinamento e 7218 imagens para teste. Neste caso, os experimentos relatados são com o modelo de sub-rede mostrada na Figura 20, encontrada no Capítulo 4 e também com o modelo de sub-rede sem o AE.

Tabela 8 – Acurácia em relação ao N° de Épocas de treinamento, utilizando o *dataset* i-LIDSVID

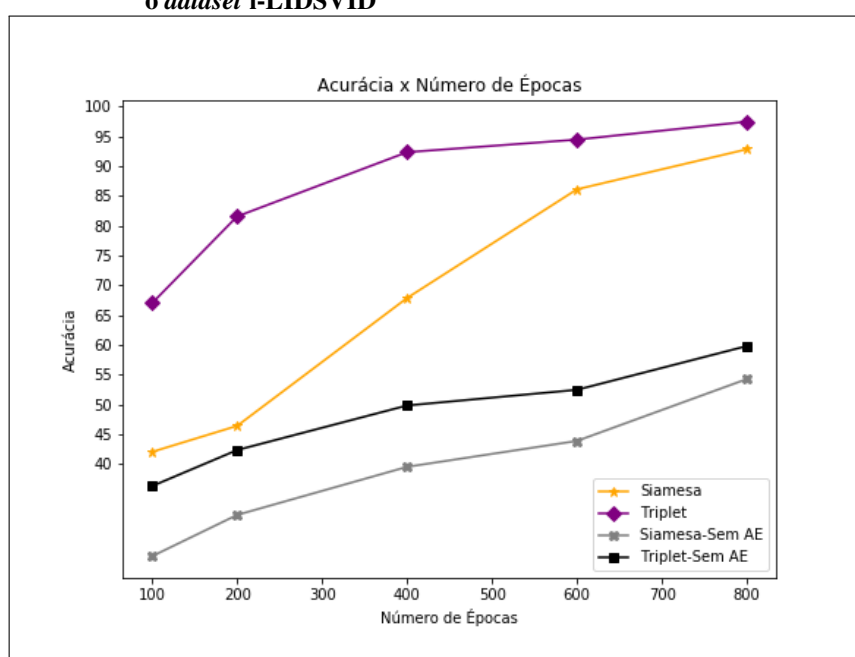
N° de Épocas	Siamesa	<i>Triplet</i>	Siamesa-Sem AE	<i>Triplet-Sem AE</i>
100	42,06%	67,04%	24,55%	36,32%
200	46,42%	81,58%	31,47%	42,36%
400	67,88%	92,36%	39,54%	49,84%
600	86,13%	94,47%	43,89	52,47%
800	92,85%	97,48%	54,28%	59,82%

Fonte: Autoria Própria

A Tabela 8 e Figura 33 apresentam as acurácias de cada rede neural implementada neste trabalho para o *dataset* i-LIDSVID. Pode ser observado que as redes que não possuem o AE após a camada do totalmente conectada nas sub-redes obtiveram taxas de acurácia inferiores em todos os casos, em relação as redes com o AE.

Comparando as duas redes que possuem o AE nas sub-redes, que é a proposta deste trabalho, ambas as redes obtiveram piores acurácias com 100 épocas de treinamento, sendo 42,06% para a Rede Neural Siamesa e 67,04% para a Rede Neural *Triplet*. Da mesma forma, as melhores acurácias foram com 800 épocas de treinamento de 92,85% e 97,48%, para a Rede Neural Siamesa e Rede Neural *Triplet*, respectivamente.

Figura 33 – Acurácia, em relação ao N° de Épocas de treinamento, utilizando o *dataset* i-LIDSVID



Fonte: Autoria própria

As Curvas CMC para este *dataset* podem ser observadas no gráfico da Figura 34 e Tabela 9. Neste caso, as abordagens propostas também foram comparadas suas versões sem o AE, sendo que as redes com o AE atingiram melhores resultados.

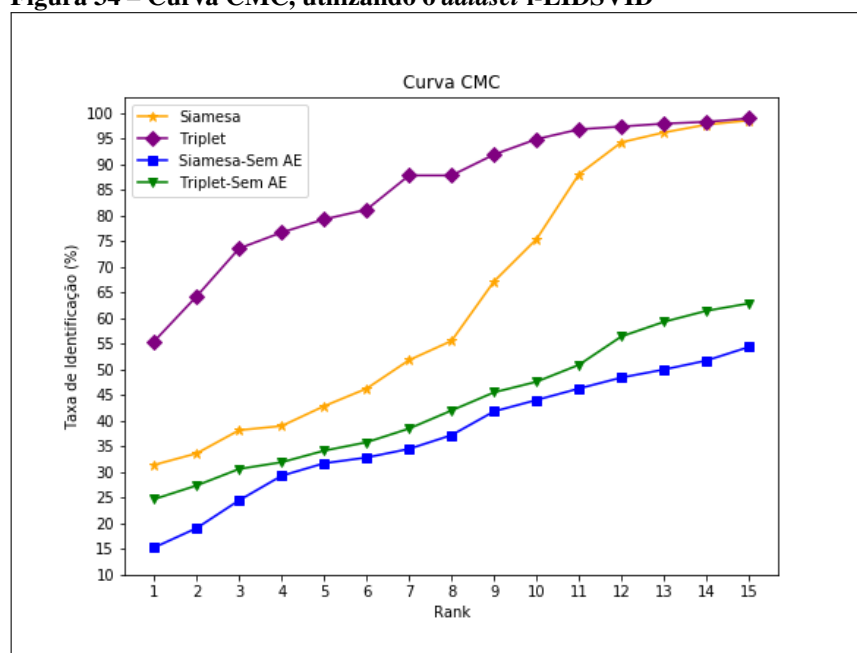
5.2.3 Experimentos com o *Dataset* CUHK03

Também foram realizados experimentos utilizando o *dataset* CUHK03 (LI *et al.*, 2014), que possui 1360 identidades diferentes e um total de 13164 imagens. Neste caso, para treinamento foram utilizadas imagens de 907 identidades diferentes e os imagens das 453 identidades restantes foram utilizadas para teste das redes. A Tabela 10 e Figura 27 mostram as acurácias obtidas para cada rede com o este *dataset*. Nestes experimentos as redes que foram implementadas sem o AE também obtiveram piores resultados, assim como nos experimentos anteriores.

