

UNIVERSIDADE DO ESTADO DO AMAZONAS - UEA
ESCOLA SUPERIOR DE TECNOLOGIA
SISTEMAS DE INFORMAÇÃO

KID MENDES DE OLIVEIRA NETO

**SMARTGLASS: UMA PROPOSTA DE INTERFACE DE REALIDADE
AUMENTADA INTELIGENTE PARA O RECONHECIMENTO FACIAL DE
ALUNOS UTILIZANDO REDES NEURAIS CONVOLUCIONAIS**

Manaus

2019

KID MENDES DE OLIVEIRA NETO

**SMARTGLASS: UMA PROPOSTA DE INTERFACE DE REALIDADE
AUMENTADA INTELIGENTE PARA O RECONHECIMENTO FACIAL DE
ALUNOS UTILIZANDO REDES NEURAIS CONVOLUCIONAIS**

Trabalho de Conclusão de Curso apresentado à banca avaliadora do Curso de Sistemas de Informação, da Escola Superior de Tecnologia, da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Jucimar Maia da Silva Junior

Manaus – Dezembro – 2019

Universidade do Estado do Amazonas - UEA
Escola Superior de Tecnologia - EST

Reitor:

Cleinaldo de Almeida Costa

Vice-Reitor:

Cleto Cavalcante de Souza Leal

Diretora da Escola Superior de Tecnologia:

Ingrid Sammyne Gadelha Figueiredo

Coordenadora do Curso de Sistemas de Informação:

Marcela Sávia Picanço Pessoa

Coordenadora da Disciplina Trabalho de Conclusão de Curso 2:

Polianny Almeida Lima

Banca Avaliadora composta por:

Data da Defesa: 05/12/2019.

Prof. Jucimar Maia da Silva Junior, D.Sc. (Orientador)

Prof. Ricardo da Silva Barboza, D.Sc.

Prof. Mário Augusto Bessa de Figueiredo, M.Sc.

CIP – Catalogação na Publicação

L864a MENDES, Kid de Oliveira Neto

SmartGlass: Uma proposta de interface de Realidade Aumentada inteligente para o Reconhecimento Facial de alunos utilizando Redes Neurais Convolucionais / Kid Mendes; [orientado por] Prof. Dr. Jucimar Maia da Silva Junior – Manaus: UEA, 2019.

80 p.: il.; 30cm

Inclui Bibliografia

Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação). Universidade do Estado do Amazonas, 2019.

CDU: _____

KID MENDES DE OLIVEIRA NETO

**SMARTGLASS: UMA PROPOSTA DE INTERFACE DE REALIDADE
AUMENTADA INTELIGENTE PARA O RECONHECIMENTO FACIAL DE
ALUNOS UTILIZANDO REDES NEURAIS CONVOLUCIONAIS**

Trabalho de Conclusão de Curso apresentado
à banca avaliadora do Curso de Sistemas de
Informação, da Escola Superior de Tecnologia,
da Universidade do Estado do Amazonas, como
pré-requisito para obtenção do título de Bacharel
em Sistemas de Informação.

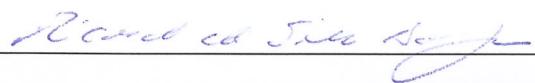
BANCA EXAMINADORA

Aprovado em: 05/12/2019



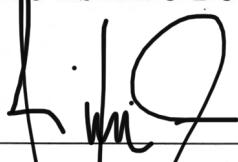
Prof. Jucimar Maia da Silva Junior, D.Sc.

UNIVERSIDADE DO ESTADO DO AMAZONAS



Prof. Ricardo da Silva Barboza, D.Sc.

UNIVERSIDADE DO ESTADO DO AMAZONAS



Prof. Mário Augusto Bessa de Figueiredo, M.Sc.

UNIVERSIDADE DO ESTADO DO AMAZONAS

Resumo

Este trabalho apresenta uma proposta para o desenvolvimento de uma interface de Realidade Aumentada inteligente para o Reconhecimento Facial de alunos da Escola Superior de Tecnologia da Universidade do Estado do Amazonas utilizando técnicas de *Deep Learning*, de modo a auxiliar o controle da entrada de alunos na universidade. Para tanto, a abordagem escolhida para a tarefa de reconhecimento facial, foi o *One-Shot Learning* implementada em uma rede siamesa, contendo duas Redes Neurais Convolucionais idênticas totalmente conectadas. Nesta tarefa foram propostos cenários de treino e teste de diferentes arquiteturas de redes neurais convolucionais, submetidas a variações de parâmetros e hiperparâmetros. Após obter o melhor modelo, este ficará embarcado em um servidor que utiliza o protocolo *WebSocket*, sendo responsável pela transmissão e recepção dos dados na aplicação de Realidade Aumentada. Os resultados obtidos mostraram que a interface inteligente pôde auxiliar em uma melhor execução das três características impostas pelo Azuma, para o melhor funcionamento de um sistema de realidade aumentada, em especial, a execução interativa em tempo real foi a mais beneficiada em virtude do alto desempenho do modelo proposto.

Palavras Chave: Reconhecimento Facial, Realidade Aumentada, *Deep Learning*, Redes Neurais Convolucionais.

Abstract

This work presents a proposal for the development of an intelligent Augmented Reality interface for the Facial Recognition of students at the School of Technology of the State University of Amazonas using Deep Learning techniques, in order to help control the entry of students into the university. The approach chosen for the facial recognition task was One-Shot Learning implemented in a Siamese network, containing two identical Convolutional Neural Networks fully connected. In this task, training and testing scenarios of different convolutional neural network architectures were proposed, subject to parameter and hyperparameter variations. After obtaining the best model, it will be embedded in a server that uses the WebSocket protocol, being responsible for the transmission and reception of data in the Augmented Reality application. The results obtained showed that the intelligent interface was able to help in a better execution of the three characteristics imposed by Azuma, for a better functioning of an augmented reality system, in particular, the interactive execution in real time was the most benefited due to the high performance of the proposed model.

Keywords: *Facial Recognition, Augmented Reality, Deep Learning, Convolutional Neural Networks.*

Agradecimentos

Agradeço primeiramente à minha mãe, Kátia Regina Farias de Oliveira, por ter me incentivado durante toda a vida, independente da situação, a buscar uma formação que traga realização profissional. Agradeço e reconheço todo o esforço exaustivo para proporcionar uma vida melhor, consequentemente permitindo a melhor educação que podia oferecer, moldando todo o meu caráter e ensinando desde cedo a voar sozinho, mesmo caindo, podia contar com seu suporte. Apesar de não se considerar a melhor mãe, digo indubitavelmente que é a mãe dos sonhos devido ao desempenho do seu papel de uma maneira fenomenal que apenas pais dedicados conseguem realizar. Embora eu não tenha uma religião no momento, agradeço as orações da minha mãe pra mim, um gesto sempre apreciado.

Agradeço meu orientador, Prof. Dr. Jucimar Maia da Silva Junior, pelos conhecimentos desde o início da graduação até o final, desde ensinar programação à orientar no projeto de conclusão de curso. Apesar de sua seriedade e severidade, é um profissional que sempre tenta ajudar o máximo seus alunos, guiando para um melhor futuro e sempre incentivando a desbravar o mundo. Agradeço pela oportunidade, confiança e dedicação que se fizeram fundamentais para meu crescimento profissional e para a concretização deste trabalho.

Agradeço também aos outros membros da minha família, em especial minhas irmãs e meu pai, ambos fundamentais para meu crescimento e estabilidade financeira necessários ao longo da minha graduação. Agradeço aos meus amigos, por todos os momentos importantes e memoráveis nesta jornada.

Agradeço ao Núcleo de Computação e a todos os professores e coordenadores que proporcionaram imenso aprendizado durante a graduação. Agradeço também à Universidade do Estado do Amazonas pelo apoio de seus servidores, prestadores de serviços e por toda a infraestrutura

fornecida. Os resultados foram obtidos através das atividades de pesquisa e desenvolvimento no Samsung Ocean Center, patrocinado pela Samsung Electronics of Amazonia Ltda., com o apoio da SUFRAMA nos termos da Lei Federal nº8.248/91.

Epígrafe

“The cave you fear to enter holds the treasure you seek”

Joseph Campbell

Sumário

Lista de Tabelas	x
Lista de Figuras	xii
1 Introdução	1
1.1 Objetivos	3
1.1.1 Objetivo Geral	3
1.1.2 Objetivos Específicos	3
1.2 Justificativa	3
1.3 Metodologia	4
1.4 Organização do Documento	5
2 Fundamentação Teórica	6
2.1 Realidade Aumentada	6
2.1.1 Conceitos e Definições	6
2.1.2 Interfaces	9
2.1.3 Realidade Aumentada Contemporânea	12
2.2 Redes Neurais Artificiais	18
2.2.1 Neurônio Artificial	18
2.2.2 <i>Multilayer Perceptron</i>	25
2.3 <i>Deep Learning</i>	26
2.3.1 Redes Neurais Convolucionais	27

2.4	Trabalhos Relacionados	31
3	Solução Proposta	33
3.1	Descrição Geral da Proposta	33
3.2	Projeto SmartGlass	34
3.2.1	Levantamento de Requisitos	35
3.2.2	Diagramas de Caso de Uso	37
3.2.3	Arquitetura do SmartGlass	38
3.2.4	Módulo Aplicação de Realidade Aumentada	40
3.2.5	Módulo Servidor	41
3.2.6	Modelos Propostos	43
3.2.7	Pré-processamento dos dados	51
4	Resultados e Discussão	53
4.1	Detectar Faces: Cascade Classifier e Multi-task Cascade CNN	53
4.1.1	Cascade Classifier	54
4.1.2	Multi-task Cascade CNN	54
4.1.3	Cascade Classifier x Multi-task Cascade CNN	55
4.2	Abordagem 1:FaceNet	56
4.3	Abordagem 2: ResNet-50 com VGGFace2	57
5	Considerações Finais	59

Lista de Tabelas

3.1	Requisito Funcional RF01	35
3.2	Requisito Funcional RF02	35
3.3	Requisito Não-Funcional RNF01	36
3.4	Requisito Não-Funcional RNF02	36
3.5	Requisito Não-Funcional RNF03	36
3.6	Descrição do Caso de Uso UC01	37
3.7	Descrição do Caso de Uso UC02	38
4.1	Resultados da FaceNet na Abordagem 1	57
4.2	Resultados da ResNet-50 com VGGFace2 na Abordagem 2	58

Lista de Figuras

2.1	Representação Simplificada de Contínuo Realidade-Virtualidade (adaptada de (MILGRAM et al., 1994)) (KIRNER; TORI, 2006)	8
2.2	Usuário utiliza a raquete para pegar, mover, soltar ou destruir modelos (Azuma et al., 2001)	9
2.3	Aplicação AnimAR (LOPES; REITER; REIS, 2018)	10
2.4	Visualização de Realidade Aumentada para Cirurgia Laparoscópica(FUCHS et al., 1998)	11
2.5	O kit Móvel de AR: (a) o sistema de processador único para alvos simples, (b) o sistema de processador duplo para cenários mais complexos; e (c) o kit de AR em uso. (RIBO et al., 2002)	11
2.6	Informação aumentada para os objetos específicos nas fotos (LEE et al., 2011) .	12
2.7	Tecnologias de Realidade Aumentada (CARUSO, 2017)	13
2.8	A evolução da interação (KUNKEL et al., 2016)	14
2.9	Google Glass (HOOKEY, 2015)	16
2.10	Como o Google Glass funciona (MISSFELDT, 2017)	16
2.11	Crie espaços e mostre-os a outras pessoas em escala real (MICROSOFT, 2019) .	17
2.12	Ajude os funcionários a aprender com instruções passo a passo (MICROSOFT, 2019)	17
2.13	Modelo de neurônio artificial (MARTINS-FILHO; MOL; ROCHA, 2005)	19
2.14	Função de ativação linear	20
2.15	Função de ativação degrau	21

2.16	Função de ativação sigmoidal	22
2.17	Função de ativação tangente hiperbólica	22
2.18	Função de ativação ReLU	23
2.19	Perceptron de camada única (BRAGA; CARVALHO; LUDELMIR, 2007)	24
2.20	Função linearmente separável e Função não linearmente separável (BOOK, 2016)	24
2.21	Representação de uma RNA <i>Multilayer Perceptron</i> (MOREIRA, 2018)	25
2.22	Exemplo de Arquitetura CNN para Classificação de Imagem (RAWAT; WANG, 2017)	28
2.23	Processo de Convolução, <i>kernel</i> 3x3, S=1 (ARAUJO, 2018)	29
2.24	Aplicação de <i>max pooling</i> em uma imagem 4x4 utilizando um filtro 2x2 (ARAUJO et al., 2017)	30
2.25	Ilustração da extração de características de uma imagem por uma CNN e sua posterior classificação (ARAUJO et al., 2017)	31
3.1	Visão Geral da Proposta	34
3.2	Diagrama de Caso de Uso do SmartGlass	37
3.3	Arquitetura do SmartGlass	38
3.4	Diagrama de Atividade do SmartGlass	39
3.5	Aplicação do Gear VR na Realidade Aumentada (MATTERPORT, 2018)	40
3.6	Interface da Aplicação	41
3.7	<i>Handshake</i> no <i>WebSocket</i> (GONÇALVES; BASTOS; OLIVEIRA, 2014)	42
3.8	Arquitetura da Rede Siamesa (GUPTA, 2017)	45
3.9	Estrutura da FaceNet (Schroff; Kalenichenko; Philbin, 2015)	46
3.10	<i>Triplet Loss</i> na FaceNet (Schroff; Kalenichenko; Philbin, 2015)	47
3.11	Módulo de Treinamento do OpenFace	48
3.12	Fluxo para a detecção de face no OpenFace (AMOS; LUDWICZUK; SATYANARAYANAN, 2016)	49
3.13	Reconhecimento facial utilizando o <i>One-shot Learning</i>	49
3.14	Representação do Modelo Proposto	51

4.1	Estrutura da rede MTCNN (ZHANG et al., 2016)	55
4.2	Comparativo Cascade Classifier e Multi-task CNN	56

Capítulo 1

Introdução

O avanço dos computadores eletrônicos resultou em uma nova forma de interação para controlar processos e equipamentos, com essas melhorias ocasionou-se o aumento da demanda de conhecimentos para a utilização do sistema e a necessidade de treinamentos, visto que o conhecimento do mundo real já não era o bastante.

De acordo com Kirner, “apesar dos benefícios da tecnologia, a sofisticação das interfaces do usuário fez com que as pessoas tivessem que se ajustar às máquinas, durante muitas décadas”(KIRNER, 2007). Conforme a evolução das tecnologias de hardware e software, o conceito das interfaces para os usuários adaptou-se para que as máquinas se ajustassem para as pessoas, surgindo então interfaces hápticas, interfaces tangíveis, interfaces de voz, etc, proporcionando, aos usuários, uma interação do mundo real como forma de comunicação para um determinado sistema.

Nesse contexto, a Realidade Virtual (RV) surge então como uma nova geração de interface, tendo como objetivo de romper a barreira da tela e viabilizar interações mais naturais utilizando representações tridimensionais mais próximas da realidade do usuário. Embora a área tenha tido os primeiros experimentos na década de 50, foi apenas concretizada na década de 90, devido o avanço tecnológico proporcionando a execução da computação gráfica interativa em tempo real.

A necessidade de equipamentos especiais como capacete, luvas, etc, para a imersão do usuário com a aplicação, na época, foi um obstáculo para a popularização da RV, devido o

desconforto e as dificuldades de interação do usuário em um ambiente virtual desconhecido. Enquanto a RV necessita de equipamentos próprios, a Realidade Aumentada (AR) não apresenta esta restrição, podendo ser utilizada em qualquer ambiente (fechado ou aberto), sendo, portanto mais abrangente e universal (KIRNER, 2007).

A AR é uma variação de Ambientes Virtuais (AV), ou mais comumente chamado de RV. Dentro de um ambiente sintético, é possível ter uma completa imersão devido às tecnologias de RV. Enquanto imerso, o usuário não pode ver o mundo real ao seu redor. Em contraste, a AR permite ao usuário enxergar o mundo real, com objetos virtuais sobrepostos ou compostos com a realidade (AZUMA, 1997).

A capacidade dos objetos virtuais serem transportados para o espaço físico do usuário proporcionou interações tangíveis descomplicadas e naturais, sem a necessidade do uso de equipamentos especiais. Diante isso, Kirner considera que “a AR é uma possibilidade concreta de vir a ser a próxima geração de interface popular, a ser usada nas mais variadas aplicações em espaços internos e externos”.

Segundo Azuma, para o melhor funcionamento de um sistema de AR, são necessárias as três seguintes características: combinar real e virtual, alinhar objetos reais e virtuais entre si e a execução interativamente em tempo real (AZUMA, 1997). A importância dessas características são dadas para proporcionar uma melhor experiência ao utilizar um sistema de AR.

Diante do que foi exposto, esta proposta de trabalho de conclusão de curso considera o desenvolvimento de uma interface de AR inteligente, baseadas na utilização de técnicas de *Deep Learning*, para o Reconhecimento Facial de alunos da Universidade do Estado do Amazonas.

Ao longo desta introdução serão mostrados os demais elementos que compõem este trabalho. A Subseção 1.1 contempla os objetivos propostos para o desenvolvimento do projeto. Na Subseção 1.2 são apresentadas as justificativas que motivam a realização do trabalho em questão. A metodologia adotada é detalhada na Subseção 1.3. Por fim, a Subseção 1.4 dispõe a organização do restante do documento.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo geral consiste em desenvolver uma interface de Realidade Aumentada inteligente para o Reconhecimento Facial de alunos da Escola Superior de Tecnologia da Universidade do Estado do Amazonas (EST-UEA) através da utilização de Redes Neurais Convolucionais, de modo a auxiliar o controle da entrada de alunos na universidade.

1.1.2 Objetivos Específicos

Para alcançar o objetivo geral, alguns objetivos específicos precisam ser contemplados, a citar:

1. Formular um referencial teórico sobre os conceitos de interfaces para a Realidade Aumentada, *Deep Learning* e Redes Neurais Convolucionais;
2. Consolidar uma base de dados com exemplos realísticos para treinamento dos modelos;
3. Explorar a utilização de arquiteturas canônicas de Redes Neurais Convolucionais;
4. Treinar e testar as Redes Neurais Convolucionais propostas;
5. Avaliar os resultados obtidos;
6. Desenvolver a aplicação de Realidade Aumentada para a comunicação com o melhor modelo avaliado;

1.2 Justificativa

O Reconhecimento Facial (RF) pode ser importante em diversos aspectos. No contexto da Realidade Aumentada, uma interface inteligente pode auxiliar em uma melhor execução das três características impostas pelo Azuma (AZUMA, 1997), para o melhor funcionamento de um sistema de AR. Uma delas, em especial, a execução interativa em tempo real pode ser aperfeiçoada com a combinação de Redes Neurais Convolucionais, empregando técnicas de

Deep Learning para a classificação de imagens provendo uma identificação facial em tempo real para um sistema de AR.

Um outro aspecto que ressalta a importância da realização de um trabalho desta natureza é a alta aceitação do uso da face comparado às outras tecnologias biométricas. Segundo Silva, a face tem várias vantagens sobre outras técnicas de biometria: por ser natural, não exigir equipamentos sofisticados, a aquisição de dados é baseada em abordagens não invasivas, e pode ser feito a distância, de maneira cooperativa ou não (SILVA, 2018b). Portanto, o RF pode ser responsável pela segurança de um lugar, monitorando o controle de acesso para pessoas autorizadas, em particular, o auxílio para o controle da entrada de alunos da EST-UEA.

Por fim, a proposta do trabalho de conclusão de curso incentiva a prática de conceitos, métodos e tecnologias de duas áreas da Computação, a Realidade Aumentada e o *Machine Learning*, contribuindo na formação de um bacharel em Sistemas de Informação.

1.3 Metodologia

Para alcançar os objetivos propostos no escopo deste trabalho, a metodologia para condução das atividades é composta pelas seguintes atividades:

1. Estudo dos conceitos teóricos e de interfaces a respeito da Realidade Aumentada;
2. Estudo dos conceitos teóricos relacionados à Redes Neurais Artificiais, *Deep Learning*;
3. Modelar a Arquitetura do sistema para a comunicação do aplicativo com o melhor modelo avaliado;
4. Consolidar uma base de dados para fins de aprendizado das Redes Neurais Convolucionais;
5. Elencar arquiteturas canônicas das Redes Neurais Convolucionais apropriadas a proposta em questão;
6. Realizar os treinamentos das Redes Neurais Convolucionais propostas;
7. Realizar os testes das Redes Neurais Convolucionais propostas;

8. Desenvolver a aplicação de Realidade Aumentada para a comunicação com a Rede Neural Convolucional;
9. Analisar e comparar os resultados obtidos a fim de identificar o melhor modelo proposto;
10. Testar e ajustar aplicação;
11. Experimentar e conclusões a respeito do aplicativo e Rede Neural Convolucional proposta;
12. Escrita da proposta do Trabalho de Conclusão de Curso;
13. Defesa da proposta do Trabalho de Conclusão de Curso;
14. Escrita do Trabalho de Conclusão de Curso;
15. Defesa do Trabalho de Conclusão de Curso;

1.4 Organização do Documento

Para a apresentação desta proposta de trabalho de conclusão de curso, este documento está organizado como segue. A Seção 2 abrange uma fundamentação teórica sobre Realidade Aumentada, Redes Neurais Artificiais, *Deep Learning* e uma análise dos trabalhos relacionados. A Seção 3 contempla os detalhes da solução proposta para uma interface inteligente de AR. Por fim, as considerações parciais do trabalho encontram-se na Seção 4.

Capítulo 2

Fundamentação Teórica

A fundamentação teórica para a realização deste trabalho compreende conceitos ligados à Realidade Aumentada e ao *Machine Learning*. Quanto ao primeiro tópico, os conceitos de Realidade Aumentada serão apresentados na Subseção 2.1, bem como suas interfaces e aplicabilidades. Quanto ao segundo tópico, a Subseção 2.2 compreende os principais conceitos, métodos de aprendizagem e algumas aplicações de Redes Neurais Artificiais. Em seguida, será apresentado os conceitos sobre *Deep Learning* na Subseção 2.3. Por fim, uma análise dos trabalhos relacionados na Subseção 2.4.

2.1 Realidade Aumentada

Jordan Herrema define a Realidade Aumentada (AR) como uma tecnologia de Interação Humano-Computador (IHC) na qual o usuário percebe simultaneamente o mundo real e os objetos virtuais. O objetivo é integrar os objetos virtuais - como modelos 3D, textos, imagens, vídeos por exemplo - sobrepostos o mais perfeitamente possível no espaço do mundo real, criando a ilusão de que todos os objetos coexistem no mesmo espaço (HERREMA, 2013).

2.1.1 Conceitos e Definições

Ronald Azuma, pesquisador da Universidade da Carolina do Norte, descreve a AR como um sistema que acrescenta ao mundo real os objetos virtuais gerados por um computador, aparentando

a coabitação no mesmo espaço (Azuma et al., 2001) e apresenta as seguintes particularidades definidas por Azuma. Mark Billinghurst as descreve da seguinte forma (BILLINGHURST, 2004):

- Combinação de objetos reais e virtuais no ambiente real: a AR requer tecnologias *display* que permitem o usuário enxergar simultaneamente informações virtuais e reais em uma visão combinada;
- Execução da interatividade em tempo real: O sistema de AR deve ser executado em taxas de quadros interativos, de modo que possa sobrepor informações em tempo real e permitir a interação do usuário;
- Nivelação de objetos reais e virtuais entre si: a AR confia em um acoplamento íntimo entre o virtual e o real que é baseado em seu relacionamento geométrico. Isso torna possível renderizar o conteúdo virtual com o posicionamento correto e a perspectiva 3D em relação ao real;
- Aplicabilidade a todos os sentidos, abrangendo o tato, audição, etc: a definição de AR não se limita ao sentido visual, pode potencialmente se aplicar a todos os sentidos, incluindo toque, audição, etc e pode ser usado para aumentar ou substituir os sentidos perdidos dos usuários através de substituição sensorial.

Em contrapartida, Alan Craig define o processo de duas etapas que são necessárias em uma aplicação de AR (CRAIG, 2013). As duas etapas são:

1. A aplicação precisa determinar o estado atual do mundo real e do mundo virtual;
2. A aplicação precisa exibir o mundo virtual inserido no mundo real de uma forma na qual os usuários detectem os elementos do mundo virtual como parte de seu mundo real e depois retornar para a etapa 1 passando pra próxima etapa de tempo.

A AR está posta em um contexto mais amplo, denominado Realidade Misturada (RM). De acordo com Kirner e Tori, “a RM pode ser definida como a sobreposição de objetos virtuais

tridimensionais gerados por computador com o ambiente físico, mostrada ao usuário, com o apoio de algum dispositivo tecnológico, em tempo real”(KIRNER; TORI, 2006).

Um sistema de RM possui o objetivo de produzir um ambiente realista capaz de dificultar a identificação de diferenças entre os elementos reais e os virtuais envolvidos na cena, considerando-os como uma coisa só.

Em 1994, Paul Milgram levantou uma questão importante sobre qual seria a relação entre AR e Realidade Virtual (RV). Milgram concorda que é válido relacionar os dois conceitos juntos, o ambiente de RV é aquele que o usuário está totalmente imerso em um mundo completamente artificial, que pode ou não imitar as propriedades de um ambiente do mundo real, seja existente ou fictício, mas que também possa exceder os limites da realidade criando um mundo onde as leis da física não regem mais. Em contraste, um ambiente do mundo real claramente deve ser regido pelas leis da física. O autor aborda que ao invés de considerar os dois conceitos como antíteses, é mais conveniente vê-los situados em extremidades opostas de um contínuo, que refere-se como Contínuo Realidade-Virtualidade (CRV) (MILGRAM et al., 1994). Este conceito é ilustrado na Figura 2.1.

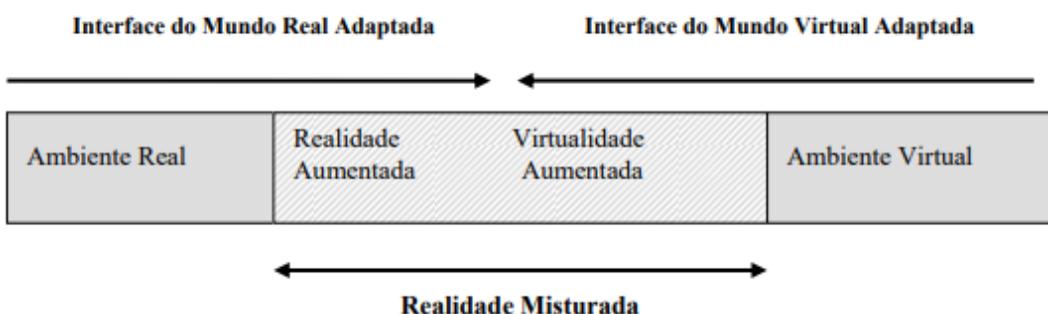


Figura 2.1: Representação Simplificada de Contínuo Realidade-Virtualidade (adaptada de (MILGRAM et al., 1994)) (KIRNER; TORI, 2006)

Os dois extremos, representados na Figura 2.1, correspondem à esquerda o meio físico real e à direita um meio completamente virtual interpretado pela RV. Neste contexto, a definição para um ambiente genérico de RM é quando os objetos do mundo real e mundo virtual são apresentados em uma única tela, logo, em qualquer lugar entre os extremos do CRV.

2.1.2 Interfaces

Segundo Julie Carmignani, um dos aspectos mais importantes da AR é a criação de técnicas apropriadas para interações intuitivas entre o usuário e o conteúdo virtual de uma aplicação de AR. Existem quatro formas principais de interação numa aplicação de AR: interfaces tangíveis, interfaces colaborativas, interfaces híbridas e as interfaces multimodais emergentes (CARMIGNANI; FURHT, 2011).

Interfaces Tangíveis

Conforme Brygg Ullmer e Hiroshi Ishii, a Interface de Usuário Tangível (IUT) são representações físicas para informações digitais, suportando interações diretas com o mundo real, explorando o uso de objetos e ferramentas físicas. Apesar de mouses e teclados ser objetos físicos, a localização e as formas dos objetos de IUT são relevantes para o mundo virtual (ULLMER; ISHII, 2000).

Um exemplo clássico de IUT para AR é a aplicação desenvolvida por Hirokazu Kato, consiste na seleção e reorganização dos móveis em uma sala de estar virtual utilizando uma pequena raquete. Empurrar, inclinar, golpear são movimentos mapeados para desempenhar comandos intuitivos baseados em gestos, a fim de proporcionar ao usuário uma experiência intuitiva (KATO et al., 2000).

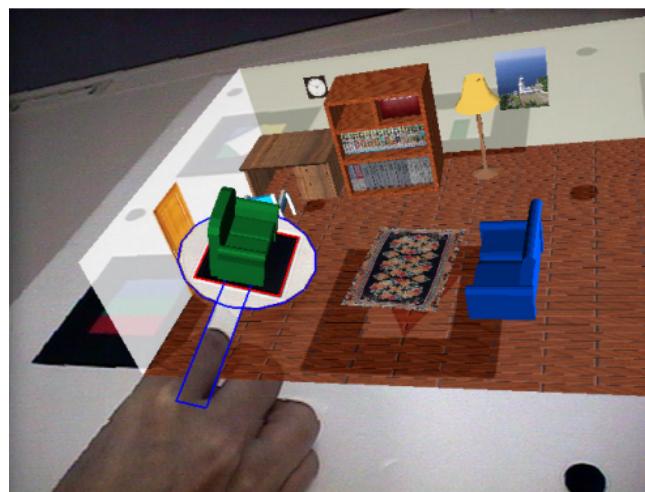


Figura 2.2: Usuário utiliza a raquete para pegar, mover, soltar ou destruir modelos (Azuma et al., 2001)

Um exemplo mais recente de IUT para AR é o AnimAR (LOPES; REITER; REIS, 2018). AnimAR é um aplicativo para criação de cenários animados, combinando AR e IUT, foi possível criar um novo nível de interação do usuário com a ferramenta, facilitando a criação de cenários e animações.

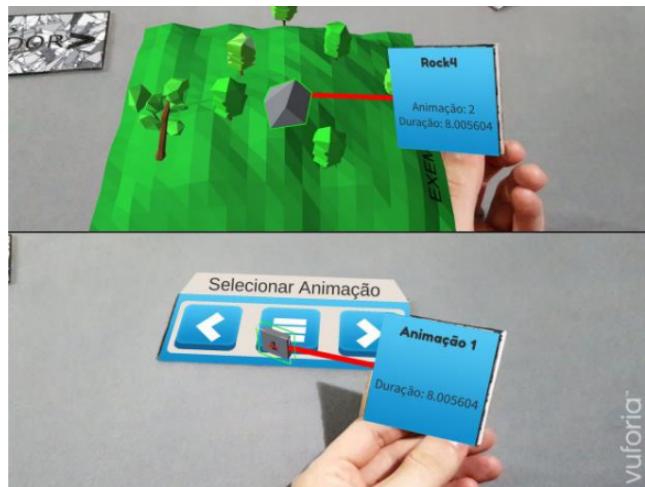


Figura 2.3: Aplicação AnimAR (LOPES; REITER; REIS, 2018)

Interfaces Colaborativas

As interfaces colaborativas de AR consiste na utilização de múltiplos *displays* para sustentar as atividades co-localizadas e remotas. O compartilhamento co-localizado utiliza interfaces 3D para aprimorar a visualização, discussão e interação a respeito de um modelo virtual 3D. No compartilhamento remoto, a AR é voltada para a integração de múltiplos dispositivos de localizações diversas para, por exemplo, aprimorar as teleconferências (BARAKONYI; FAHMY; SCHMALSTIEG, 2004).

Em 1998, Henry Fuchs apresentou a implementação de um sistema colaborativo de AR tendo uma visualização tridimensional voltada para o auxílio de cirurgias laparoscópicas, a fim de reduzir o tempo dos procedimentos, encurtar o tempo de treinamento de médicos para a aprendizagem das técnicas cirúrgicas, aumentar a precisão nos procedimentos devido a melhor compreensão das estruturas em questão e a melhor coordenação *hand-eye*, reduzir o trauma do paciente através de cirurgias mais curtas e precisas, e aumentar a disponibilidade de cirurgias devido a facilidade de realizá-las (FUCHS et al., 1998).

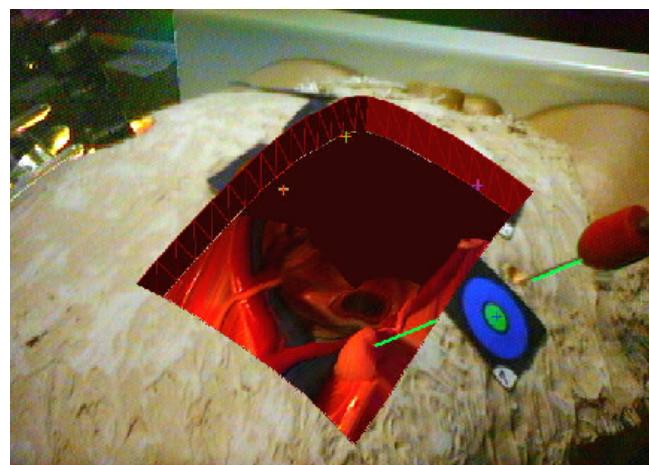


Figura 2.4: Visualização de Realidade Aumentada para Cirurgia Laparoscópica(FUCHS et al., 1998)

Interfaces Híbridas

As interfaces híbridas de AR são uma combinação de diferentes, porém complementares interfaces. De acordo com Zhou, “para algumas aplicações de AR, a visão computacional sozinha não pode fornecer uma solução de rastreamento robusta e, portanto, métodos híbridos foram desenvolvidos, que combinam várias tecnologias de detecção”(ZHOU; DUH; BILLINGHURST, 2008) .

Por exemplo, Ribo et al. propuseram o desenvolvimento de um sistema de AR totalmente móvel e vestível que combina um rastreador baseado na visão computacional, com um rastreador inercial (hardware especializados) baseados em acelerômetros e giroscópios (RIBO et al., 2002).

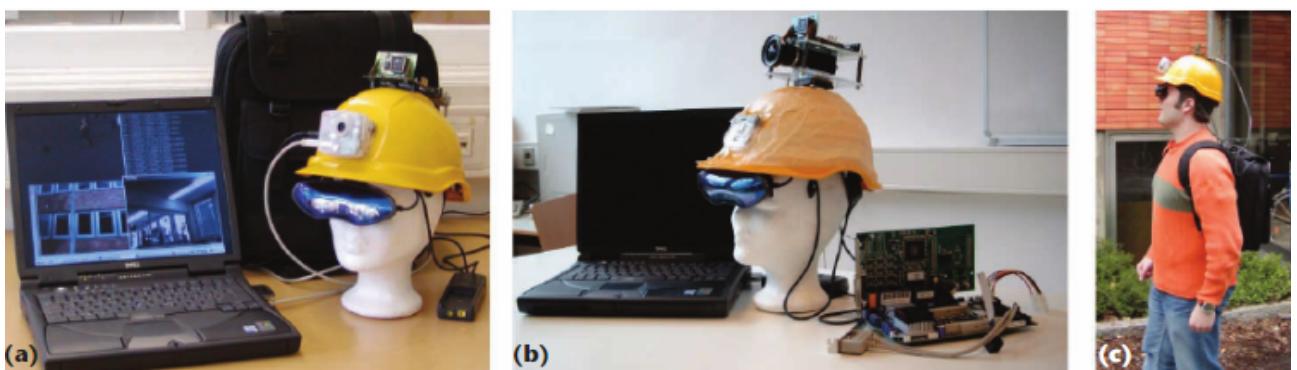


Figura 2.5: O kit Móvel de AR: (a) o sistema de processador único para alvos simples, (b) o sistema de processador duplo para cenários mais complexos; e (c) o kit de AR em uso. (RIBO et al., 2002)

Interfaces Multimodais Emergentes

De acordo com Carmignani, as interfaces multimodais de AR combinam a entrada de objetos reais com formas de linguagem e comportamentos naturais como fala, tato, gestos naturais das mãos ou olhar (CARMIGNANI; FURHT, 2011).

Um exemplo de interface multimodal é o sistema de AR desenvolvido por Lee et al. que utiliza as informações do olhar e piscar do usuário para a interação com os objetos. O amplo desenvolvimento deste tipo de interação é justificado por oferecer uma forma relativamente robusta, eficiente, expressiva e altamente portátil, representando o estilo preferido de interação dos usuários (LEE et al., 2011).

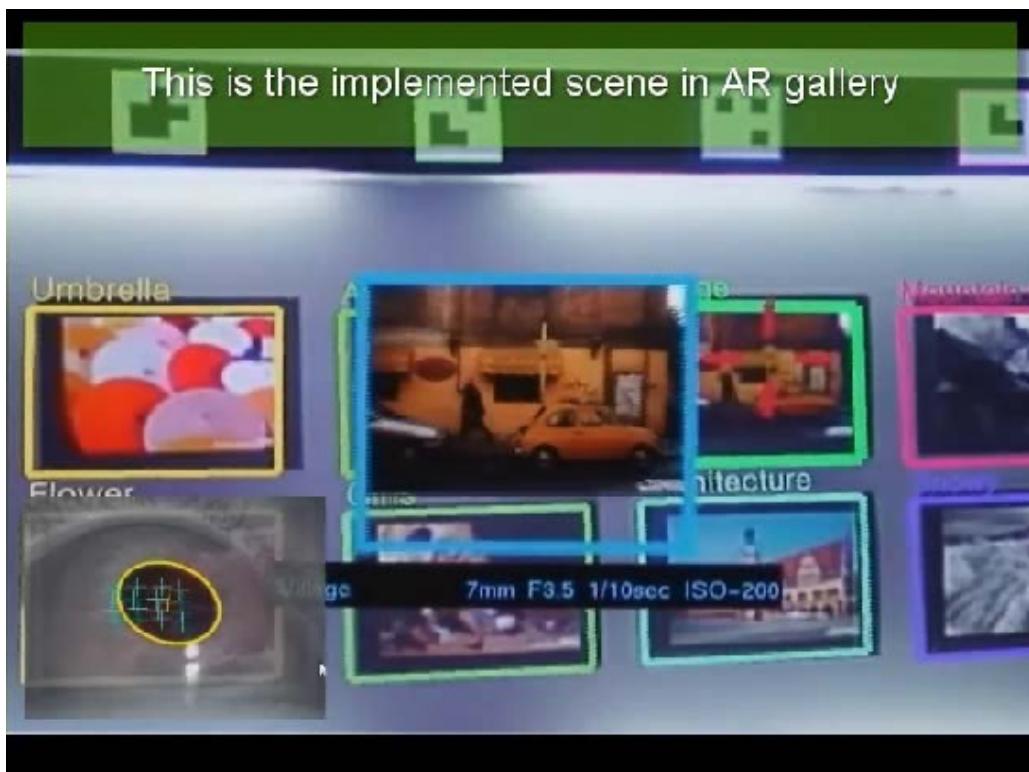


Figura 2.6: Informação aumentada para os objetos específicos nas fotos (LEE et al., 2011)

2.1.3 Realidade Aumentada Contemporânea

O avanço da computação portátil viabilizou a criação de um subconjunto da AR: a realidade aumentada móvel. Segundo Zhou, a AR móvel tornou-se uma boa alternativa aos sistemas HMD (*Head Mounted Display*), especialmente por ser minimamente invasiva, socialmente aceita, pron-

tamente disponível e altamente portátil. Atualmente, os dispositivos portáteis disponíveis para um sistema de AR móvel são: *smartphones* e *tablets* (ZHOU; DUH; BILLINGHURST, 2008).