Tabela 9 – Curva CMC, utilizando o *dataset* i-LIDSVID

Rank	Siamesa	Triplet	Siamesa-Sem AE	Triplet-Sem AE
1	31,45%	55,55%	15,34%	24,76%
2	33,69%	64,32%	19,12%	27,43%
3	38,21%	73,65%	24,54%	30,65%
4	39,03%	76,76%	29,32%	31,96%
5	42,90%	79,32%	31,76%	34,21%
6	46,32%	81,21%	32,88%	35,85%
7	51,87%	87,87%	34,56%	38,52%
8	55,62%	87,87%	37,24%	42,05%
9	67,21%	91,94%	41,87%	45,58%
10	75,45%	94,95%	44,09%	47,65%
11	88,10%	96,86%	46,33%	50,95%
12	94,33%	97,41%	48,47%	56,49%
13	96,25%	97,97%	50,03%	59,33%
14	97,77%	98,32%	51,78%	61,48%
15	98,59%	98,99%	54,43%	62,92%

Fonte: Autoria Própria

Figura 34 – Curva CMC, utilizando o *dataset* i-LIDSVID

Fonte: Autoria própria

No caso das com o AE, a Rede Neural Siamesa obteve como pior e melhor acurácia 44,76% e 94,09%, para 100 e 800 épocas, respectivamente. No caso da Rede Neural *Triplet*, a melhor acurácia também foi com 800 épocas de treinamento de 97,62% e a pior de 51,91% com 100 épocas.

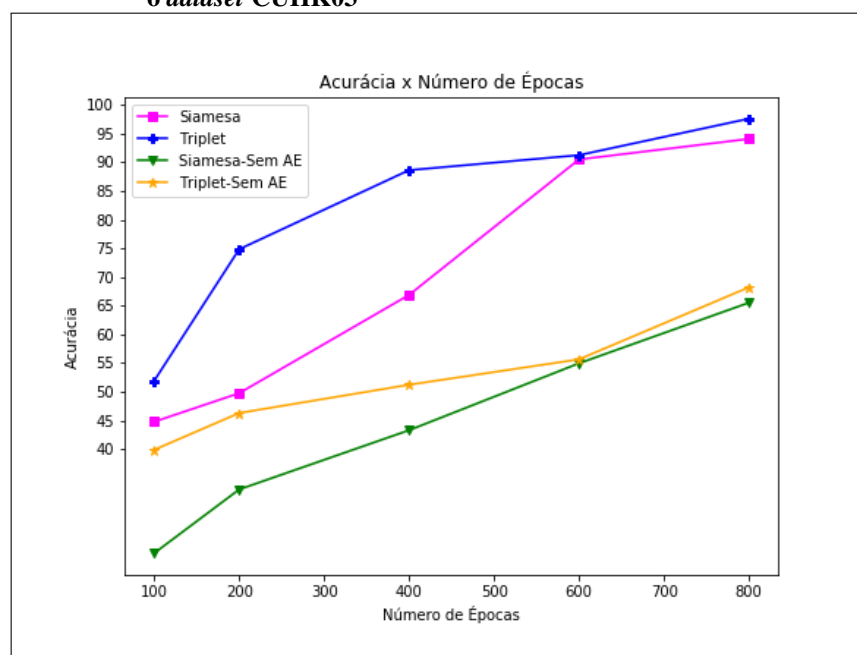
A Tabela 11 e Figura 36 apresentam as Curvas CMC, para a rede treinada e testada com o *dataset* CUHK03. Seguindo o mesmo resultado dos experimentos com os outros *datasets*, as abordagens com o AE tiveram melhores resultados em todos os ranks. Na próxima seção serão discutidos os resultados da curva CMC dos três *datasets* utilizados, bem como uma comparação

Tabela 10 – Acurácia em relação ao N° de Épocas de treinamento, utilizando o dataset CUHK03

N° de Épocas	Siamesa	Triplet	Siamesa-Sem AE	Triplet-Sem AE
100	44,76%	51,91%	21,84%	39,87
200	49,74%	74,85%	32,95%	46,28
400	66,88%	88,64%	43,28%	51,21
600	90,51%	91,26%	54,95%	55,64
800	94,09%	97,62%	65,52%	68,21

Fonte: Autoria Própria

Figura 35 – Acurácia, em relação ao N° de Épocas de treinamento, utilizando o dataset CUHK03



Fonte: Autoria própria

entre as duas redes implementadas com o AE.

5.3 DISCUSSÃO DOS RESULTADOS ENCONTRADOS

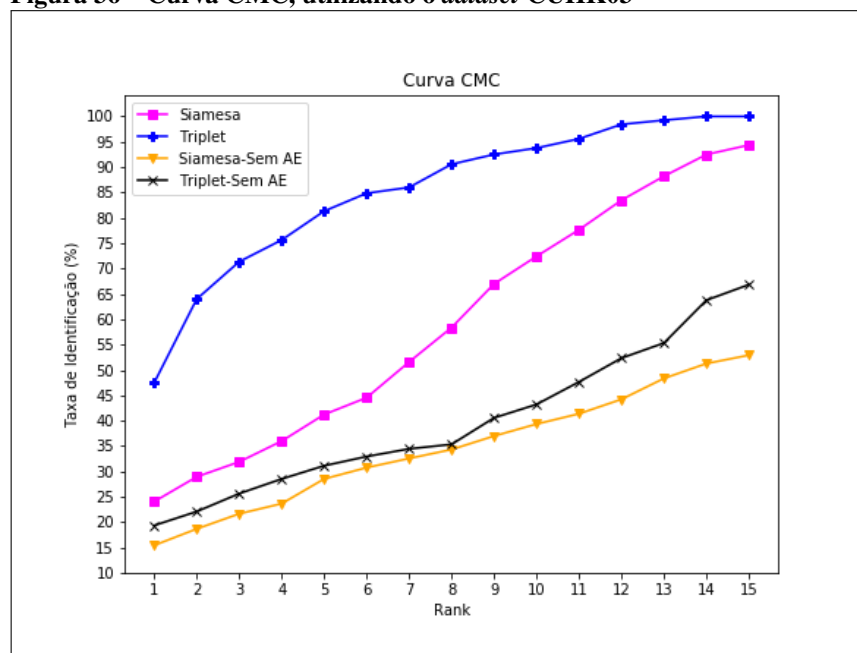
Por meio dos experimentos realizados foi constatado que utilização do AE proporcionou melhores resultados para ambas as abordagens implementadas. Para os resultados com o dataset VIPeR, a acurácia teve um aumento de 43,68% e 35,54% nos experimentos com a Rede Neural Siamesa e Rede Neural *Triplet*, respectivamente. No caso do dataset i-LIDSVID as redes obtiveram as maiores taxas de aumento na acurácia de 71,05% e 62,95%. Os resultados para o dataset também se mostraram melhores com a utilização do AE, sendo que a melhoria da acurácia para cada rede foi de 43,61% e 43,11%.