Devido ao aperfeiçoamento dos dispositivos computacionais, foi possível desenvolver diversas tecnologias que podem ser envolvidas em um sistema de AR. Giandomenico Caruso, pesquisador da Universidade Politecnico di Milano em Milão, reuniu algumas dessas tecnologias e ilustrou na Figura 2.7 (CARUSO, 2017).

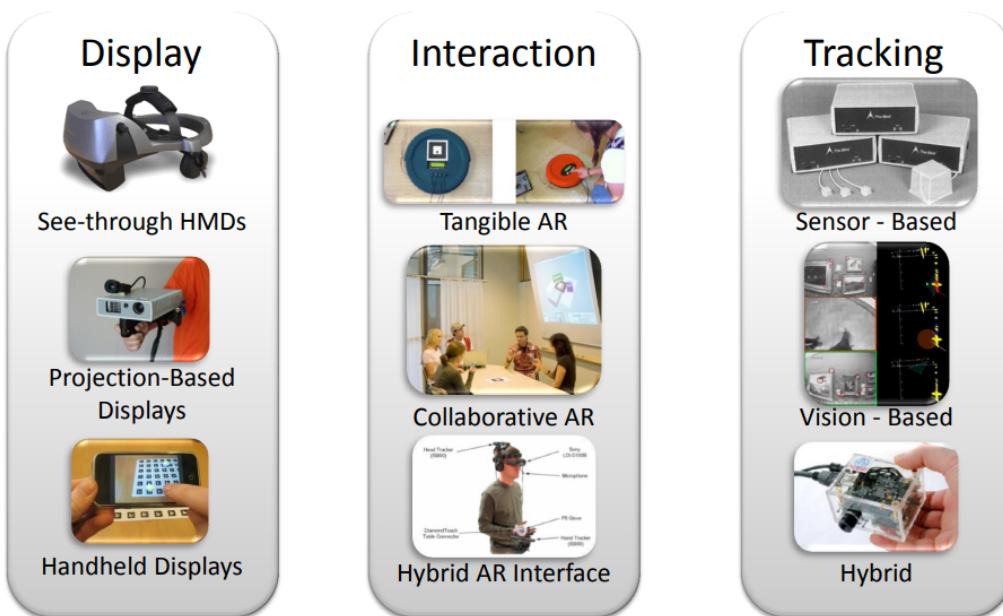


Figura 2.7: Tecnologias de Realidade Aumentada (CARUSO, 2017)

Schmalstieg e Höllerer em seu livro (“Augmented Reality: Principles and Practice”) listam diversos componentes e elementos utilizados para implementações de AR (Schmalstieg; Höllerer, 2017). Cada área, representadas na Figura 2.7, gerou inovações e tecnologias relevantes, Boquimpani e Figueira Filho as exemplifica da seguinte forma (BOQUIMPANI; FILHO, 2017):

- **Display:** baseado no melhor entendimento dos fundamentos da percepção visual humana, diversos tipos de *displays* como HMDs, *projective displays* e *handheld displays* evoluíram suas capacidades (Schmalstieg; Höllerer, 2017);
- **Interação:** em uma aplicação de AR várias técnicas e estilos de interação são relevantes, podem ir desde a interação situacional (baseada em elemenetos da interface do usuário), na qual retorna informações simples até a interação tridimensional completa.

- **Rastreamento:** a evolução dos sensores e as câmeras contribuíram para a fácil criação de novas técnicas e dispositivos de AR. Os sensores passaram de estacionários a móveis e novos métodos começaram a ser utilizados, como o rastreamento óptico, do qual se beneficiou do imenso avanço das pesquisas de algoritmos destinados a Visão Computacional. As câmeras tornaram-se partes integrantes da maioria dos dispositivos computacionais, contribuindo para o alinhamento geométrico das partes físicas e virtuais que compõem a experiência da AR (Schmalstieg; Höllerer, 2017).

Kunkel et al. em 2016, descreveram os principais impactos do uso da AR no âmbito profissional e destacam que “a realidade aumentada e virtual ajudam a acelerar a coalescência dos usuários em volta de sua experiência do mundo com os dispositivos, melhorando a fidelidade da intenção, aumentando a eficiência e impulsionando a inovação”(KUNKEL et al., 2016).

Figure 1. The evolution of interaction

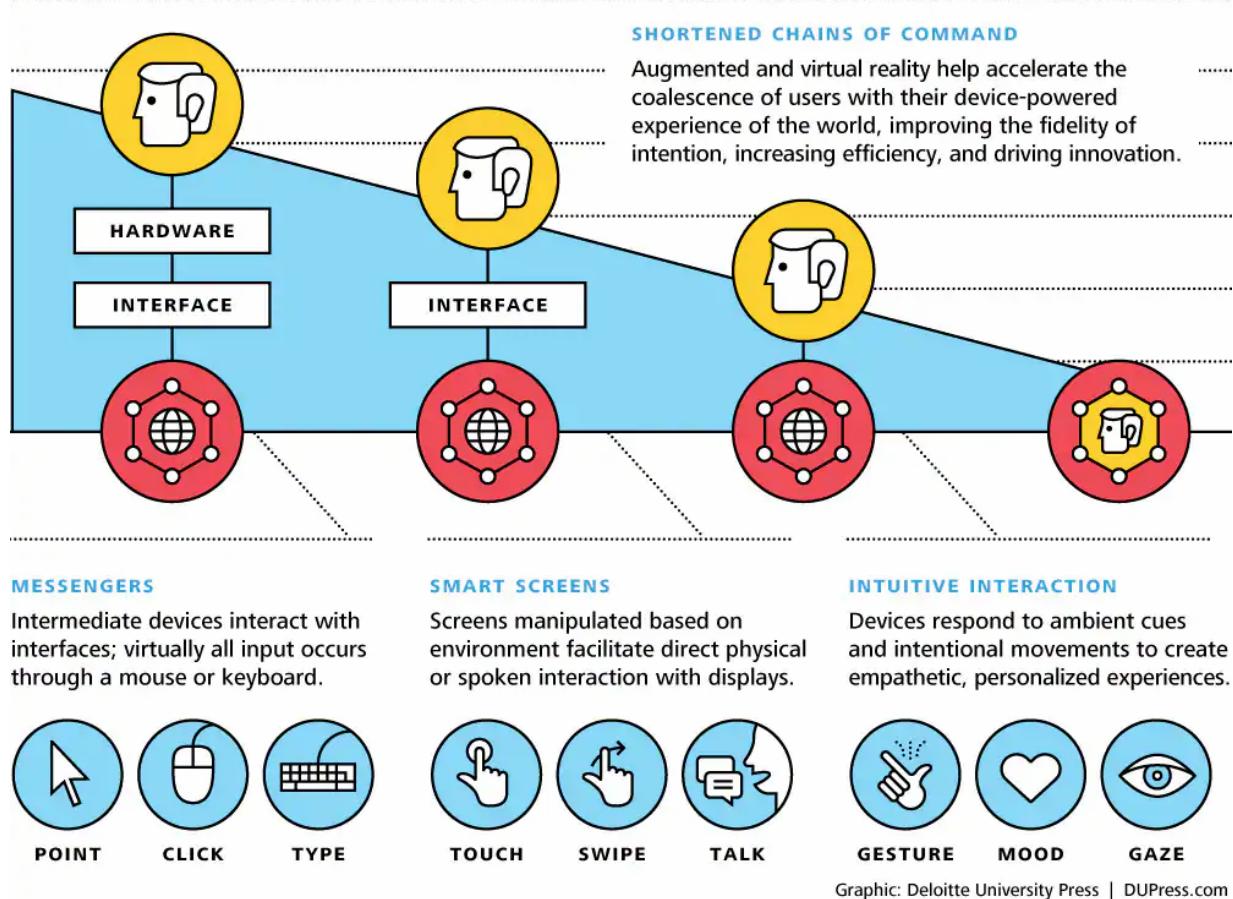


Figura 2.8: A evolução da interação (KUNKEL et al., 2016)

Segundo Kunkel et al., devido a criação de uma experiência personalizada que responde a sinais do ambiente e a movimentos intencionais, as aplicações de AR podem causar sensações de alívio diante o enorme trabalho de cognição das pessoas ao tentar decodificar as complexas quantidades de informações presentes no mundo físico, ocasionado uma diminuição das barreiras de uso dos aplicativos (KUNKEL et al., 2016).

Empresas e investidores referências no setor tecnológico, apontam que o mercado de AR se apresenta promissor para os próximos anos e para alguns analistas de mercado é considerada como a quarta nova tendência em computação para os consumidores, depois de PCs, internet e *smartphones*.

Segundo o portal de notícias Newtrade, baseado nos dados disponibilizados pela consultoria Digi-Capital, “os investimentos em startups da área já ultrapassaram US\$ 2 bilhões nos últimos dois anos. Quanto mais a tecnologia é desenvolvida e implantada, maior é o valor do mercado de AR”. E a International Data Corporationm - empresa líder em inteligência de mercado e consultoria nas indústrias de tecnologia da informação - realizam as seguintes estimativas baseadas nos valores do mercado de anos anteriores (NEWTRADE, 2017):

- US\$ 570 milhões: valor do mercado de AR em 2015;
- US\$ 5,2 bilhões: valor do mercado de AR em 2016;
- US\$ 162 bilhões: valor do mercado de AR previsto para 2020.

As áreas de maiores aplicações e crescimento previstos para o mercado de RV e AR são as industriais e automotivas. O emprego dessas tecnologias suprirá as dificuldades de montagem e manutenção de máquinas complexas. Segundo uma pesquisa realizada pelo Goldman Sachs, “O mercado de realidade aumentada automotiva é estimado para ter crescimento com um CAGR (*Compound Annual Growth Rate* - Taxa de Crescimento Anual Composta) de 80% de 2016 para 2024”. Em relação à indústria automotiva, a AR ajudará a reduzir distrações dos motoristas e aumentar a segurança. A educação é outro segmento que tem um prometido crescimento para os próximos anos, tanto na área de RV quanto com a AR. Em 2025, US\$ 700 milhões serão investidos na tecnologia de AR na educação (BELLINI et al., 2016).

A Google foi uma das grandes empresas pioneiras a investir na criação de produtos para a área de AR. O Google Glass, um dispositivo *wearable* criado pela Google, que por meio de uma pequena tela propunhava a visualização de mapas, opções de música, previsão do tempo, rotas de mapas, e além disso, efetuar chamadas de vídeo ou tirar fotos e compartilhar imediatamente através da internet, entregando assim uma experiência de AR.



Figura 2.9: Google Glass (HOOKEY, 2015)

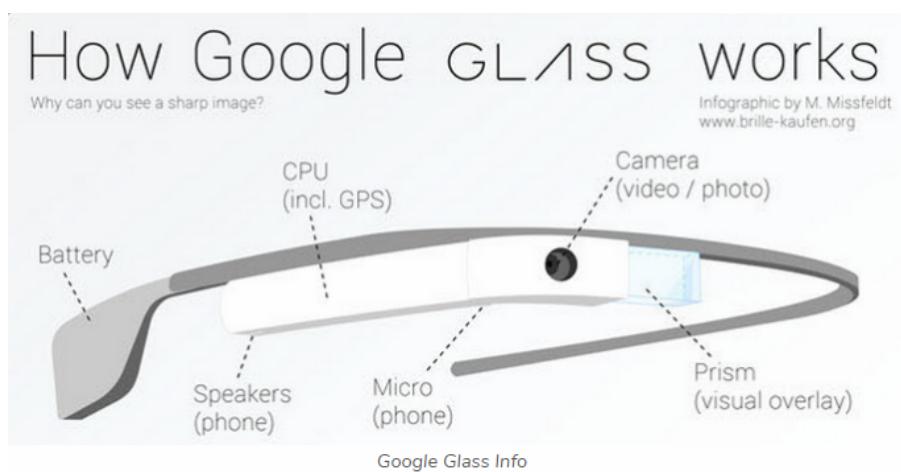


Figura 2.10: Como o Google Glass funciona (MISSFELDT, 2017)

O dispositivo mais recente que integra as mais recentes evoluções tecnológicas tanto de hardware como de software para proporcionar uma experiência de AR é o HoloLens 2, criado pela Microsoft. De acordo com a Microsoft, “a RM no HoloLens 2 combina um dispositivo livre

com aplicativos e soluções que ajudam as pessoas em todo a empresa a aprender, comunicar-se e colaborar de maneira mais eficaz. Ela é o resultado de avanços no design de hardware, inteligência artificial (IA) e desenvolvimento de realidade misturada da Microsoft, criada para ajudar a levar as indústrias para o futuro a partir de hoje”(MICROSOFT, 2019).



Figura 2.11: Crie espaços e mostre-os a outras pessoas em escala real (MICROSOFT, 2019)

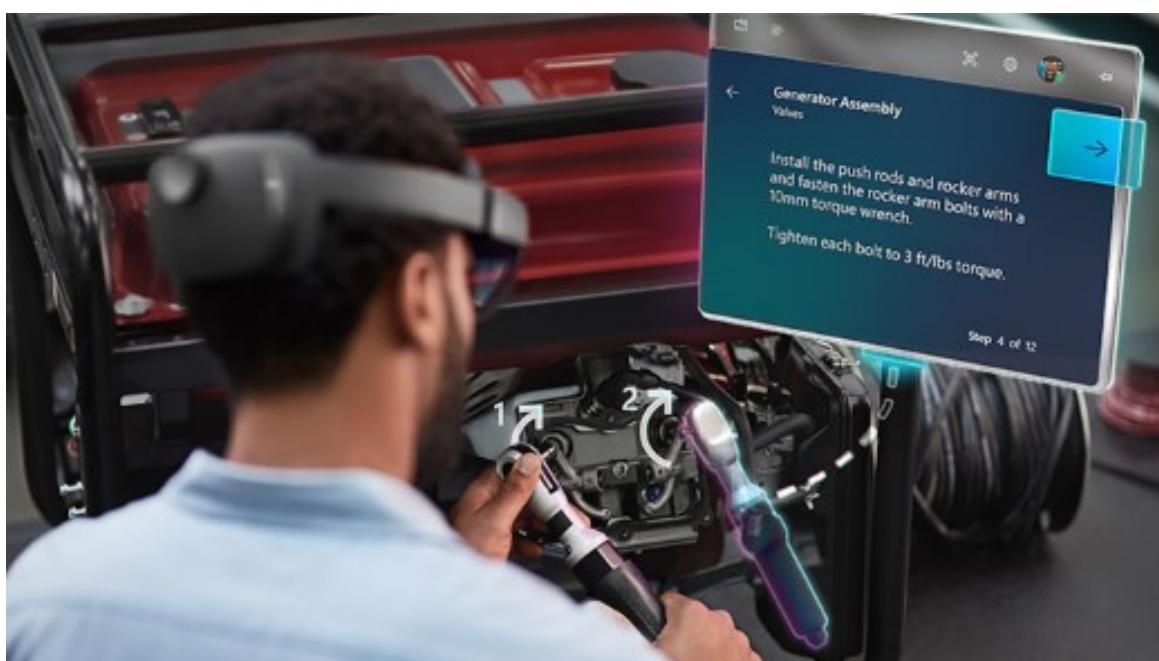


Figura 2.12: Ajude os funcionários a aprender com instruções passo a passo (MICROSOFT, 2019)

2.2 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) são modelos computacionais inspirados na estrutura neural do cérebro humano, isto é, procuram adquirir seus conhecimentos através da experiência (HAYKIN, 2001). Segundo Braga, Carvalho e Ludermir, as RNAs são sistemas paralelamente distribuídos compostos por unidades de processamentos simples, os neurônios artificiais, que calculam determinadas funções matemáticas (normalmente não-lineares) para produzirem resultados (BRAGA; CARVALHO; LUDEMRIR, 2007).

A composição da RNA é feita por um conjunto de neurônios artificiais conectados entre si, sendo para cada ligação entre neurônios, são vinculados pesos que retratam as influências que determinado dado possui dentro da RNA e os ajustes dos pesos está diretamente relacionado ao aprendizado das RNAs.

A capacidade de aprender padrões complexos a partir de dados e de predizer resultados coerentes para dados não conhecidos, demonstra a capacidade de generalização das informações aprendidas de uma RNA, tornando-as um atrativo principal para solução de problemas através de RNAs (BRAGA; CARVALHO; LUDEMRIR, 2007).

2.2.1 Neurônio Articial

De acordo com Haykin, o neurônio articial é a unidade de processamento de informação básica para o funcionamento de uma RNA, que por sua vez consiste em camadas interligadas entre esses elementos processadores (HAYKIN, 2001).

O primeiro modelo que simulava as características de um neurônio biológico foi desenvolvido por McCulloch e Pitts em 1943, conhecido como neurônio MCP (MCCULLOCH; PITTS, 1943). Nesse modelo, cada neurônio pode ser implementado conforme a Figura 2.13.

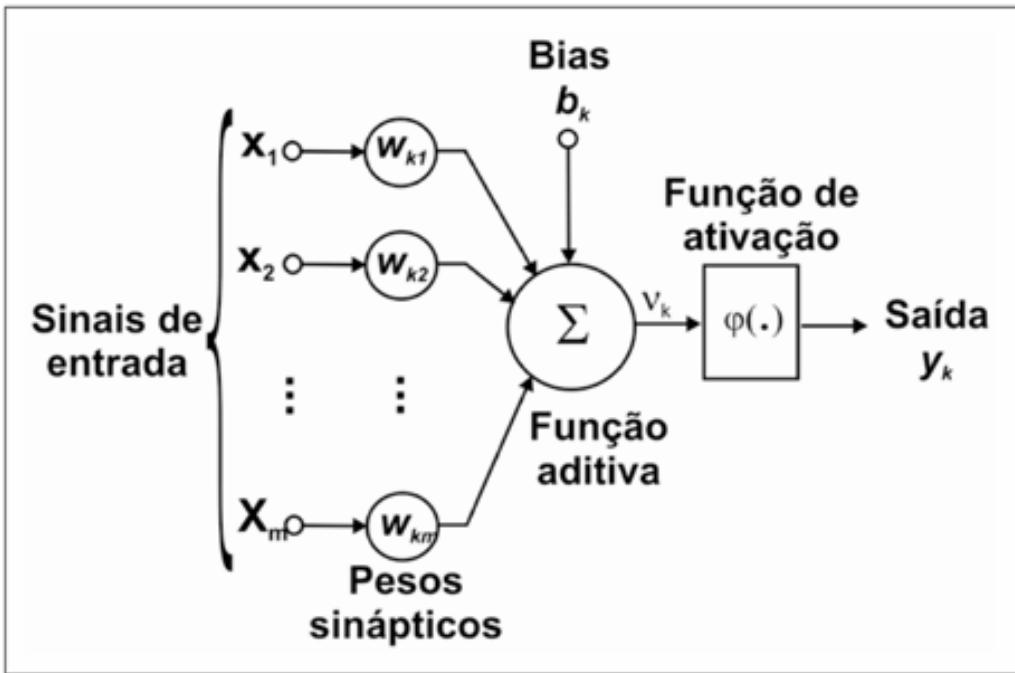


Figura 2.13: Modelo de neurônio artificial (MARTINS-FILHO; MOL; ROCHA, 2005)

O conjunto $\{x_1 + x_2 + \dots + x_n\}$ representa os múltiplos sinais de entrada da rede, interpretando os dentritos de um neurônio biológico. Os pesos que existem nas junções sinápticas de uma rede são implementadas no neurônio artificial como um conjunto de pesos sinápticos $\{w_{k1} + w_{k2} + \dots + w_{kn}\}$. O pesos refletem a importância de cada entrada $\{x_i\}$. Matematicamente, o peso sináptico de cada dendrito artificial é multiplicado pelo sinal de entrada. Os sinais ponderados da entrada são então somados ao bias ou limiar de ativação b_k , variável usada para especificar o limiar específico que a saída v_k ativará a saída através da função de ativação, que ficará responsável por determinar a saída y_k (SILVA, 2018a). No caso do MCP, a função de ativação é do tipo degrau deslocada do limiar de ativação b_k em relação a origem, o modelo mostrado na Figura 2.13 é definido por

$$v_k = \sum_{i=1}^n w_{ki} x_i - b_k \quad (2.1)$$

$$\phi(v_k) = \begin{cases} 1, & v_k \geq 0 \\ 0, & v_k < 0 \end{cases} \quad (2.2)$$

$$y_k = \phi(v_k) \quad (2.3)$$

As funções de ativação são significativamente importantes nas RNAs, são responsáveis em decidir se um neurônio deve ser ativado ou não, isto é, definem o grau de relevância da informação que o neurônio está recebendo para então ser enviada para a camada de saída. Segundo (REIS, 2016), as principais representantes são as indicadas abaixo:

- **Linear**

É a função de ativação mais básica devido a não alteração do valor de saída de um neurônio. Normalmente utilizada nas camadas de saída em redes neurais de regressão. Representada na Equação 2.4 e Figura 2.14.

$$\phi(x) = x \quad (2.4)$$

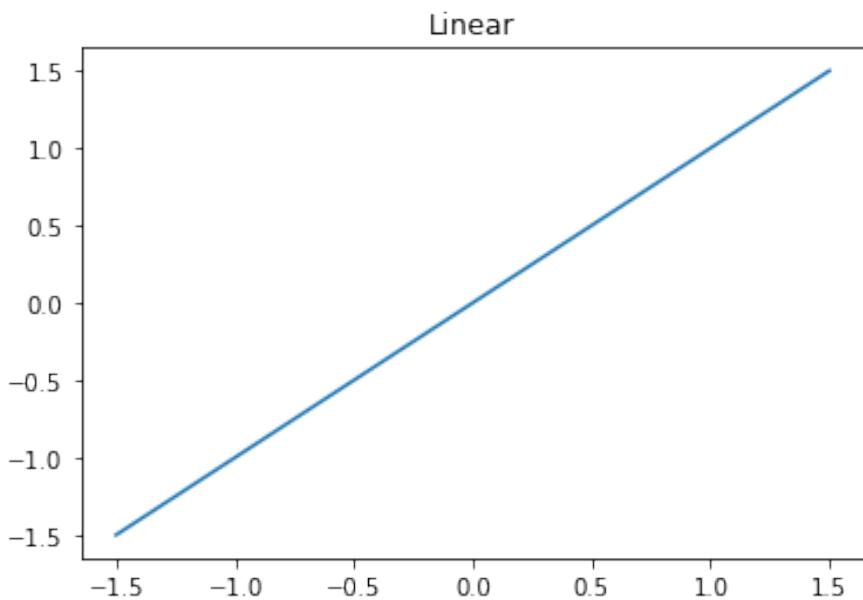


Figura 2.14: Função de ativação linear

- **Degrau**

É uma simples função de ativação que define a saída de 1 ou 0, de acordo com um limite estabelecido. Representada na Equação 2.5 e Figura 2.15.

$$\phi(x) = \begin{cases} 1, & x \geq 0.5 \\ 0, & x < 0.5 \end{cases} \quad (2.5)$$

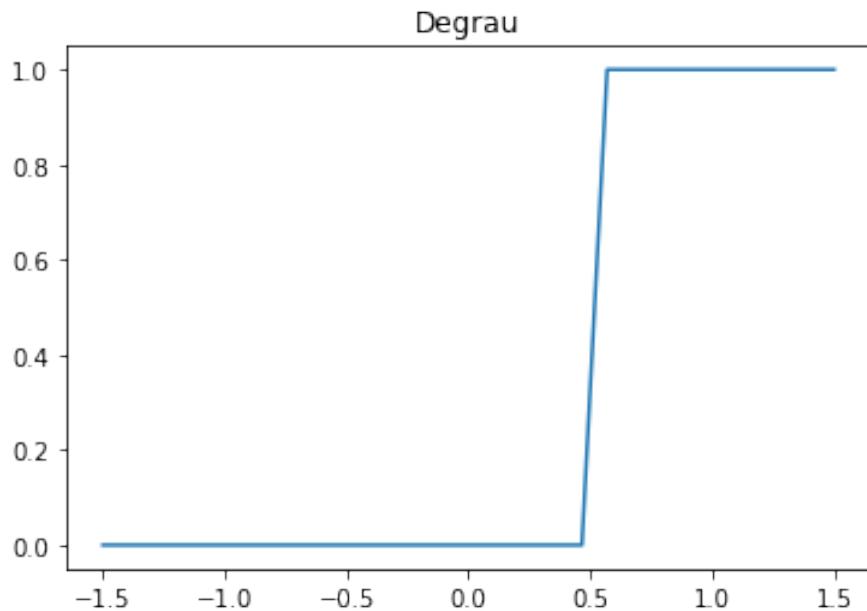


Figura 2.15: Função de ativação degrau

- **Sigmoidal**

A função de ativação sigmoidal é normalmente utilizada por RNAs com propagação positiva (*Feedforward*), que necessitam ter como saída apenas números positivos em redes neurais com múltiplas camadas. Representada na Equação 2.6 e Figura 2.16.

$$\phi(x) = \frac{1}{1 + e^{-x}} \quad (2.6)$$

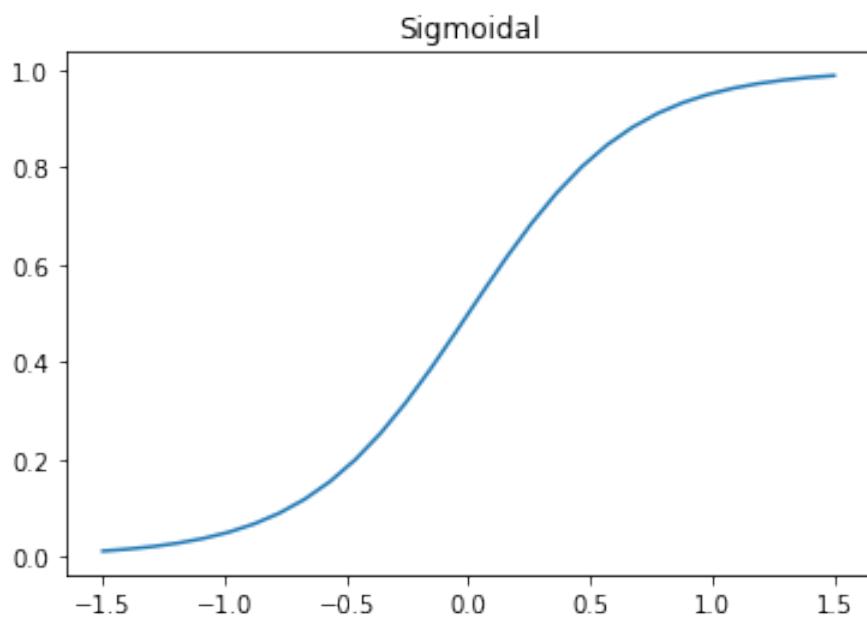


Figura 2.16: Função de ativação sigmoidal

- **Tangente Hiperbólica**

A função de ativação tangente hiperbólica possui uso muito comum em RNAs cujas saídas devem ser entre -1 e 1. Representada na Equação 2.7 e Figura 2.17.

$$\phi(x) = \tanh(x) \quad (2.7)$$

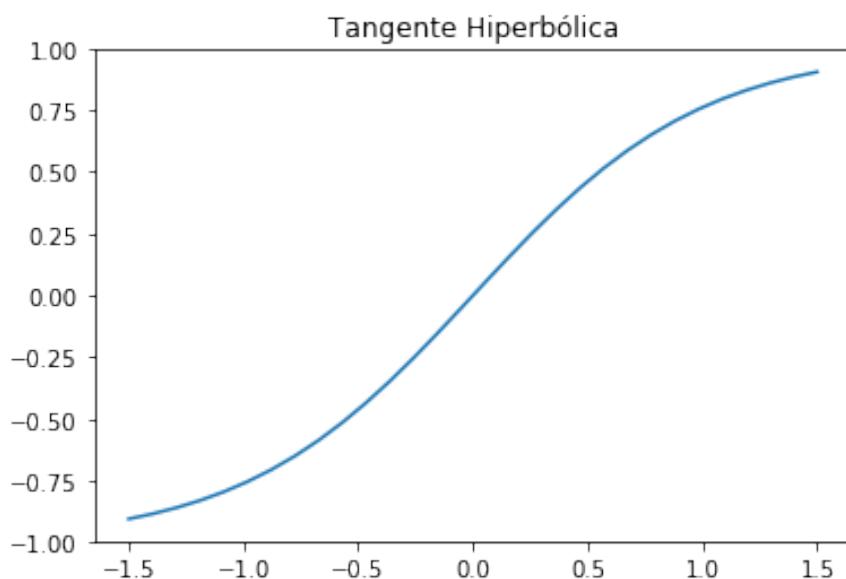


Figura 2.17: Função de ativação tangente hiperbólica

- **Unidade Linear Retificada (ReLU)**

A função de ativação ReLU é amplamente utilizada ao projetar RNAs atualmente, a principal vantagem sobre as outras funções de ativação é que ela não ativa todos os neurônios ao mesmo tempo, ou seja, se a entrada for negativa, o valor será convertido para zero e o neurônio não ativará, tornando a rede eficiente e fácil para a computação. Representada na Equação 2.8 e Figura 2.18.

$$\phi(x) = \max(0, x) \quad (2.8)$$

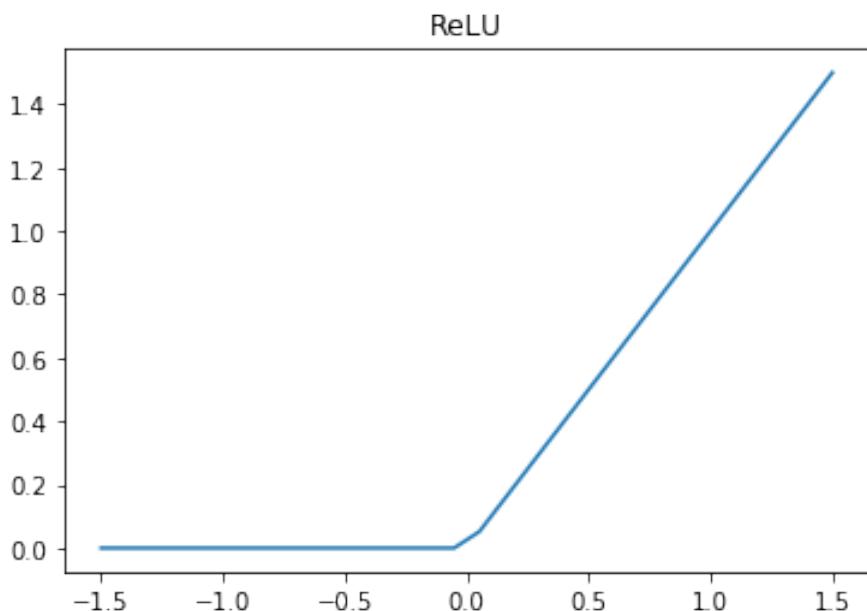


Figura 2.18: Função de ativação ReLU

Apesar do modelo MCP detalhar as formas de como o comportamento e a memória de neurônios biológicos podem ser representados de forma artificial, não é apresentado nenhuma forma de realizar a fase de aprendizagem em uma rede.

Avanços significativos surgiram com os trabalhos sobre aprendizagem de Donald Hebb, demonstrando que a aprendizagem da rede pode ser realizada através da inclusão de pesos em cada uma das conexões entre os neurônios e de Frank Rosenblatt com a publicação do modelo *perceptron* (HEBB, 1949; ROSENBLATT, 1958).

A topologia original do *perceptron* era composta por três camadas: retina, unidades sensoras capazes de receber informações do mundo externo; associação, formada por neurônios MCP com pesos fixos, definidos antes do período de treinamento; resposta, consistindo de apenas um neurônio MCP com propriedades adaptivas responsável por disponibilizar o processamento produzido pela rede para o mundo externo (BRAGA; CARVALHO; LUDELMIR, 2007). A Figura 2.19 mostra um esboço da topologia do *perceptron*.

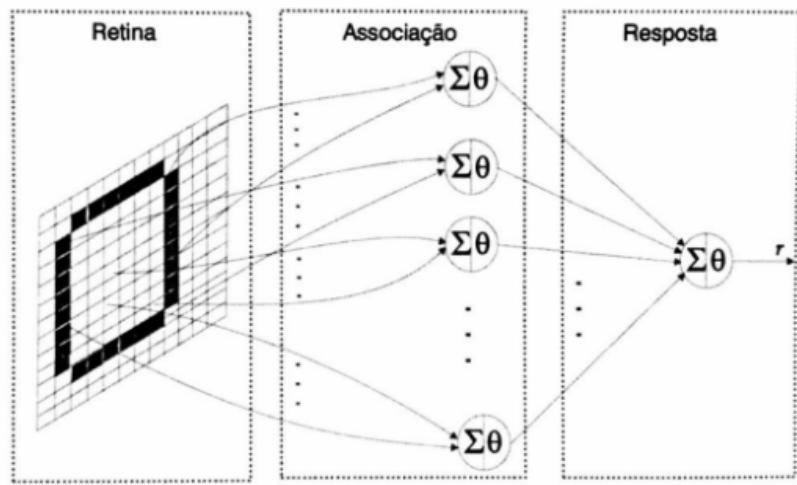


Figura 2.19: Perceptron de camada única (BRAGA; CARVALHO; LUDELMIR, 2007)

O modelo *perceptron* apresentava limitações ao ser utilizado para classificar padrões, cuja solução somente poderá ser encontrada em casos no qual a resolução do problema é linearmente separável. Conforme a Figura 2.20, um problema é considerado linearmente separável quando a solução é obtida através de uma reta dividindo o espaço em duas partes.

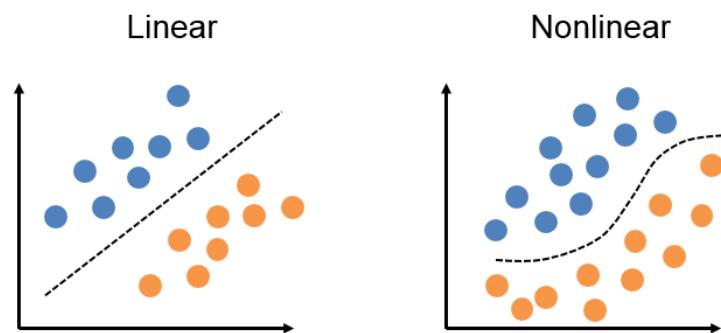


Figura 2.20: Função linearmente separável e Função não linearmente separável (BOOK, 2016)

Devido à limitação do *perceptron* de camada única, Minsky e Papert criticam a sua capacidade computacional, atentando para problemas onde uma reta não consegue solucionar a separação de duas ou mais classes, ou seja, problemas não linearmente separáveis. O modelo proposto pelos autores foi a inserção de múltiplas camadas intermediárias, denominado de *Multilayer Perceptron* (MINSKY; PAPERT, 1969).

2.2.2 Multilayer Perceptron

O *Multilayer Perceptron* (MLP) é uma RNA composta de uma camada de entrada, uma ou mais camadas ocultas para solucionar problemas mais complexos e uma camada de saída. As camadas ocultas são formadas por neurônios conectados entre si através de pesos, que são ajustados durante a fase de treinamento, para maximizar a sua assertividade para a classificação de padrões (Figura 2.21).

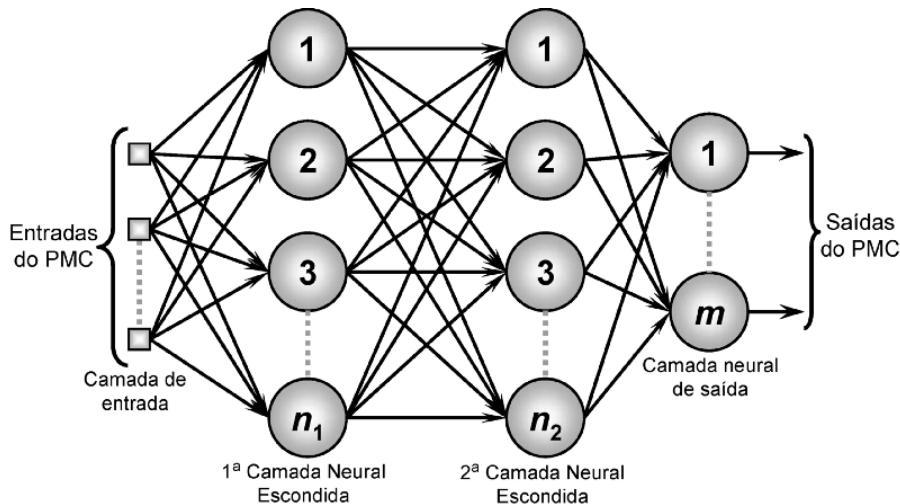


Figura 2.21: Representação de uma RNA *Multilayer Perceptron* (MOREIRA, 2018)

Uma característica importante das RNAs é a sua capacidade de aprender através de exemplos e fazer interpolações e extrapolações do que aprenderam. Conforme Braga, Carvalho e Ludermir, “a etapa de aprendizagem consiste em um processo iterativo de ajuste de parâmetros da rede, os pesos das conexões entre as unidades de processamento, que guardam, ao final do processo, o conhecimento que a rede adquiriu do ambiente em que está operando”(BRAGA; CARVALHO; LUDERMIR, 2007).

Os ajustes de parâmetros da rede são realizados por algoritmos de aprendizagem formados por um conjunto de procedimentos bem definidos que especificam quando e como deve ser alterado o valor de cada peso. Diversos métodos para treinamento foram elaborados, podendo ser associados em dois paradigmas principais: aprendizado supervisionado e o aprendizado não-supervisionado.