O gráfico da Figura 37 apresenta todas as acurácias obtidas nos experimentos com a Rede Neural Siamesa e com a Rede Neural *Triplet*, ambas utilizando o AE, uma vez que

Tabela 11 – Curva CMC, utilizando o *dataset* CUHK03

Rank	Siamesa	Triplet	Siamesa-Sem AE	Triplet-Sem AE
1	24,06%	47,50%	15,43%	19,35%
2	28,94%	63,98%	18,67%	22,09%
3	31,89%	71,35%	21,65%	25,63%
4	35,98%	75,63%	23,65%	28,52%
5	41,22%	81,32%	28,52%	31,12%
6	44,54%	84,85%	30,75%	32,94%
7	51,58%	86,00%	32,56%	34,45%
8	58,36%	90,57%	34,32%	35,34%
9	66,97%	92,50%	36,98%	40,56%
10	72,41%	93,76%	39,34%	43,21%
11	77,65%	95,58%	41,37%	47,65%
12	83,47%	98,43%	44,22%	52,34%
13	88,21%	99,25%	48,34%	55,32%
14	92,48%	100,00%	51,27%	63,76%
15	94,35%	100,00%	52,92%	66,83%

Fonte: Autoria Própria

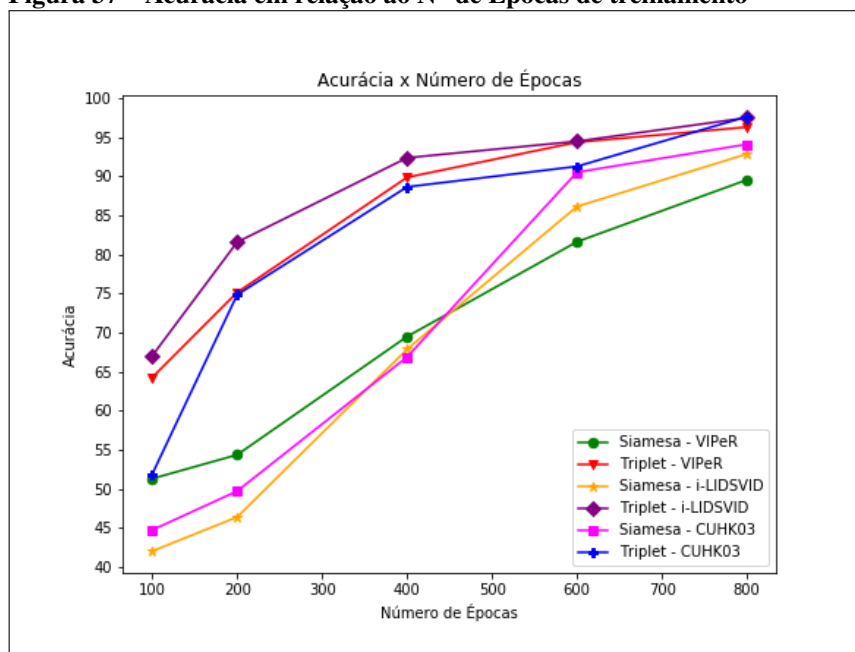
Figura 36 – Curva CMC, utilizando o *dataset* CUHK03

Fonte: Autoria própria

foram as redes que obtiveram os melhores desempenhos. Pode ser observado que a Rede Neural *Triplet* obteve melhores resultados em todos os casos, se comparada com a Rede Neural Siamesa, utilizando os mesmos *datasets* e sub-rede. Uma explicação para isso é que como a Rede Neural *Triplet* recebe como um par de imagens positivo e um negativo (imagem âncora + imagem positiva e imagem âncora + imagem negativa) por entrada, a rede consegue diferenciar imagens com identidades diferentes melhor, bem como aproximar imagens com identidades iguais, em comparação com a rede neural siamesa que recebe apenas um par de imagens positivo ou um par negativo por entrada. Isso pode ser explicado pois em todas as entradas do treinamento ela

pode comparar um par de imagens positivo e um negativo, diferente da Rede Neural Siamesa que apenas recebe um par por entrada.

Figura 37 – Acurácia em relação ao N° de Épocas de treinamento

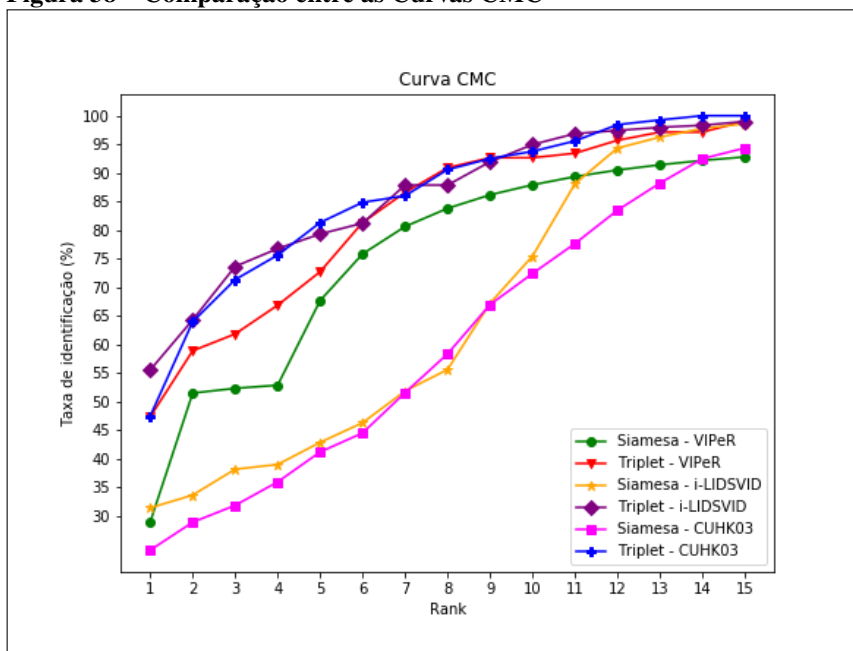


Fonte: Autoria própria

A Figura 38 mostra as Curvas CMC geradas para todos os experimentos realizados. Neste gráfico, observa-se que assim como para as acurácias, a Rede Neural *Triplet* também obteve melhores resultados, comparados com os resultados da Rede Neural Siamesa. Analisando o gráfico pode-se destacar a Curva CMC da Rede Neural Siamesa utilizando o *dataset* i-LIDSVID obteve taxas de identificação muito próximas das taxas obtidas pela outra rede, isso pode ser devido ao fato de que esse *dataset* é do tipo *multi shot*, possuindo mais imagens com uma mesma identidade.