O aprendizado supervisionado é o mais comum no treinamento das RNAs, a rede aprende por meio de pares entrada-saída, ou seja, através de um conjunto de dados, são ajustados os pesos da rede, de forma a encontrar uma ligação entre os pares entrada-saída fornecidos, adquirindo assim o conhecimento para predizer o valor da saída de uma determinada entrada.

O algoritmo de aprendizado de retropulação de erros, ou *back-propagation* é o mais utilizado para o treinamento de redes MLP, por ser supervisionado utiliza mecanismo de correção de erros para alterar os pesos e reduzir a distância entre o resultado desejado e o resultado produzido pela rede.

O *back-propagation* é constituído da iteração de duas fases: a fase *forward* e a fase *backwards*. Na fase *forward*, o sinal se propaga da entrada da rede até a saída, sendo considerado apenas os valores atuais dos pesos e do bias. Na fase *backwards*, a saída da rede é comparada com a saída desejada, obtendo-se assim, um sinal de erro que será retropropagado da camada de saída até a camada de entrada. Após esta repropagação, os erros de cada neurônio são usados, para atualizar os pesos das conexões entre os neurônios e desta forma minimizar o erro (HAYKIN, 2001).

2.3 Deep Learning

François Chollet define *Deep Learning* (DL), ou Aprendizado Profundo, como uma subárea específica de *Machine Learning* (ML), sendo uma nova abordagem das representações de aprendizado a partir de dados que enfatizam o aprendizado de camadas sucessivas de representações significativas. No DL, essas representações em camadas são (quase sempre) aprendidas por meio de RNAs com grandes quantidades de camadas e neurônios para aprender padrões complexos em um largo volume de dados (CHOLLET, 2017).

Apesar do DL ser uma subárea antiga de ML, só obteve sua notoriedade no ínicio de 2010 devido ao avanço computacional para a execução de modelos robustos e o aumento da quantidade de dados disponíveis sobre temas complexos. Nos poucos anos desde que alcançou nada menos do que uma revolução no campo, com resultados notáveis em tarefas inteligentes como detecção de objetos, reconhecimento facial, reconhecimento de voz, tradução automática (LE-CUN; BENGIO; HINTON, 2015). Arquiteturas profundas (*deep architectures*) concederam ao DL a possibilidade de resolver problemas de IA mais complexos (BENGIO, 2009).

Os modelos de DL, como Redes Neurais Convolucionais (*Convolutional Neural Networks, CNN*), *Feed-Forward*, Redes Neurais Recorrentes e *Long-Short Term Memory*, têm sido aplicados em tarefas de aprendizado supervisionado e não-supervisionado, na qual a CNN, no momento, está em ascensão. A próxima subseção comprehende os conceitos, características e detalhamentos deste modelo.

2.3.1 Redes Neurais Convolucionais

Redes Neurais Convolucionais (*Convolutional Neural Networks, CNN*) estão sendo aplicadas em tarefas de classificação, detecção, localização, etc, destacam-se no reconhecimento de padrões de alta dimensionalidade, como imagens e vídeos (KHAN et al., 2018). Choi et al., em 2017, informam que o DL surge como uma ferramenta poderosa para analisar imagens médicas, devido a aplicação de uma CNN para a detecção automatizada de múltiplas doenças retinianas, ocasionando em um aumento na precisão do diagnóstico da doença (CHOI et al., 2017).

As CNN são modelos de RNAs do tipo *feedforward*, onde o fluxo de informações ocorre somente em uma direção, de suas entradas para suas saídas. Em geral, CNNs contém muitas camadas ocultas, na qual são formadas por camadas convolucionais (*convolutional*), *pooling*, normalização, etc, que são agrupadas em módulos, ou blocos. E no final da rede, ter uma ou duas camadas totalmente conectadas (RAWAT; WANG, 2017; GOODFELLOW; BENGIO; COURVILLE, 2016).

Os módulos são frequentemente empilhados uns sobre os outros para formar um modelo profundo. A Figura 2.22 ilustra a arquitetura típica da CNN para uma tarefa de classificação

de imagem de veículos. Uma imagem é inserida diretamente na rede, e isso é seguido por várias etapas de convolução e *pooling*. Posteriormente, as representações dessas operações alimentam uma ou mais camadas totalmente conectadas. Finalmente, a última camada totalmente conectada produz o rótulo da classe (RAWAT; WANG, 2017).

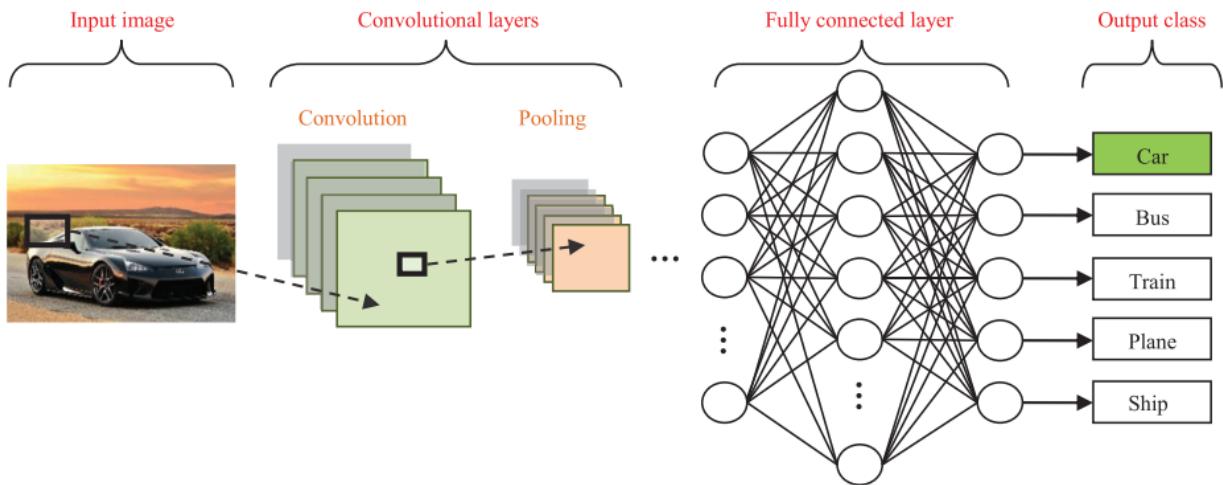


Figura 2.22: Exemplo de Arquitetura CNN para Classificação de Imagem (RAWAT; WANG, 2017)

Camada Convolucional

As camadas convolucionais servem como extratores de características, e assim, aprendem as representações das características das imagens de entrada. No processo de convolução são definidos os filtros, ou *kernels*, que percorrerá a imagem, gerando como saída um mapa de características, ou *feature maps*.

Um *kernel*, em processamento de imagens, é uma pequena matriz de tamanho NxN utilizado para aplicar filtros e transformações em imagens por meio da convolução. A Figura 2.23 demonstra com detalhes o processo de convolução, onde o *kernel* percorre toda a imagem por S pixels, e para cada localidade do *kernel*, uma multiplicação de matrizes é feita, e a somatória dessa multiplicação resulta em um pixel na nova imagem de saída (ARÁUJO, 2018).

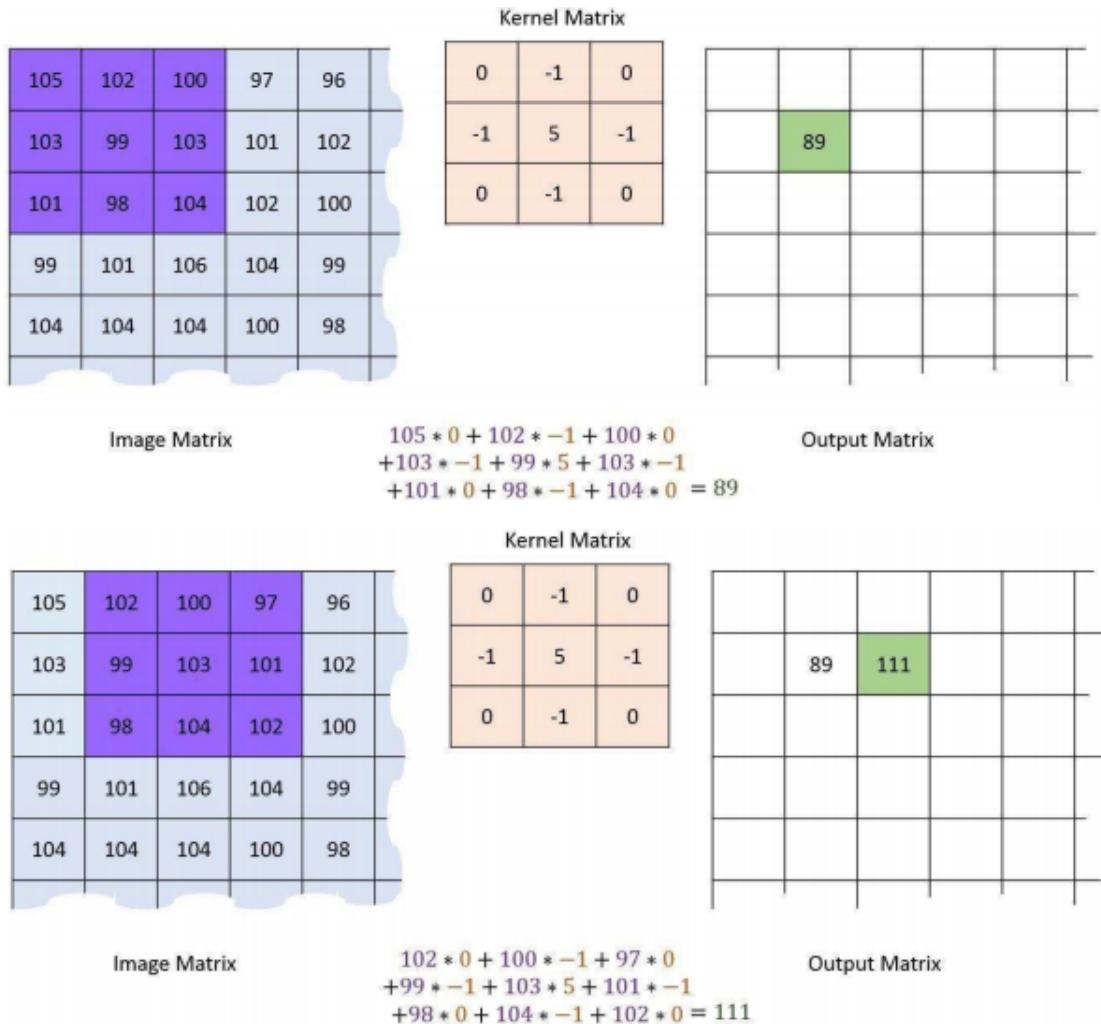


Figura 2.23: Processo de Convolução, *kernel* 3x3, S=1 (ARÁUJO, 2018)

As CNNs, em Visão Computacional (VC), recebem imagens como entrada, as camadas de convolução atuam reconhecendo características visuais. As primeiras camadas convolucionais começam identificando aspectos mais simples, como bordas e traços. Conforme o aumento da profundidade nos processos de convoluções, é possível obter características mais complexas, como texturas, objetos e até rostos.

Camada de *Pooling*

Após uma camada convolucional, geralmente existe uma camada de *pooling*. A finalidade dessa camada é reduzir a dimensão espacial dos *feature maps* e, assim, obter invariância espacial para inserir distorções e translações, resultando na redução do custo computacional da rede e evitando

o *overfitting* (LECUN; BENGIO; HINTON, 2015; RAWAT; WANG, 2017; KARPATHY, 2017).

No processo de *pooling*, os valores relativos a uma certa área do *feature map*, são modificados por alguma função dessa área. A função *max pooling*, é a mais utilizada e consiste em substituir os valores de uma área pelo valor máximo, ocasionando na redução da dimensão da representação de dados, eliminação de valores desprezíveis, e aceleração do processamento computacional necessário para as camadas seguintes (GOODFELLOW; BENGIO; COURVILLE, 2016), representada na Figura 2.24.

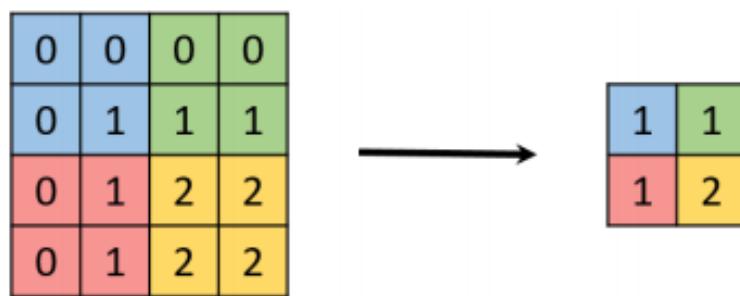


Figura 2.24: Aplicação de *max pooling* em uma imagem 4x4 utilizando um filtro 2x2 (ARAÚJO et al., 2017)

Camada Totalmente Conectada

As últimas camadas de uma CNN normalmente são as camadas totalmente conectadas, ou *Fully Connected Layers*, que são adicionadas após as camadas convolucionais e de *pooling*. O objetivo dessa camada é utilizar as características extraídas da imagem de entrada obtida pela saída dos processos convolucionais e de *pooling* para classificar a imagem em uma classe pré-determinada, como ilustrado na Figura 2.25.

As camadas totalmente conectadas são principalmente correspondidas às MLP que utilizam a função de ativação *softmax* na camada de saída, ou, às camadas de convolução que utilizam *kernels* de tamanho 1x1, na qual cada elemento é densamente conectado à todos os elementos da camada anterior (KHAN et al., 2018).

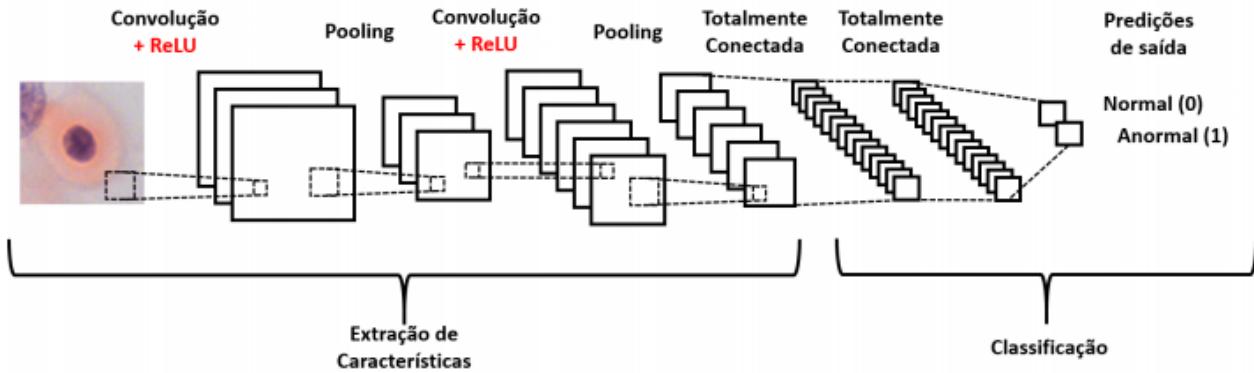


Figura 2.25: Ilustração da extração de características de uma imagem por uma CNN e sua posterior classificação (ARAÚJO et al., 2017)

Durante o processo de treinamento, as CNNs utilizam técnicas, como o *dropout*, para reduzir o tempo de treinamento e evitar erros de generalização, tais como o *overfitting* (GOODFELLOW; BENGIO; COURVILLE, 2016). Aráujo et al, esclarece que essa técnica consiste em remover, aleatoriamente a cada iteração de treinamento, uma determinada porcentagem dos neurônios de uma camada, readicionando-os na iteração seguinte. Essa técnica também confere à rede a habilidade de aprender atributos mais robustos, uma vez que um neurônio não pode depender da presença específica de outros neurônios (ARAÚJO et al., 2017).

2.4 Trabalhos Relacionados

O Reconhecimento Facial (RF) tem sido adotado para a área de segurança, sendo a autenticação e a privacidade, os principais temas em que a Realidade Aumentada (AR) atua como ferramenta auxiliar. Nesta seção serão apresentados alguns trabalhos da literatura similares ao abordado neste trabalho, que utilizam a AR junto com o RF.

O trabalho de Acquisti et al., em 2014, devido a alta disponibilidade de dados faciais identificados, como *Facebook* e *LinkedIn*; e não identificados, como sites onde geralmente os usuários utilizam pseudônimos, tais qual *Flickr* e *Tumblr*. Os autores projetaram um aplicativo para vincular, reidentificar os usuários em diferentes contextos (*online* e *offline*) e inserir informações das redes sociais do usuário (ACQUISTI et al., 2014).

Para o RF, foi utilizado o software *PittPatt*, um software para RF em imagens e vídeos

adquirido pela Google (NECHYBA; SCHNEIDERMAN, 2007). O aplicativo captura a imagem de uma pessoa, através do *cloud computing*, o aplicativo se comunica com o software *PittPatt* e, em seguida, sobrepõe seu nome previsto na tela (ACQUISTI et al., 2014).

O trabalho de Dolan et al., por sua vez, aborda um sistema HMD (*Head Mounted Display*) de AR. O sistema é configurado para realizar o RF para a entrada de uma segunda pessoa, e assim, é gerado um *token* para iniciar uma transferência *peer-to-peer* (DOLAN; KURIAN; WYLLIE, 2018).

Uma das vantagens apontadas pelos autores consiste quando um usuário de AR emprega *tokens* para permitir o envio de dados, resulta em, utilizar menos informações do que outros sistemas existentes. Ao utilizar menos informações para transferência de dados, o sistema reduz a quantidade de dados que serão enviados através da rede, ocasionando em uma melhora no desempenho da rede (DOLAN; KURIAN; WYLLIE, 2018).

Em Li et al., é proposto um sistema que aplica técnicas de *Machine Learning* para distinguir de forma confiável as contas de usuários com apenas câmeras comuns para tornar os logins de RF mais acessíveis aos desenvolvedores de sites e softwares (LI; CAI; SAXBERG, 2018).

O artigo de Menescal e Melo, em 2018, apresentou um protótipo de RF, baseado em métodos matemáticos, desenvolvido para atuar no registro e identificação dos alunos em sala de aula. Os métodos utilizados para o processo de reconhecimento busca extraír informações, relevantes de uma imagem, para, em seguida codificá-las e compara-las com outras imagens salvas em um banco de dados. A Rede Neural Convolucional demonstrou a sua eficiência no reconhecimento dos alunos e obteve a maior acurácia comparadas aos métodos tradicionais da Visão Computacional (MENESCAL; MELO, 2018).

Embora o trabalho de Acquisti et al. e Li et al. abordem o tema de RF como principal, são utilizadas ferramentas comerciais para o RF, como o *PittPatt*, devido ao custo da ferramenta, inviabiliza a utilização (ACQUISTI et al., 2014; LI; CAI; SAXBERG, 2018). Esta proposta de conclusão de curso se propõe a, de maneira análoga aos trabalhos de Dolan et al. e Menescal (DOLAN; KURIAN; WYLLIE, 2018; MENESCAL; MELO, 2018), considerar um sistema HMD de AR combinado com o modelo de redes neurais convolucionais.

Capítulo 3

Solução Proposta

Neste capítulo serão discorridos a estruturação da solução proposta, como características e funcionamento. Para a realização deste trabalho compreende a descrição da proposta, exposta na Seção 3.1. Os detalhamentos da proposta, tais como levantamento de requisitos, diagramas de caso de uso, a arquitetura, aplicação de Realidade Aumentada (AR), servidor, conjunto de dados, métricas de desempenho utilizadas e os modelos propostos, situados na Seção 3.2.

3.1 Descrição Geral da Proposta

Este trabalho de conclusão de curso teve por objetivo, como citado anteriormente, auxiliar o controle da entrada de alunos da Escola Superior de Tecnologia da Universidade do Estado do Amazonas (EST-UEA), por meio de uma interface de AR inteligente para o Reconhecimento Facial (RF) dos mesmos. Sendo dividida em 3 módulos: Aplicação de AR, Servidor e Modelo de Aprendizagem. Esta divisão é ilustrada na Figura 3.1.

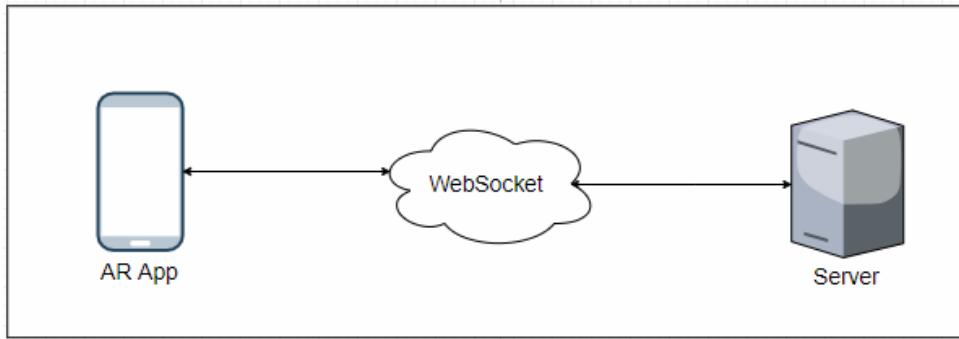


Figura 3.1: Visão Geral da Proposta

No módulo de aplicação de AR, implementou-se uma aplicação HMD (*Head Mounted Display*) de AR utilizando o Unity¹ como plataforma responsável por viabilizar o uso da AR na aplicação.

O módulo servidor, comunica-se de forma bidirecional utilizando o protocolo *WebSocket*. O servidor foi responsável pela transmissão e recepção dos dados na aplicação de AR, ou seja, a aplicação envia uma imagem para o servidor, este então, encaminha a imagem para o modelo de aprendizado proposto, o modelo classifica a imagem, e por fim, o resultado retorna para a aplicação.

O último módulo representa o modelo de aprendizado, encarregado de gerar uma predição de uma face em uma determinada imagem, informando a qual pessoa a face pertence. Para este modelo, fez-se uso das bibliotecas Keras² e Tensorflow³, oferecendo suporte à implementação das Redes Neurais Convolucionais, resultando em um modelo de aprendizado profundo.

3.2 Projeto SmartGlass

Esta seção apresenta um detalhamento da proposta, apresentando o levantamento de requisitos e diagramas de caso de uso nas Subseções 3.2.1 e 3.2.2. Em seguida, discorre-se a respeito da arquitetura do SmartGlass na Subseção 3.2.3. Os módulos de aplicação de AR e servidor, são encontrados respectivamente nas Subseções 3.2.4 e 3.2.5. A Subseção 3.2.6 comprehende a

¹<https://unity.com/pt>

²<https://keras.io>

³<https://www.tensorflow.org>

descrição do conjunto de dados, métricas de desempenho utilizadas e a proposição de modelos. Por fim, o pré-processamento dos dados é apresentada Subseção 3.2.7.

3.2.1 Levantamento de Requisitos

Levando-se em consideração a descrição geral da proposta na subseção anterior, um levantamento de requisitos foi realizado considerando possíveis serviços que a proposta ofereça. Para isto, foram elencados requisitos funcionais, para descrever explicitamente as funcionalidades da proposta, e requisitos não-funcionais, para definir as propriedades e restrições da proposta, por exemplo, o desempenho. Para um melhor entendimento da importância e complexidade dos requisitos, foram definidos graus de relevância: alta, média, e baixa.

Requisitos Funcionais

Tabela 3.1: Requisito Funcional RF01

Identificação	[RF01]
Requisito do usuário	Capturar face
Requisito do sistema	O sistema deve permitir que o usuário capture a face de uma pessoa e enviá-la para o servidor.
Importância	Alta
Complexidade	Média

Tabela 3.2: Requisito Funcional RF02

Identificação	[RF02]
Requisito do usuário	Reconhecer Face
Requisito do sistema	Após a captura da face de uma pessoa, o sistema deve receber o resultado do servidor e exibi-lo na realidade aumentada.
Importância	Alta
Complexidade	Alta

Requisitos Não-Funcionais

Tabela 3.3: Requisito Não-Funcional RNF01

Identificação	[RNF01]
Requisito do usuário	Desempenho
Requisito do sistema	O tempo limite para o reconhecimento facial de uma pessoa e o servidor enviar o resultado deve ser de no máximo 5 segundos.
Importância	Alta
Complexidade	Alta

Tabela 3.4: Requisito Não-Funcional RNF02

Identificação	[RNF02]
Requisito do usuário	Utilizar interface agradável e intuitiva
Requisito do sistema	A interface deve ser agradável ao utilizar um óculos HMD e intuitiva para facilitar o aprendizado do usuário.
Importância	Média
Complexidade	Baixa

Tabela 3.5: Requisito Não-Funcional RNF03

Identificação	[RNF03]
Requisito do usuário	Apresentar resultados de forma prática e elegante
Requisito do sistema	Os resultados devem ser visualizados de forma prática para o usuário e elegante para o conforto do usuário.
Importância	Média
Complexidade	Baixa

3.2.2 Diagramas de Caso de Uso

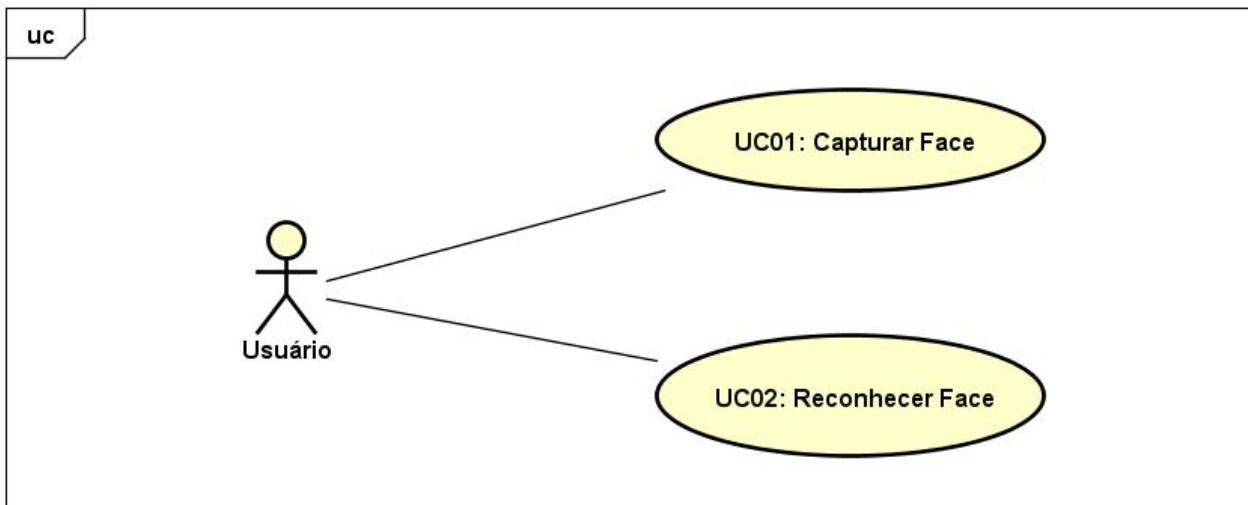


Figura 3.2: Diagrama de Caso de Uso do SmartGlass

Descrições dos Casos de Uso

Tabela 3.6: Descrição do Caso de Uso UC01

Identificação	[UC01] Capturar Face
Descrição	Este caso de uso tem por finalidade realizar a captura da face de uma pessoa, por meio da câmera do <i>smartphone</i> , e enviá-la para o servidor.
Atores	Usuário.
Pré-Condições	O usuário deve estar em um ambiente iluminado para capturar a face da pessoa.
Pós-Condições	Após capturar a face da pessoa, esta imagem será enviada para o servidor.
Fluxo Principal	<ol style="list-style-type: none"> 1. O usuário seleciona o botão para a captura da face; 2. O sistema captura a face da pessoa; 3. O sistema envia a face capturada para o servidor.
Fluxo Alternativo	1. O usuário não seleciona o botão para a captura da face.
Fluxo de Exceção	1. Caso o sistema verifique que o dispositivo não possui as especificações mínimas para utilizar a realidade aumentada, será exibido a mensagem (M01).
Mensagem	(M01) “Smartphone incompatível para a Realidade Aumentada.”
Regra de Negócio	<p>(RN01) O <i>smartphone</i> deve possuir uma versão do Android igual ou superior que a 4.0.3.</p> <p>(RN02) O <i>smartphone</i> deve possuir uma câmera igual ou superior a 8 <i>megapixels</i>.</p> <p>(RN03) O <i>smartphone</i> deve estar conectado à internet.</p>

Tabela 3.7: Descrição do Caso de Uso UC02

Identificação	[UC02] Reconhecer Face
Descrição	Este caso de uso tem por finalidade receber a face capturada pelo usuário, reconhecer a face, e devolver o resultado para a aplicação.
Atores	Usuário.
Pré-Condições	O usuário deve ter capturado a face da pessoa, e o sistema deve ter enviado a face capturada para o servidor.
Pós-Condições	O resultado do reconhecimento facial será enviado para aplicação, e apresentado na realidade aumentada.
Fluxo Principal	<ol style="list-style-type: none"> 1. O servidor recebe a face capturada da aplicação; 2. O servidor realiza o reconhecimento facial; 3. O servidor envia o resultado para a aplicação de realidade aumentada; 4. O sistema recebe o resultado e o apresenta na realidade aumentada.
Fluxo Alternativo	
Fluxo de Exceção	<ol style="list-style-type: none"> 1. Caso o sistema não encontre conexão com a internet, será exibido a mensagem (M01).
Mensagem	(M01) “Você está off-line. Verifique sua conexão.”
Regra de Negócio	(RN01) O smartphone deve estar conectado à internet.

3.2.3 Arquitetura do SmartGlass

A arquitetura do SmartGlass é baseado em uma estrutura de aplicação distribuída cliente-servidor, devido ao servidor executar as tarefas solicitadas e enviam uma resposta para a requisição do cliente. A Figura 3.3 representa a arquitetura do SmartGlass, sendo (I) a representação do servidor e a (II) a aplicação de AR.

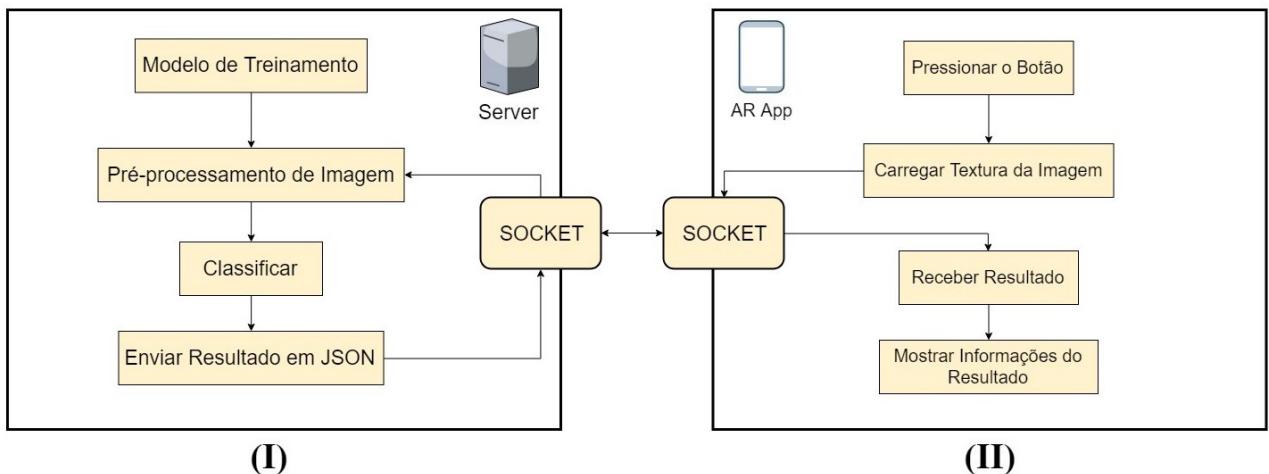


Figura 3.3: Arquitetura do SmartGlass

Na Figura 3.4 é ilustrado a funcionalidade principal do SmartGlass, isto é, capturar uma imagem, realizar o reconhecimento facial e apresentar o resultado em realidade aumentada.

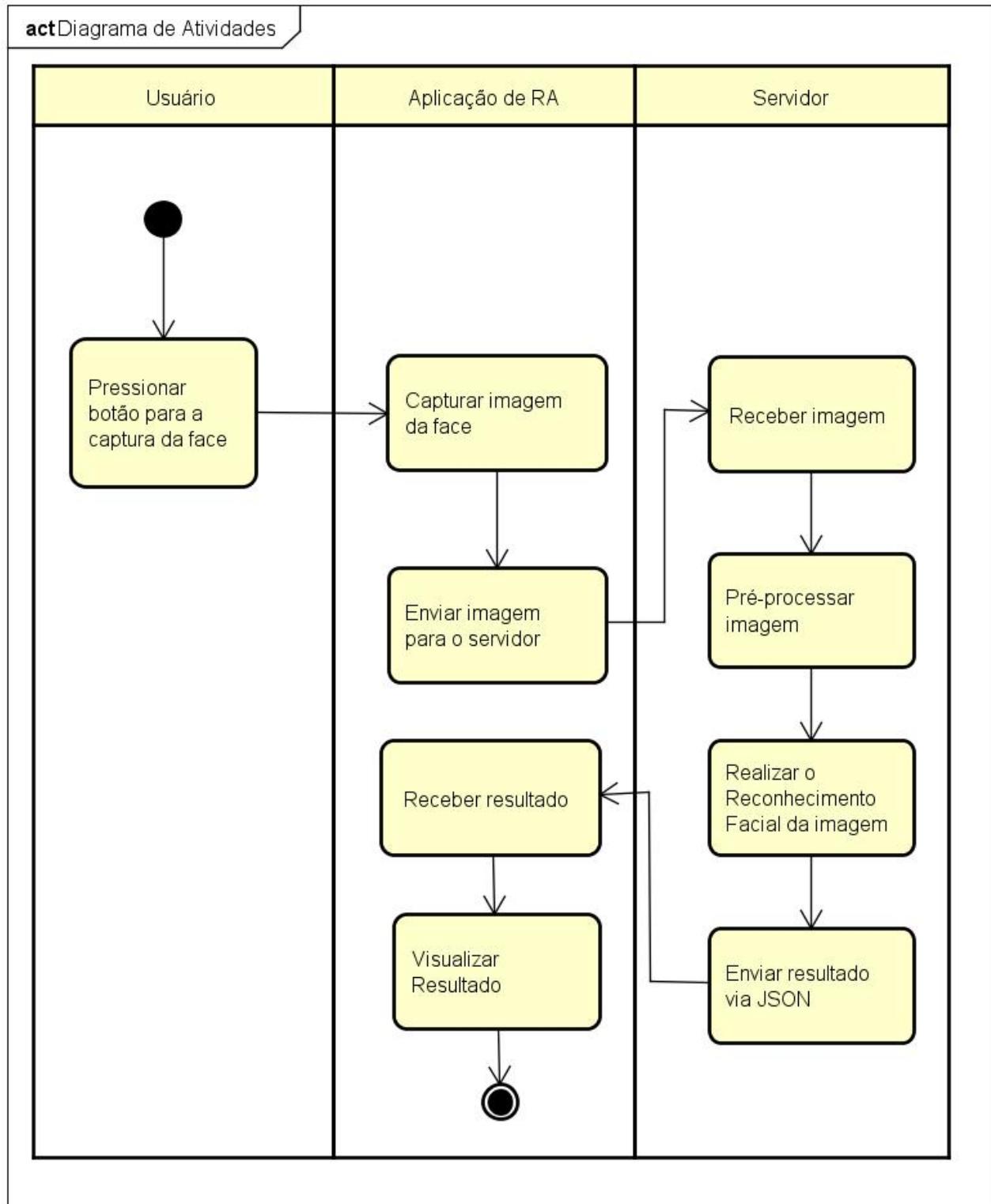


Figura 3.4: Diagrama de Atividade do SmartGlass

3.2.4 Módulo Aplicação de Realidade Aumentada

O módulo aplicação de Realidade Aumentada é a parte cliente da estrutura da aplicação. O aplicativo foi implementado na Unity *Engine*, utilizando o pacote Vuforia¹ para o desenvolvimento de realidade aumentada no aplicativo, e o óculos Gear VR², para o uso do *display HMD*. Apesar do óculos ser voltado para realidade virtual (RV), é possível aplicá-lo em AR sem perder o propósito do HMD, apenas retirando a tampa frontal do dispositivo para a uso da câmera do *smartphone*. Na Figura 3.5 é apresentado a aplicação do Gear VR neste módulo.

O Vuforia possui a maior gama de tipos de targets possíveis, sendo capaz de identificar além de imagens planas, objetos complexos, texto, e marcadores especiais, de escaneamento rápido. Está disponível para as plataformas iOS, Android, e também para a plataforma de desenvolvimento de jogos Unity (BOQUIMPANI; FILHO, 2017).

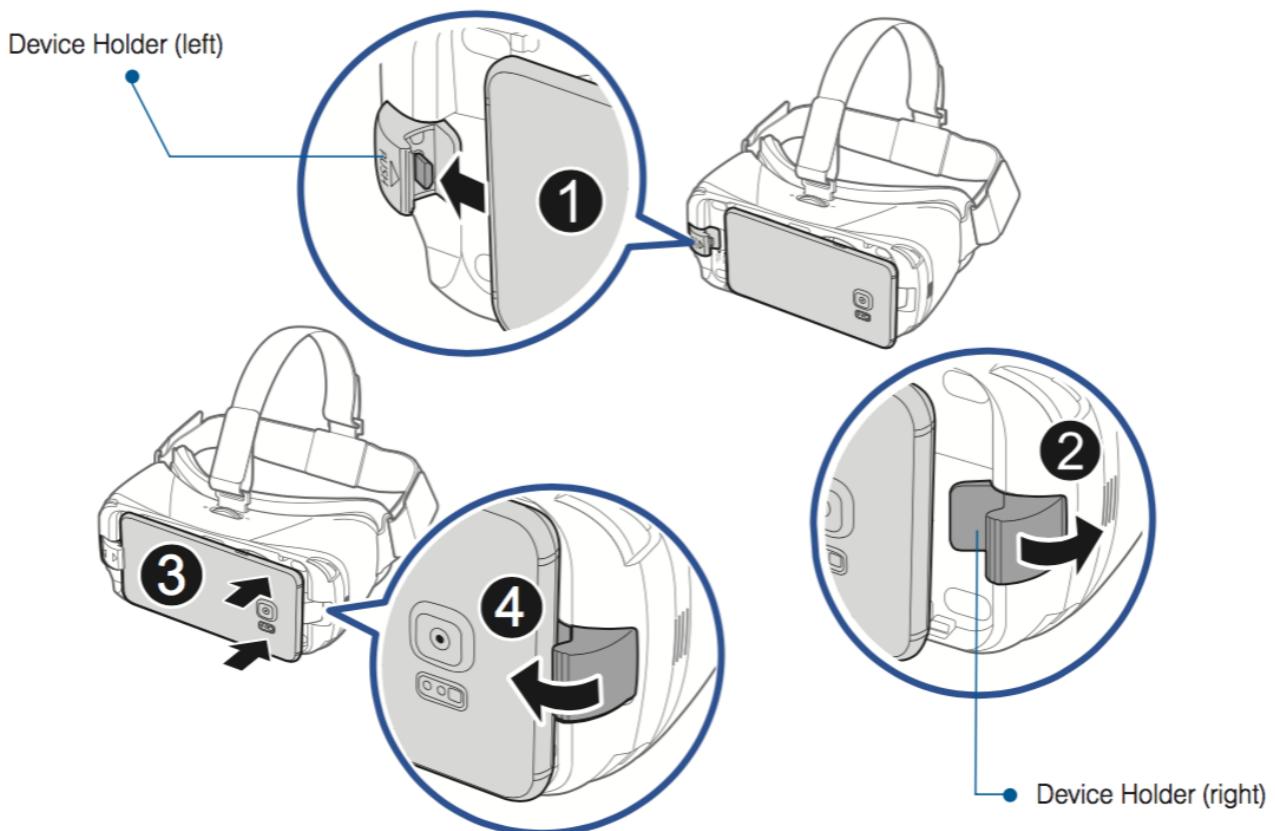


Figura 3.5: Aplicação do Gear VR na Realidade Aumentada (MATTERPORT, 2018)

¹<https://developer.vuforia.com>

²<https://www.samsung.com/global/galaxy/gear-vr/>

A Figura 3.6 representa a interface da aplicação, no momento em que a face é reconhecida pelo modelo de rede neural proposto, o resultado será enviado para a aplicação através de *WebSockets*, este então é exibido na tela com suas respectivas informações de nome e curso do aluno.