Outro resultado relevante foram as Curvas CMC do *dataset* CUHK03. Em ambas as redes, as Curvas deste *dataset* se iniciou com taxas inferiores aos outros *datasets* para a mesma rede. Uma explicação para isso é que este *dataset* possui um maior número de imagens. No entanto, ao longo do experimento, a Curva CMC da Rede Neural *Triplet*, a partir do *rank-12* obteve as maiores taxas de identificação, sendo que nos dois últimos *ranks*, a rede atingiu 100% de acerto. Isso, significa, que para uma lista com 14 ou 15 identidades mais próximas, de uma imagem, a identidade correta estava presente nessas listas em todos os casos testados. Para uma situação em que for necessário que a rede encontre uma imagem com a mesma identidade de uma imagem de referência, se a rede trouxer uma lista com 15 possíveis identidades, uma delas sempre vai ser a correta. Vale ressaltar que a lista de candidatos total, neste caso, possuía cerca de 30 imagens para cada identidade.

Figura 38 – Comparação entre as Curvas CMC



Fonte: Autoria própria

5.4 COMPARAÇÃO COM O ESTADO DA ARTE

Esta Seção tem como objetivo apresentar uma comparação dos resultados obtidos por *dataset*, após o treinamento e teste dos dois modelos de redes propostas com o AE, com alguns métodos disponíveis no estado da arte. Para comparação dos métodos foi utilizada a Curva CMC.

5.4.1 Dataset VIPeR

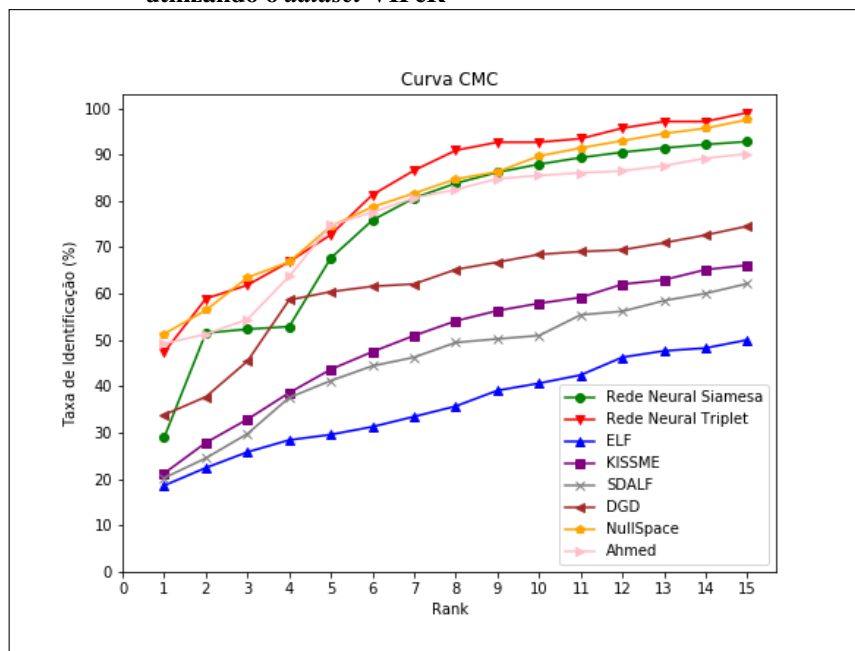
Os métodos utilizados para comparação do desempenho das redes utilizando o *dataset* VIPeR foram: ELF (GRAY; TAO, 2008), KISSME (KOESTINGER *et al.*, 2012), SDALF (FARENZENA *et al.*, 2010), DGD (XIAO *et al.*, 2016), NullSpace (ZHANG; XIANG; GONG, 2016) e o método proposto por Ahmed, Jones e Marks (2015).

A Tabela 12 compara os resultados com as abordagens desenvolvidas neste trabalho. Como pode ser observado a Rede Neural *Triplet*, desenvolvida neste trabalho, teve o melhor desempenho no Rank-2 e a partir do Rank-6 foi a melhor entre os métodos. Nos outros casos, incluindo o Rank-1, essa rede ficou entre os três melhores métodos. Na Figura 39 são plotadas as Curvas CMCs de cada um desses métodos.

Tabela 12 – Comparação das Curva CMC com métodos do estado da arte, utilizando o *dataset* VIPeR

Rank	Rede Neural Siamesa	Rede Neural Triplet	ELF	KISSME	SDALF	DGD	NullSpace	Ahmed
1	28,98%	47,36%	18,62%	21,25%	20,26%	33,87%	51,33%	49,22%
2	51,52%	58,91%	22,44%	27,85%	24,52%	37,74%	56,42%	51,26%
3	52,34%	61,85%	25,84%	32,91%	29,78%	45,45%	63,49%	54,46%
4	52,89%	66,87%	28,43%	38,61%	37,59%	58,64%	66,84%	63,72%
5	67,58%	72,72%	29,61%	43,67%	41,23%	60,41%	74,55%	74,73%
6	75,84%	81,32%	31,29%	47,47%	44,41%	61,58%	78,69%	77,54%
7	80,62%	86,58%	33,49%	50,95%	46,27%	62,08%	81,65%	80,65%
8	83,80%	90,90%	35,73%	54,11%	49,47%	65,21%	84,71%	82,45%
9	86,16%	92,65%	39,12%	56,33%	50,23%	66,79%	86,28%	84,78%
10	87,91%	92,65%	40,67%	57,91%	50,96%	68,46%	89,67%	85,48%
11	89,34%	93,43%	42,46%	59,18%	55,43%	69,07%	91,43%	86,04%
12	90,50%	95,69%	46,28%	62,03%	56,19%	69,45%	92,98%	86,48%
13	91,41%	97,12%	47,66%	62,97%	58,49%	70,96%	94,52%	87,58%
14	92,17%	97,12%	48,27%	65,19%	60,05%	72,64%	95,67%	89,21%
15	92,79%	99,04%	50,03%	66,14%	62,17%	74,56%	97,52%	90,15%

Fonte: Autoria Própria

Figura 39 – Comparação das Curva CMC com métodos do estado da arte, utilizando o *dataset* VIPeR

Fonte: Autoria própria

5.4.2 Dataset i-LIDSVID

O *dataset* i-LIDSVID foi utilizado em experimentos para comparação com os seguintes métodos disponíveis no estado da arte: KISSME (KOESTINGER *et al.*, 2012), SDALF (FARENZENA *et al.*, 2010), DGD (XIAO *et al.*, 2016) e Ahmed, Jones e Marks (2015). O gráfico

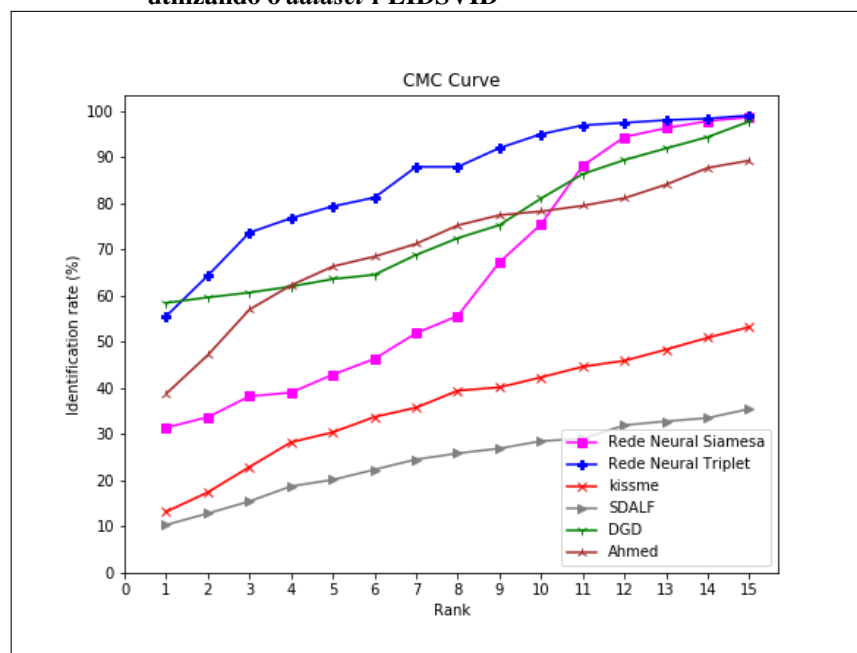
da Figura 40 mostras as curvas CMC dos experimentos realizados. Os resultados individuais de cada método são apresentados na Tabela 13

Tabela 13 – Comparação das Curva CMC com métodos do estado da arte, utilizando o dataset i-LIDSVID

Rank	Rede Neural Siamesa	Rede Neural Triplet	KISSME	SDALF	DGD	Ahmed
1	31,45%	55,55%	13,23%	10,39%	58,43%	38,76%
2	33,69%	64,32%	17,44%	12,87%	59,61%	47,13%
3	38,21%	73,65%	22,92%	15,48%	60,66%	56,98%
4	39,03%	76,76%	28,28%	18,76%	61,98%	62,24%
5	42,90%	79,32%	30,42%	20,14%	63,58%	66,32%
6	46,32%	81,21%	33,74%	22,35%	64,52%	68,45%
7	51,87%	87,87%	35,76%	24,56%	68,81%	71,22%
8	55,62%	87,87%	39,43%	25,87%	72,44%	75,21%
9	67,21%	91,94%	40,15%	26,93%	75,27%	77,42%
10	75,45%	94,95%	42,32%	28,52%	81,03%	78,23%
11	88,10%	96,86%	44,63%	29,05%	86,32%	79,46%
12	94,33%	97,41%	45,91%	31,96%	89,35%	81,09%
13	96,25%	97,97%	48,33%	32,80%	91,86%	84,03%
14	97,77%	98,32%	50,90%	33,53%	94,33%	87,66%
15	98,59%	98,99%	53,21%	35,47%	97,69%	89,23%

Fonte: Autoria Própria

Figura 40 – Comparação das Curva CMC com métodos do estado da arte, utilizando o dataset i-LIDSVID



Fonte: Autoria própria

Analisando a Tabela 13, pode-se observar que nesses experimentos a Rede Neural Triplet obteve os melhores resultados a partir do rank-2. Sendo que no rank-1 o método DGD

de Xiao *et al.* (2016) obteve o melhor resultado, no entanto, o método da Rede Neural *Triplet* ficou em segundo lugar. A Rede Neural Siamesa ficou em segundo lugar em todos os casos a partir do rank-11.

5.4.3 Dataset CUHK03

Com o *dataset* CUHK03, as abordagens desenvolvidas neste trabalho foram comparadas com seis métodos disponíveis no estado da arte, são eles: KISSME (KOESTINGER *et al.*, 2012), SDALF (FARENZENA *et al.*, 2010), DGD (XIAO *et al.*, 2016), NullSpace (ZHANG; XIANG; GONG, 2016), NormXcorr (SUBRAMANIAM; CHATTERJEE; MITTAL, 2016) e o método de Ahmed, Jones e Marks (2015). Na Tabela 14 podem ser observados os resultados para a curva CMC para ranks de 1 à 15.

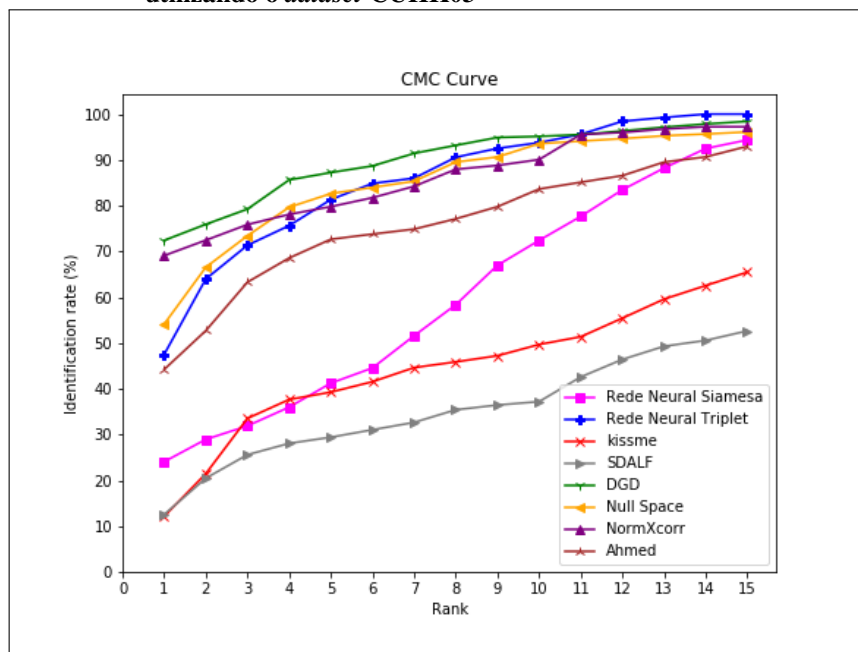
Tabela 14 – Comparação das Curva CMC, com métodos do estado da arte,, utilizando o *dataset* CUHK03