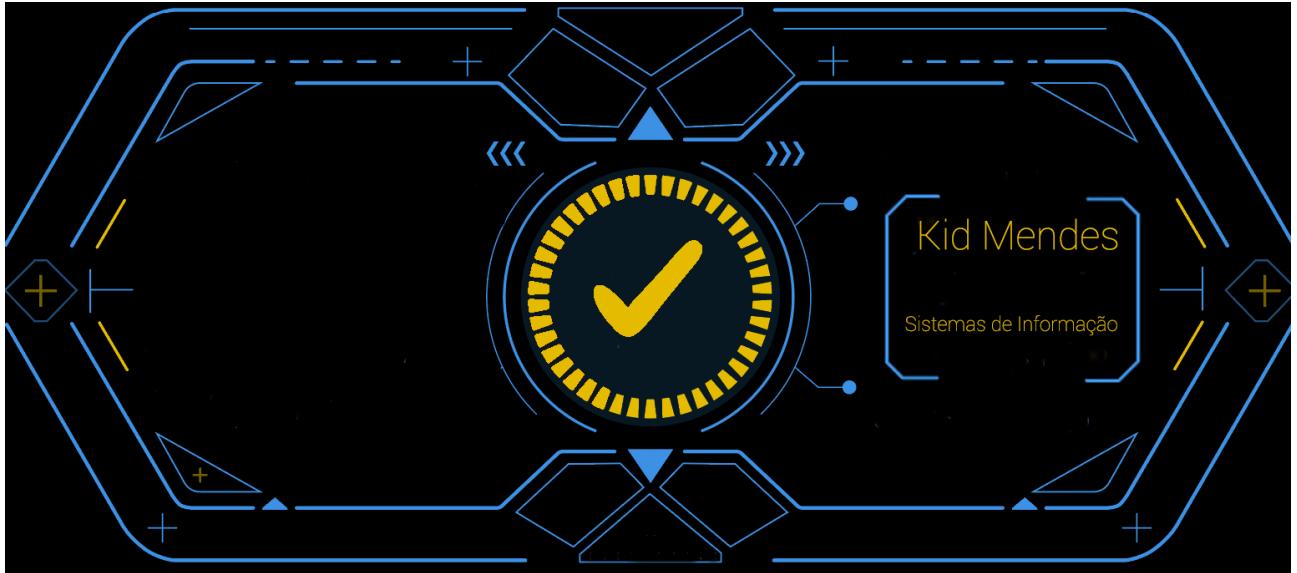


Figura 3.6: Interface da Aplicação

3.2.5 Módulo Servidor

Este módulo é responsável pela parte servidor da estrutura da aplicação. O servidor foi implementado na linguagem de programação Python, utilizando a biblioteca Python-SocketIO¹, para possibilitar uma comunicação bidirecional entre cliente e servidor usando o protocolo *WebSocket*.

Para contornar os problemas de atraso de comunicação, de forma a obter latências de conexões quase nulas ou em tempo real entre clientes e servidores é necessário utilizar mecanismos dos quais oferecem mais recursos do que o serviço oferecido pelo protocolo HTTP. Os *WebSockets* surgiram para prover esse tipo de serviço com retardo reduzido, o qual o HTTP não oferece apoio, sendo capazes de processar mais mensagens em uma determinada faixa de tempo em comparação aos sistemas que são baseados no protocolo HTTP, dessa forma aumentando a vazão de informação que o sistema é capaz de processar(GONÇALVES; BASTOS; OLIVEIRA,

¹<https://pypi.org/project/python-socketio/>

2014).

Devido a implementação de um canal de transmissão bidirecional e por ter um cabeçalho menor, o *WebSocket* ultrapassa o HTTP em relação ao número de requisições feitas e no tamanho das mensagens. O *handshake* do *WebSocket* é trivial, dado que é realizado apenas uma conexão, enquanto no HTTP é realizado por requisição. Após o estabelecimento da conexão, somente as mensagens serão trocadas, sem a necessidade de ter um cabeçalho extenso, conforme a Figura 3.7.



Figura 3.7: *Handshake* no *WebSocket* (GONÇALVES; BASTOS; OLIVEIRA, 2014)

O servidor foi responsável por inicialmente receber imagens de faces capturadas através da aplicação de AR. Após o recebimento das imagens, é realizado um pré-processamento nas imagens, para a normalização delas conforme a base de treinamento utilizada para a criação do modelo de aprendizagem. Por fim, é realizado a classificação facial da imagem, e o resultado é transmitido para a aplicação de AR.

3.2.6 Modelos Propostos

Os modelos de *Deep Learning* (DL) a serem considerados neste trabalho são as Redes Neurais Convolucionais (CNNs), onde serão abordadas diferentes arquiteturas e parâmetros de treinamento, para fins de comparação dos resultados obtidos e identificação de um modelo mais apto para a tarefa.

Dentre as CNNs a serem consideradas para este cenário, enfatiza-se a importância de treinar e testar arquiteturas que possam ser utilizadas em um servidor. Considerou-se inicialmente, a utilização da arquitetura FaceNet como proposta para o reconhecimento facial. Além da FaceNet, foi considerada o ResNet-50, um modelo mais profundo com a mesma finalidade e usando a base de dados VGGFace2.

As CNNs tornaram-se o estado da arte em métodos para tarefas de classificações de imagens. No entanto, uma das maiores limitações é que elas exigem uma grande quantidade de dados rotulados. Em muitas aplicações, coletar esses dados muitas vezes não são triviais.

Por exemplo, criar um sistema de reconhecimento facial em uma empresa utilizando métodos padrões de *deep learning*, o modelo de aprendizagem demandaria um grande número de imagens rotuladas dos funcionários para o treinamento em diversas épocas. Esses métodos podem não ser adequados, pois toda vez que houver a entrada de um novo funcionário, o modelo precisará ser retreinado. Outra abordagem seria o modelo ser treinado com poucas imagens dos funcionários, podendo ser usado para novos funcionários, sem a necessidade de retreinar. Esta abordagem é chamada de *One-shot Learning*, na qual foi baseada para a construção dos modelos propostos.

One-shot Learning

De acordo com Li, Fergus e Perona, “acredita-se que os humanos podem reconhecer entre 5.000 e 30.000 categorias de objetos (BIEDERMAN, 1987). A observação informal nos diz que aprender uma nova categoria é rápido e fácil, as vezes, exigindo muito poucos exemplos de treinamento: dadas 2 ou 3 imagens de um animal que você nunca viu antes, você pode geralmente reconhecê-lo mais tarde. Isso deve ser contrastado com o estado da arte na visão computacional, onde aprender uma nova categoria normalmente requer milhares, se não dezenas de milhares de

imagens de treinamento. Estas têm que ser coletadas e, as vezes, manualmente segmentadas e alinhadas, sendo uma tarefa tediosa e dispendiosa”(LI; FERGUS; PERONA, 2003).

Li, Fergus e Perona, indagaram duas perguntas: “O sistema visual humano viola o que parece ser um limite fundamental de aprendizagem?”, e “Os algoritmos de visão computacional poderiam ser igualmente eficientes?”. Eles discutem que uma possível explicação para a eficiência humana é que, ao aprender uma nova categoria, aproveita-se uma experiência anterior. Embora não tivesse visto uma jaguatirica antes, viu-se gatos, cachorros, cadeiras, pianos. A aparência das categorias conhecidas e, mais importante, a variabilidade em sua aparência, resulta em informações importantes sobre o que esperar em uma nova categoria. Isto pode permitir a aprendizagem de novas categorias a partir de poucos exemplos de treinamento (LI; FERGUS; PERONA, 2003).

Os autores exploraram essa hipótese em um *framework* bayesiano. Os métodos bayesianos permitem incorporar informações prévias sobre objetos em uma função de densidade de probabilidade “anterior” que é atualizada, quando as observações se tornam disponíveis, em um “posterior” a ser usado para reconhecimento. Mostraram que o algoritmo que propuseram foi capaz de aprender uma nova categoria não conhecida, utilizando apenas alguns exemplos de treinamento, esta abordagem foi denominada de *One-shot Learning*.

O *One-shot Learning* é um problema de categorização de objetos na visão computacional. Enquanto a maioria dos algoritmos de categorização de objetos baseados em *Machine Learning* exigem treinamento em centenas ou milhares de imagens e conjunto de dados muito grandes, o *One-shot Learning* visa aprender informações sobre categorias de objetos de uma, ou algumas, imagens de treinamento (KARUNAKARAN, 2018).

Rede Siamesa

O *One-shot Learning* pode ser implementado usando uma Rede Siamesa. As Rede Siamesas são um tipo especial de arquitetura de rede neural. Ao invés de um modelo aprender a classificar suas entradas, as redes neurais aprendem a diferenciar duas entradas, e encontrar a similaridade entre elas.

Uma Rede Siamesa consiste em duas redes neurais idênticas, cada uma inserindo uma das duas imagens de entrada. As últimas camadas das duas redes são então alimentadas para uma função *contrastive loss*, que calcula a similaridade entre as duas imagens (GUPTA, 2017), como ilustrado na Figura 3.8.

As redes são otimizadas utilizando uma função *contrastive loss* ou *triplet loss*. Como o objetivo da arquitetura siamesa não é classificar as imagens de entrada, mas sim diferenciá-las, então, estas funções apenas avaliam quão bem a rede está distinguindo um determinado par de imagens.

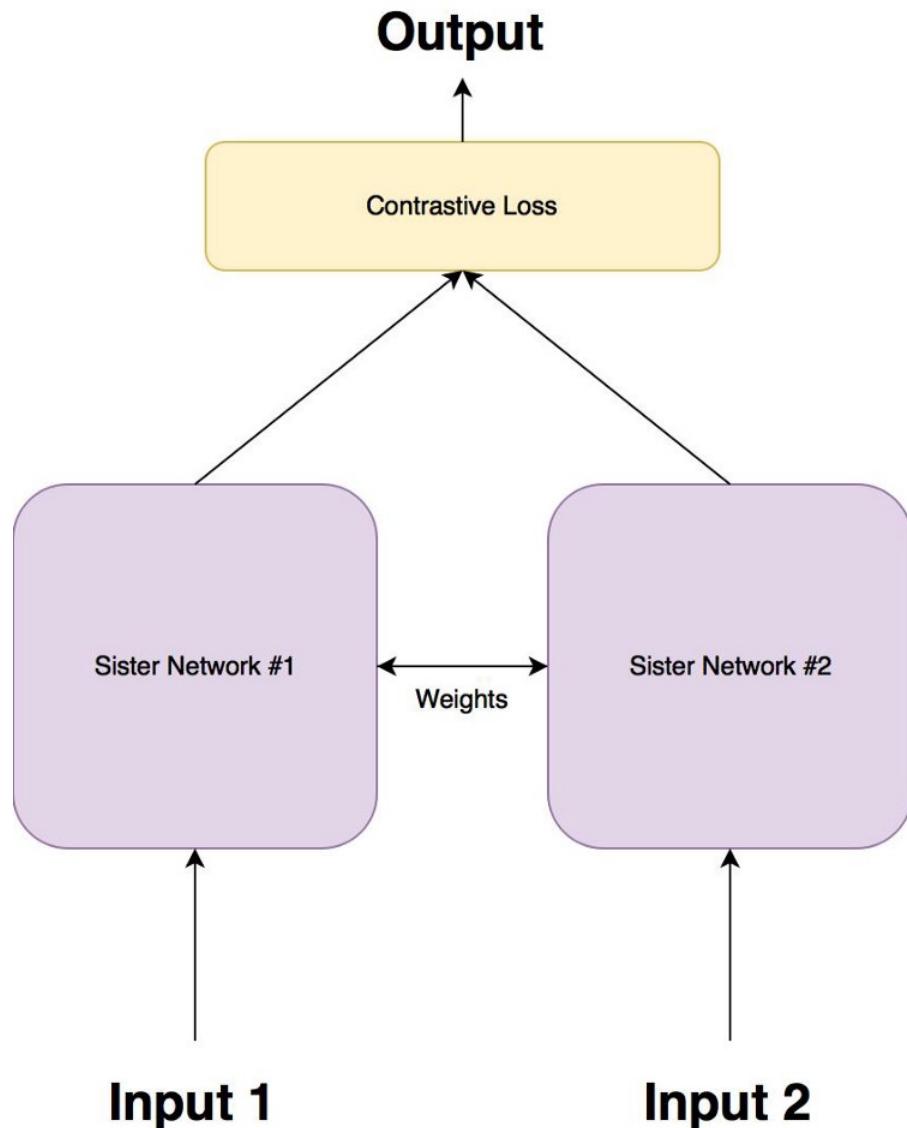


Figura 3.8: Arquitetura da Rede Siamesa (GUPTA, 2017)

FaceNet

Em 2015, a FaceNet foi introduzido pelos pesquisadores do Google. Os pesquisadores apontam que alcançaram o estado da arte do reconhecimento facial utilizando apenas 128 *bytes* por face, transformando a face em um espaço euclidiano de 128 vetores dimensionais. Dado que o modelo FaceNet tenha sido treinado com *triplet loss* para diferentes classes de faces para capturar as semelhanças e diferenças entre elas, a incorporação de 128 dimensões retornada pelo modelo pode ser usada para clusters (agrupar) de faces efetivamente (Schroff; Kalenichenko; Philbin, 2015).

A partir da criação desse espaço vetorial (*embedding*), tarefas como reconhecimento de faces e clustering (agrupamento) podem ser facilmente implementadas utilizando técnicas padrões baseadas na FaceNet, como os vetores de características. Desta forma, a distância seria mais próxima para rostos semelhantes e mais afastados para rostos não semelhantes.

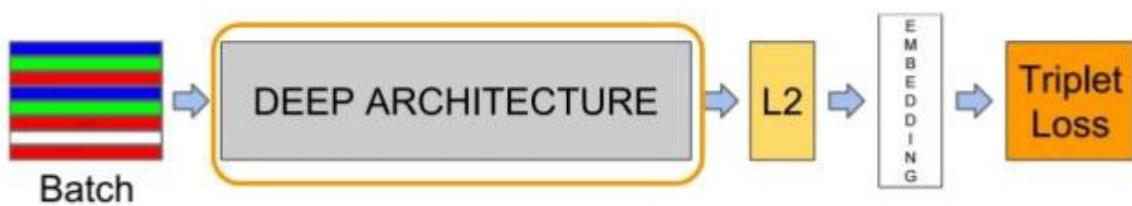


Figura 3.9: Estrutura da FaceNet (Schroff; Kalenichenko; Philbin, 2015)

O artigo descreve que a rede consiste na entrada *batch* em uma CNN profunda, seguida da normalização por L2, resultando no *embedding* da face. Por fim, o *triplet loss* é utilizado durante o treinamento.

A normalização L2, mais conhecida como distância Euclidiana, será utilizada para descobrir a distância entre dois vetores. Esta normalização é representada na Equação 3.1.

$$d(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2} \quad (3.1)$$

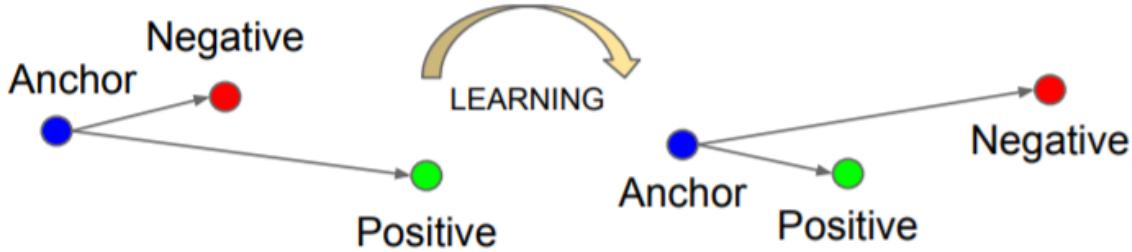


Figura 3.10: *Triplet Loss* na FaceNet (Schroff; Kalenichenko; Philbin, 2015)

O *triplet loss* minimiza a distância entre uma âncora (anchor) e um positivo (positive), ambos com a mesma identidade, e maximiza a distância entre a âncora (anchor) e um negativo (negative) de uma identidade diferente (Schroff; Kalenichenko; Philbin, 2015).

Implementação do Modelo FaceNet

O modelo para o reconhecimento facial, foi inspirado no artigo “FaceNet: A Unified Embedding for Face Recognition and Clustering”, devido ao uso de redes neurais convolucionais, e uma alta acurácia de 99.63% no *Labeled Faces in the Wild dataset* (LFW). Além do LFW, o Youtube Faces DB foi aplicado para o treinamento do modelo (Schroff; Kalenichenko; Philbin, 2015).

Para a implementação do modelo, utilizou-se do projeto *open source* Keras-OpenFace¹. Keras-OpenFace é um projeto que converte a implementação orginal do OpenFace em PyTorch para uma versão em Keras. Open-Face² é um projeto livre e *open source*, implementado em Python e Torch³ para reconhecimento facial com redes neurais profundas, baseado no artigo “FaceNet: A Unified Embedding for Face Recognition and Clustering”(AMOS; LUDWICZUK; SATYANARAYANAN, 2016). O módulo de treinamento do OpenFace é representado na Figura 3.11 .