Rank	Rede Neural Siamesa	Rede Neural Triplet	KISSME	SDALF	DGD	Null Space	Norm Xcorr	Ahmed
1	24,06%	47,50%	12,12%	12,52%	72,41%	54,12%	69,08%	44,21%
2	28,94%	63,98%	21,45%	20,46%	75,92%	66,54%	72,45%	52,69%
3	31,89%	71,35%	33,53%	25,59%	79,25%	73,42%	75,87%	63,29%
4	35,98%	75,63%	37,65%	28,07%	85,62%	79,65%	78,09%	68,56%
5	41,22%	81,32%	39,26%	29,41%	87,23%	82,63%	79,78%	72,65%
6	44,54%	84,85%	41,54%	31,04%	88,67%	83,98%	81,73%	73,78%
7	51,58%	86,00%	44,62%	32,65%	91,42%	85,32%	84,22%	74,87%
8	58,36%	90,57%	45,87%	35,41%	93,17%	89,52%	87,94%	77,12%
9	66,97%	92,50%	47,21%	36,44%	94,89%	90,63%	88,78%	79,77%
10	72,41%	93,76%	49,67%	37,19%	95,13%	93,54%	90,06%	83,63%
11	77,65%	95,58%	51,36%	42,58%	95,51%	94,09%	95,54%	85,11%
12	83,47%	98,43%	55,43%	46,41%	96,32%	94,64%	95,58%	86,56%
13	88,21%	99,25%	59,58%	49,28%	97,14%	95,26%	96,73%	89,52%
14	92,48%	100%	62,54%	50,56%	97,84%	95,65%	97,23%	90,64%
15	94,35%	100%	65,48%	52,65%	98,41%	96,09%	97,23%	92,96%

Fonte: Autoria Própria

Nestes experimentos, o método DGD (XIAO *et al.*, 2016) teve uma melhor taxa de identificação nos dez primeiros ranks. No restante, a Rede Neural *Triplet* desenvolvida neste trabalho obteve os melhores resultados, atingindo uma taxa de 100% no rank-14 e rank-15. Vale ressaltar que a Rede Neural *Triplet*, ficou entre os dois melhores métodos a partir do rank-6. As Curvas CMCs são mostradas no gráfico da Figura 41.

Figura 41 – Comparação das Curva CMC com métodos do estado da arte, utilizando o *dataset* CUHK03



Fonte: Autoria própria

6 CONCLUSÃO

Este trabalho teve como principal objetivo desenvolver e comparar duas abordagens para re-identificar pessoas em imagens digitais. Uma das abordagens consiste em uma Rede Neural Siamesa e a outra em uma Rede Neural *Triplet*. Para isso, foi desenvolvido um modelo de sub-rede composto por uma CNN e um AE. A principal função da CNN neste caso é a extração de características da imagem e o AE foi utilizado para reconstruir o vetor gerado pela CNN, de forma a manter as características mais importantes.

Com isso, a Rede Neural Siamesa recebe duas imagens de entrada, que devem passar cada uma por uma sub-rede e retornar um vetor de características. A distância Euclidiana desses vetores é calculada e a rede deve informar se pertencem ou não a mesma pessoa. Já a Rede Neural *Triplet* recebe três imagens de entrada, sendo que duas são de uma mesma identidade e outra de uma identidade diferente. Após passar pelas sub-redes, a rede deve reduzir a distância entre os vetores das imagens com a mesma identidade e aumentar a distância entre imagens com identidades diferentes.

Para realização dos experimentos foram selecionados três *datasets* diferentes, sendo eles VIPeR (GRAY; BRENNAN; TAO, 2007), i-LIDSVID (WANG *et al.*, 2014) e CUHK03 (LI *et al.*, 2014) e aplicadas duas medidas de avaliação: acurácia e curva CMC. A acurácia calcula o desempenho de uma rede através do número total de acertos em relação ao total de amostras de testes. Já a Curva CMC calcula uma taxa de acerto com base em uma lista dos candidatos mais próximos de uma amostra analisada, através do conceito de *ranks*, ou seja, se o primeiro candidato da lista corresponder a mesma identidade da amostra analisada, a taxa de acerto, neste caso, será de 100% .

Foram realizados experimentos com e sem o AE no final das sub-redes. Com isso, foi possível verificar qual o impacto do AE nas redes. Foi observado que a utilização do AE gerou um ganho significativo nos resultados, tanto para a Rede Neural Siamesa, quanto para a Rede Neural *Triplet*. Com o AE, as taxas de acurácia aumentaram em todos os casos, assim como a Curva CMC, sendo que foi constatado que o AE proporcionou um ganho de até 71.05%. O AE também tornou o treinamento da rede mais estável, aumentando a acurácia a medida que o número de épocas aumentou.

A partir dos experimentos efetuados utilizando as redes com o AE, também foi constatado que a Rede Neural *Triplet* obteve um melhor desempenho na re-identificação de pessoas se comparada com a Rede Neural Siamesa, em todos os casos testados ela se comportou melhor. Uma explicação para isso é que a Rede Neural *Triplet* é capaz comparar um par negativo de amostras e um par positivo em todos os casos de treinamento, conseguindo reduzir a distância entre as imagens dos pares positivos de forma mais eficiente.

Além disso, as abordagens desenvolvidas também foram comparadas com alguns mé-

todos disponíveis no estado da arte. Analisando os experimentos foi constatado que Rede Neural *Triplet* obteve bons resultados na re-identificação de pessoas em imagens. Na maioria dos experimentos essa rede ficou entre as duas melhores se comparadas com os outros métodos reproduzidos e, em alguns casos, a rede obteve o melhor desempenho. Com o *dataset* VIPeR a rede ficou em 12 *ranks* de 15 em primeiro lugar. Já com o *dataset* CUHK03 a rede atingiu a taxa de 100% de acerto nos *ranks*-14 e 15.

Com isso, pode-se concluir que a Rede Neural *Triplet* desenvolvida neste trabalho apresenta um grande potencial para re-identificar pessoas em imagens digitais, em comparação com a Rede Neural Siamesa e também com outros métodos disponíveis na literatura que foram implementados. Essa rede ganhou em grande parte dos resultados nas Curvas CMC, para os três *datasets* utilizados. Experimentos futuros podem ser realizados, a fim de analisar o comportamento das redes em outros cenários.

Um desses cenários seria adaptação das redes desenvolvidas para a re-identificação de pessoas em uma base com vídeos, e não apenas com imagens, como é o caso deste trabalho. Com isso, testes devem ser realizados para analisar como as redes irão se comportar e se haverá uma melhoria ou não nas taxas de acerto. Pensando em vídeo, também sugere-se realizar a re-identificação em um ambiente monitorado por várias câmeras, assim, a partir de uma imagem de uma pessoa vista por uma câmera deve-se encontrar a da mesma pessoa em imagens geradas por outra câmera, sendo que várias pessoas podem transitar nesse ambiente.

Além disso, outros trabalhos futuros são sugeridos, como a realização de experimentos utilizando outros *datasets* e também a criação de um *dataset* próprio para validação das redes. O desenvolvimento de um *dataset* próprio poderia abordar outros problemas na área de re-identificação de pessoas, como, por exemplo, oclusões de partes do corpo.

Também sugere-se a implementação de um modelo de Rede Neural *Quadriplet* utilizando a mesma sub-rede desse trabalho. Essa rede poderia comprovar se aumentando o número de entradas e, conseqüentemente, o número de sub-redes, o potencial para re-identificar pessoas em imagens sempre vai aumentar. Neste modelo de rede, a entrada consiste em uma imagem âncora, uma imagem positiva e duas imagens negativas, que não podem ser da mesma pessoa. Experimentos devem ser realizados para comparar com a Rede Neural Siamesa e com a Rede Neural *Triplet* e, assim, verificar se os resultados seguem a mesma regra deste trabalho, ou seja, se aumentando o número de sub-redes, o potencial de re-identificação da rede também aumenta.

REFERÊNCIAS

- ABADI, M. *et al.* **TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems**. 2015. Software available from tensorflow.org. Disponível em: <<https://www.tensorflow.org/>>. Acesso em: 15 out. 2018.
- AHMED, E.; JONES, M.; MARKS, T. K. An improved deep learning architecture for person re-identification. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **Proceedings...** [S.l.], 2015. p. 3908–3916.
- AVRAHAM, T. *et al.* Learning implicit transfer for person re-identification. In: EUROPEAN CONFERENCE ON COMPUTER VISION. **Proceedings...** [S.l.], 2012. p. 381–390.
- BALTIERI, D.; VEZZANI, R.; CUCCHIARA, R. 3dpes: 3d people dataset for surveillance and forensics. In: ACM WORKSHOP ON HUMAN GESTURE AND BEHAVIOR UNDERSTANDING. **Proceedings...** [S.l.], 2011. p. 59–64.
- BEDAGKAR-GALA, A.; SHAH, S. K. A survey of approaches and trends in person re-identification. **Image and Vision Computing**, Elsevier, v. 32, n. 4, p. 270–286, 2014.
- BEZERRA, E. Introdução à aprendizagem profunda. **Tópicos em Gerenciamento de Dados e Informações**, Sociedade Brasileira de Computação, Porto Alegre, p. 57–86, 2016.
- BEZERRA, S. **Reservoir Computing com Hierarquia para Previsão de Vazões Médias Diárias**. Tese (Doutorado), 07 2016.
- BOLLE, R. M. *et al.* The relation between the roc curve and the cmc. In: IEEE WORKSHOP ON AUTOMATIC IDENTIFICATION ADVANCED TECHNOLOGIES (AUTOID'05). **Proceedings...** [S.l.], 2005. p. 15–20.
- BROMLEY, J. *et al.* Signature verification using a "siamese" time delay neural network. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS. **Proceedings...** [S.l.], 1994. p. 737–744.
- CHENG, D. *et al.* Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **Proceedings...** [S.l.], 2016. p. 1335–1344.
- CHENG, D. S. *et al.* Custom pictorial structures for re-identification. In: BMVC. **Proceedings...** [S.l.], 2011. v. 1, n. 2, p. 6.
- CHOLLET, F. **Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek**. [S.l.]: MITP-Verlags GmbH & Co. KG, 2018.
- CHOLLET, F. *et al.* **Keras**. 2015. Disponível em: <<https://keras.io>>. Acesso em: 15 out. 2018.
- CHOPRA, S. *et al.* Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.], 2005. p. 539–546.
- DECANN, B.; ROSS, A. Relating roc and cmc curves via the biometric menagerie. In: IEEE SIXTH INTERNATIONAL CONFERENCE ON BIOMETRICS: THEORY, APPLICATIONS AND SYSTEMS (BTAS). **Proceedings...** [S.l.], 2013. p. 1–8.

- DENG, L.; YU, D. Deep learning: methods and applications. **Foundations and Trends in Signal Processing**, Now Publishers, Inc., v. 7, n. 3–4, p. 197–387, 2014.
- DICTIONARY, Oxford. Oxford dictionaries. **Language Matters**, 2014.
- DIETTERICH, T. Overfitting and undercomputing in machine learning. **ACM computing surveys (CSUR)**, ACM New York, NY, USA, v. 27, n. 3, p. 326–327, 1995.
- DUNSTONE, Ted; YAGER, Neil. **Biometric system and data analysis: Design, evaluation, and data mining**. [S.l.]: Springer Science & Business Media, 2008.
- FARENZENA, M *et al.* Person re-identification by symmetry-driven accumulation of local features. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.], 2010. p. 2360–2367.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE. **Deep learning**. [S.l.]: MIT press Cambridge, 2016. v. 1.
- GRAUPE, D. **Principles of Artificial Neural Networks**. 2. ed. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2007. ISBN 9812706240.
- GRAY, D.; BRENNAN, S.; TAO, H. Evaluating appearance models for recognition, reacquisition, and tracking. In: IEEE INTERNATIONAL WORKSHOP ON PERFORMANCE EVALUATION FOR TRACKING AND SURVEILLANCE (PETS). **Proceedings...** [S.l.], 2007. v. 3, n. 5, p. 1–7.
- GRAY, D.; TAO, H. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In: EUROPEAN CONFERENCE ON COMPUTER VISION. **Proceedings...** [S.l.], 2008. p. 262–275.
- HADSELL, R.; CHOPRA, S.; LECUN, Y. Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.], 2006. v. 2, p. 1735–1742.
- HAYKIN S., S. **Neural networks and learning machines**. 3. ed. Upper Saddle River, NJ: Pearson Education, 2009.
- HIRZER, M. *et al.* Person re-identification by descriptive and discriminative classification. In: SCANDINAVIAN CONFERENCE ON IMAGE ANALYSIS. **Proceedings...** [S.l.], 2011. p. 91–102.
- HOFFER, E; AILON, N. Deep metric learning using triplet network. In: INTERNATIONAL WORKSHOP ON SIMILARITY-BASED PATTERN RECOGNITION. **Proceedings...** [S.l.], 2015. p. 84–92.
- IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: 32ND INTERNATIONAL CONFERENCE ON MACHINE LEARNING. **Proceedings...** [S.l.]: JMLR.org, 2015. (ICML'15), p. 448–456.
- JR, A. C. Nazare; SCHWARTZ, W. R. A scalable and flexible framework for smart video surveillance. **Computer Vision and Image Understanding**, Elsevier, v. 144, p. 258–275, 2016.

- JUBA, B.; LE, H S. Precision-recall versus accuracy and the role of large data sets. In: AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE. **Proceedings...** [S.l.], 2019. v. 33, p. 4039–4048.
- KOCH, G.; ZEMEL, R.; SALAKHUTDINOV, R. Siamese neural networks for one-shot image recognition. In: ICML DEEP LEARNING WORKSHOP. **Proceedings...** [S.l.], 2015. v. 2.
- KOESTINGER, M *et al.* Large scale metric learning from equivalence constraints. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.], 2012. p. 2288–2295.
- KRIESEL, D. **A Brief Introduction to Neural Networks.** [S.l.: s.n.], 2007.
- KRISHNA, R. **Computer Vision: foundations and applications.** [S.l.]: Stanford University, 2017.
- LECUN, Y.; BENGIO, Y. Convolutional networks for images, speech, and time series. **The handbook of brain theory and neural networks**, v. 3361, n. 10, p. 1995, 1995.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015.
- LI, W.; ZHAO, R.; WANG, X. Human reidentification with transferred metric learning. In: ACCV. **Proceedings...** [S.l.], 2012.
- LI, W. *et al.* Deepreid: Deep filter pairing neural network for person re-identification. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **Proceedings...** [S.l.], 2014. p. 152–159.
- MCLAUGHLIN, N.; RINCON, J. Martinez del; MILLER, P. Recurrent convolutional network for video-based person re-identification. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **Proceedings...** [S.l.], 2016. p. 1325–1334.
- MITCHELL, T M. **Machine Learning.** 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072.
- NETO, F. G. M. *et al.* Aprendizado profundo: conceitos, técnicas e estudo de caso de análise de imagens com java.
- PROSSER, B. J. *et al.* Person re-identification by support vector ranking. In: BMVC. **Proceedings...** [S.l.], 2010. v. 2, n. 5, p. 6.
- SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.], 2015. p. 815–823.
- SCHWARTZ, W.R.; DAVIS, L.S. Learning Discriminative Appearance-Based Models Using Partial Least Squares. In: **Proceedings...** [S.l.: s.n.], 2009.
- SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding machine learning: From theory to algorithms.** [S.l.]: Cambridge university press, 2014.

- SHARMA, S. **Activation Functions in Neural Networks**. 2017. Disponível em: <<https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>>. Acesso em: 20 jan. 2019.
- SOKOLOVA, M; JAPKOWICZ, N; SZPAKOWICZ, S. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In: AUSTRALASIAN JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE. **Proceedings...** [S.l.], 2006. p. 1015–1021.
- SRIVASTAVA, N *et al.* Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.
- SUBRAMANIAM, A; CHATTERJEE, M; MITTAL, A. Deep neural networks with inexact matching for person re-identification. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS. **Proceedings...** [S.l.], 2016. p. 2667–2675.
- SZELISKI, R. **Computer Vision: Algorithms and Applications**. 1. ed. Berlin, Heidelberg: Springer-Verlag, 2010. ISBN 1848829345, 9781848829343.
- TU, P. H. *et al.* An intelligent video framework for homeland protection. In: UNATTENDED GROUND, SEA, AND AIR SENSOR TECHNOLOGIES AND APPLICATIONS IX. **Proceedings...** [S.l.], 2007. v. 6562, p. 65620C.
- WANG, J. *et al.* Learning fine-grained image similarity with deep ranking. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.], 2014. p. 1386–1393.
- WANG, J *et al.* Transferable joint attribute-identity deep learning for unsupervised person re-identification. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.], 2018. p. 2275–2284.
- WANG, N.; YEUNG, D. Learning a deep compact image representation for visual tracking. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS. **Proceedings...** [S.l.], 2013. p. 809–817.
- WANG, T. *et al.* Person re-identification by video ranking. In: EUROPEAN CONFERENCE ON COMPUTER VISION. **Proceedings...** [S.l.], 2014. p. 688–703.
- WU, S. *et al.* An enhanced deep feature representation for person re-identification. In: IEEE WINTER CONFERENCE ON APPLICATIONS OF COMPUTER VISION (WACV). **Proceedings...** [S.l.], 2016. p. 1–8.
- XIAO, T *et al.* Learning deep feature representations with domain guided dropout for person re-identification. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.], 2016. p. 1249–1258.
- YI, D. *et al.* Deep metric learning for person re-identification. In: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION (CVPR) (ICPR), 22. **Proceedings...** [S.l.], 2014. p. 34–39.
- ZHANG, L; XIANG, T; GONG, S. Learning a discriminative null space for person re-identification. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.], 2016.

ZHAO, R.; OUYANG, W.; WANG, X. Unsupervised saliency learning for person re-identification. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. **Proceedings...** [S.l.], 2013. p. 3586–3593.

ZHENG, L. *et al.* Scalable person re-identification: A benchmark. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION. **Proceedings...** [S.l.], 2015. p. 1116–1124.

ZHENG, Z; ZHENG, L; YANG, Y. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION. **Proceedings...** [S.l.], 2017. p. 3754–3762.