¹<https://github.com/iwantooxxoox/Keras-OpenFace>

²<https://cmusatyalab.github.io/openface/>

³<http://torch.ch>

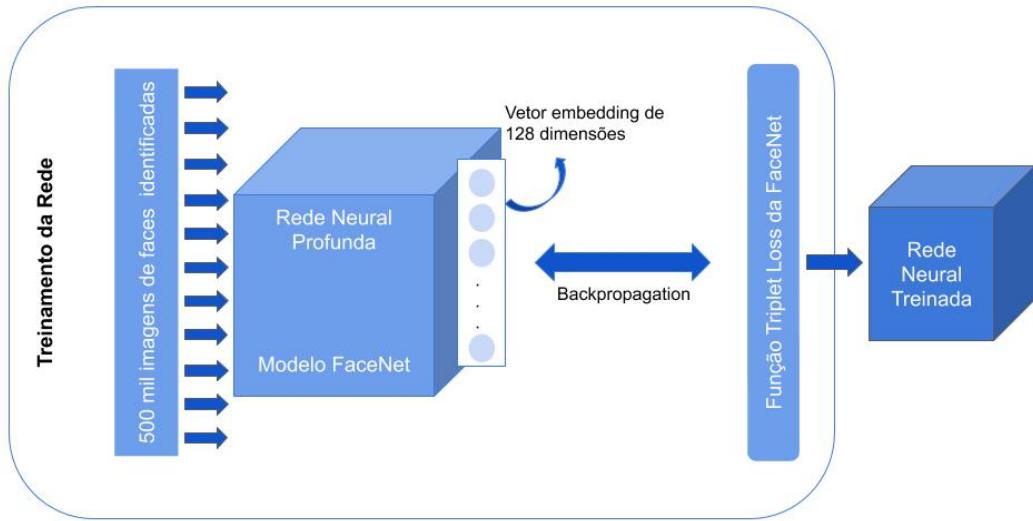


Figura 3.11: Módulo de Treinamento do OpenFace

O fluxo para a detecção da face em uma imagem, baseou-se no fluxo do OpenFace, podendo ser exemplificado na Figura 3.12 de Sylvester Stallone avaliado no *Labeled Faces in the Wild dataset*. Este fluxo é representado na Figura 3.13.

1. Detectar faces com modelos pré-treinados através do OpenCV¹;
2. Transformar a face para a entrada de uma rede neural. O OpenFace usa a estimativa de pose em tempo real² do dlib com a transformação *affine*³ do OpenCV, para tentar fazer com que os olhos e o lábio inferior apareçam no mesmo lugar na imagem;
3. Utilizar uma rede neural convolucional profunda para representar (*embed*) a face em 128 dimensões. O *embed* é uma representação genérica para o rosto de qualquer pessoa;
4. Ao final, pode ser aplicado técnicas de agrupamento (*clustering*), detecção de similaridades, e classificação para a conclusão da tarefa de reconhecimento. Neste modelo, foi escolhida a técnica de detecção de similaridades, para a aplicação da abordagem *One-shot Learning*.

¹<https://opencv.org>

²<http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html>

³<https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/warpAffine/warpAffine.html>

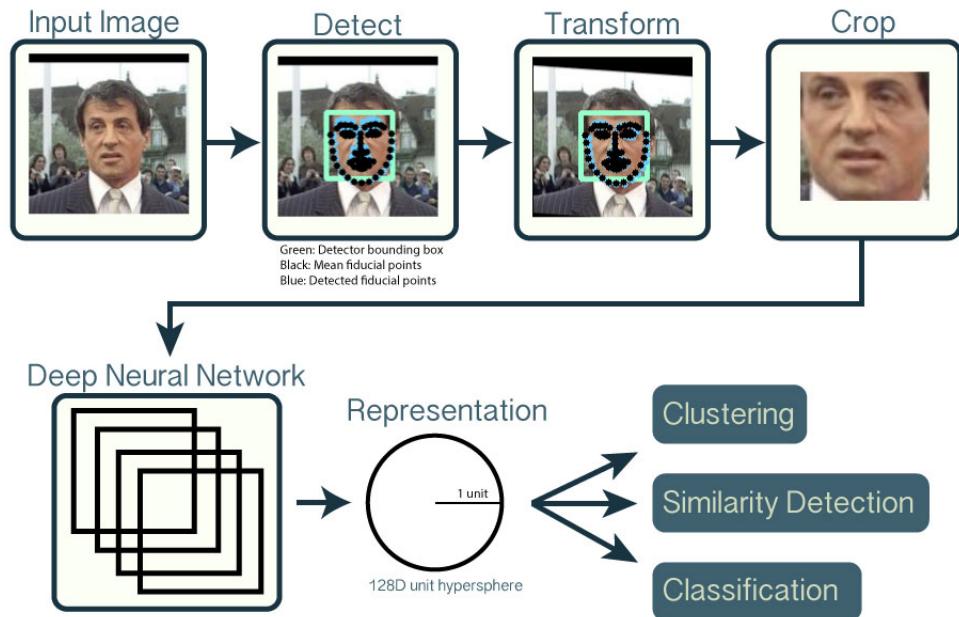


Figura 3.12: Fluxo para a detecção de face no OpenFace (AMOS; LUDWICZUK; SATYANARAYANAN, 2016)

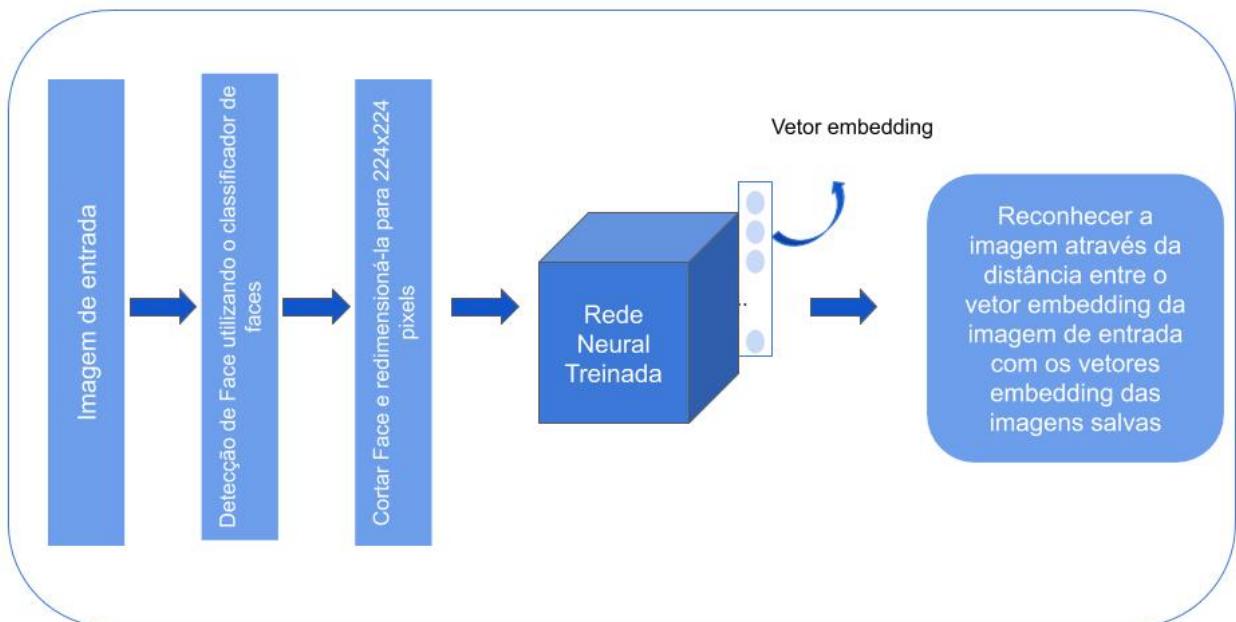


Figura 3.13: Reconhecimento facial utilizando o *One-shot Learning*

ResNet-50 com VGGFace2

A ResNet-50 é uma arquitetura desenvolvida por He et al., segundo os autores, o objetivo é apresentar uma estrutura de aprendizado residual para facilitar o treinamento de redes substancialmente mais profundas. Além disso, a profundidade das representações é de importância central para muitas tarefas de reconhecimento visual (HE et al., 2015).

A VGGFace2 é *dataset* em larga escala de faces, contém 3.31 milhões de imagens de 9131 categorias. As imagens são baixadas através do *Google Image Search* e abrange grandes variedades de postura, idade, iluminação, etnia e profissão. Segundo os autores, o *dataset* foi coletado com três objetivos em mente: (i) ter um grande número de identidades e um grande número de imagens para cada identidade; (ii) cobrir uma grande variedade de postura, idade e etnia; e (iii) minimizar o ruído do rótulo (CAO et al., 2017).

Para avaliar o desempenho do reconhecimento facial com VGGFace2, os autores treinaram uma CNN ResNet-50 com VGGFace2 que levou a uma melhor performance de reconhecimento. Afirmam que, utilizando os modelos treinados nesse *dataset* alcança-se a performance de estado da arte no reconhecimento facial do *dataset* IJB (CAO et al., 2017).

O cálculo de similaridade entre as faces é feito através de um descritor de faces, este descritor é obtido das redes treinadas da seguinte forma: (i) a caixa delimitadora da face é redimensionada para que o lado mais curto tenha 256 pixels; (ii) então o centro de corte de 224x224 da imagem do rosto é usado como entrada para a rede. O espaço vetorial (*embedding*) de faces é um vetor de 2048 dimensões, o comprimento do vetor é normalizado através da distância euclidiana L2 (Equação 3.1). A distância entre os descritores de face é calculada através da similaridade de cossenos, representada na Equação 3.2.

$$d(x, y) = \frac{\sqrt{\sum x * y}}{\sqrt{\sum x^2} * \sqrt{\sum y^2}} \quad (3.2)$$

Visão Geral do Modelo Proposto

Após o processo de detecção de face em uma imagem, esta será enviada para a rede com o melhor desempenho. Esta rede contém duas CNNs idênticas totalmente conectadas, com os mesmos

pesos e aceitando duas imagens diferentes. Para comparar duas imagens, cria-se o *embedding* para ambas imagens, alimentando o modelo separadamente. Por fim, usa-se a métrica de desempenho adequada, podendo ser a Equação 3.1 euclidiana ou Equação 3.2 similaridade de cossenos para encontrar a distância, que o valor será menor para faces semelhantes e maior para diferentes faces, caso a distância seja menor que o *Threshold* (definido inicialmente como 0.68), a imagem será reconhecida, caso contrário será retornado que não foi possível reconhecer a face. A Figura 3.14 representa o modelo de rede proposta.

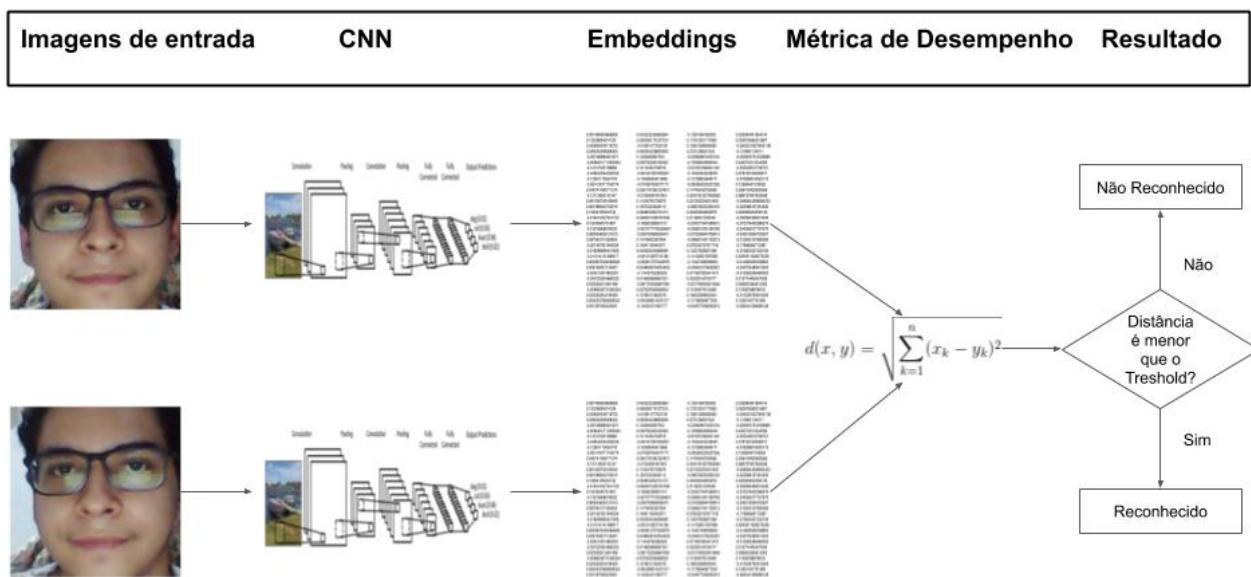


Figura 3.14: Representação do Modelo Proposto

A comparação feita entre as duas imagens é estruturada da forma de uma rede siamesa. Então pode-se dizer que este é um modo de aprendizado para comparar duas faces. Finalmente, pode-se concluir que utilizando o modelo FaceNet ou ResNet-50, é alcançada uma implementação da abordagem *One-shot Learning*.

3.2.7 Pré-processamento dos dados

A fim de adequar melhor o conjunto de dados para os modelos de CNNs utilizados, realizou-se um pré-processamento das imagens da base de dados usada neste trabalho. Para isto,

padronizou-se o modo RGB, tamanho das imagens para 224 x 224 *pixels* e a utilização de classificadores para detectar faces em uma imagem. Além disso foi efetuada a normalização das imagens, com o objetivo de escalar os valores dos *pixels* para o intervalo [0,1], realizada por meio da divisão dos valores de entrada por 255. Segundo Chollet, a prática de normalizar as imagens proporciona a convergência das redes neurais convolucionais, obtendo-se um melhor ajuste de pesos quando as entradas são números pequenos (CHOLLET, 2017).

Capítulo 4

Resultados e Discussão

Este capítulo comprehende a apresentação e discussão dos resultados obtidos decorrentes do treinamento e teste das CNNs, para endereçar a tarefa de *Deep Learning* considerada na geração do modelo de aprendizado que será integrado na parte do módulo servidor, descrito na Subseção 3.3.6. Os resultados obtidos estão organizados segundo abordagens sequenciais, levando em consideração o aumento da complexidade das estratégias específicas da prática de aprendizado profundo visando obter melhores resultados.

4.1 Detectar Faces: Cascade Classifier e Multi-task Cascade CNN

A detecção de faces é um problema na visão computacional de situar e localizar um ou mais rostos em uma imagem. Situar uma face em uma imagem refere-se a encontrar a coordenada do rosto na imagem, enquanto a localizar refere-se a demarcação da extensão do rosto, geralmente por meio de uma caixa delimitadora em torno do rosto. Dada uma imagem, um sistema de detecção de faces produzirá zero ou mais caixas delimitadoras que contêm rostos. As faces detectadas podem ser fornecidas como entrada para um sistema subsequente, como um sistema de reconhecimento facial (HJELMĀS; LOW, 2001). Para a detecção facial foi utilizado o Cascade Classifier e Multi-task Cascade CNN, descritos nas próximas subseções.

4.1.1 Cascade Classifier

O Cascade Classifier foi introduzido por Paul Viola e Michael Jones em 2001. No artigo publicado, características efetivas são aprendidas usando o algoritmo AdaBoost, embora sendo o mais importante, vários modelos são organizados em uma hierarquia de complexidade crescente, chamada de “cascata”(Viola; Jones, 2001).

Classificadores mais simples operam diretamente nas regiões da face candidata, agindo como um rude filtro, enquanto classificadores complexos operam apenas nas regiões candidatas que assemelham-se como faces. Cascade Classifier é um classificador modestamente complexo que também foi aprimorado e refinado nos últimos 20 anos.

O uso do detector facial Cascade Classifier é fornecido através da biblioteca OpenCV¹, na qual é implementada em C++ que fornece uma interface em Python. O benefício dessa biblioteca é que fornece modelos de detecção de faces pré-treinadas e fornece uma interface para treinar um modelo em seu próprio conjunto de dados.

4.1.2 Multi-task Cascade CNN

Diversos métodos de *Deep Learning* foram desenvolvidos para a detecção facial, uma das abordagens mais populares é a *Multi-Task Cascaded Convolutional Neural Network* (MTCNN), desenvolvida por Kaipeng Zhang, et al. em 2016 (ZHANG et al., 2016).

O MTCNN é popular por alcançar resultados de estado da arte em uma variedade de conjuntos de dados e por inclusive ser capaz de reconhecer outros recursos faciais, como olhos e boca, chamado de detecção *landmark*.

A rede usa uma estrutura em cascata com três redes: primeiro a imagem é redimensionada para uma variedade de tamanhos diferentes (denominada de *image pyramid*), o primeiro modelo (*Proposal Network* ou P-Net) propõe-se a candidatar regiões de face; o segundo modelo (*Refine Network* ou R-Net) é responsável por filtrar as caixas delimitadoras; e o terceiro modelo (*Output Network* ou O-Net) indica pontos de referência faciais (ZHANG et al., 2016). Esta estrutura é representada na Figura 4.1.

¹<https://opencv.org>

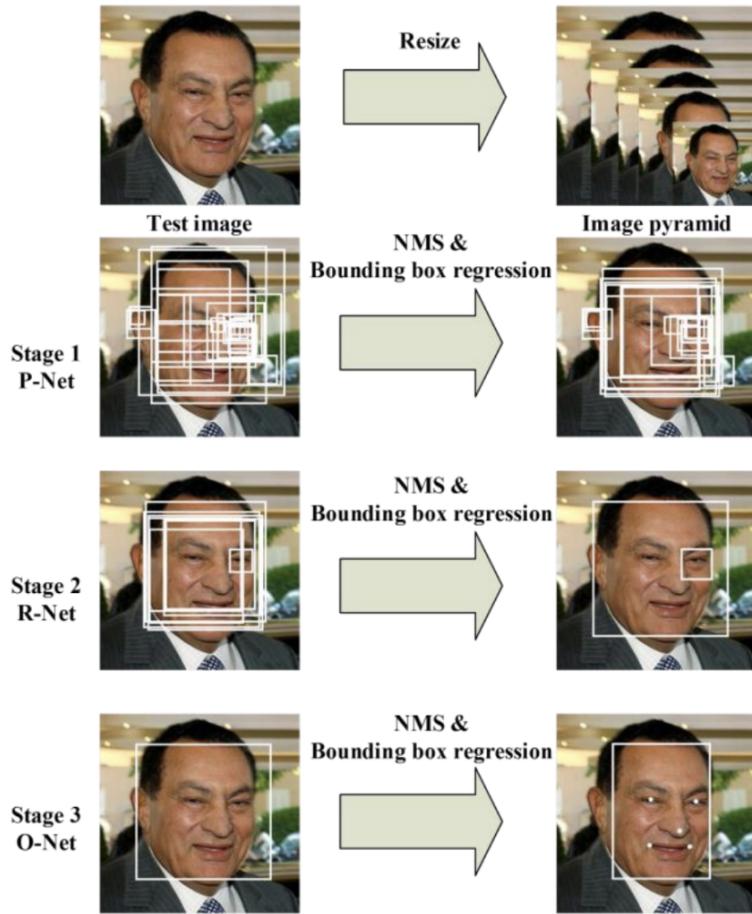


Figura 4.1: Estrutura da rede MTCNN (ZHANG et al., 2016)

Devido a complexidade para implementar a arquitetura MTCNN, utilizou-se implementações de código aberto desta arquitetura, podendo ser treinada em novos conjunto de dados, tal como modelos pré-treinados que podem ser usados diretamente para a detecção facial. Iván de Paz Centeno disponibilizou a versão da MTCNN¹ para Python, esta versão foi escolhida para representar a MTCNN no projeto.

4.1.3 Cascade Classifier x Multi-task Cascade CNN

A fim de obter o melhor detector de faces no *SmartGlass*, foram realizados testes comparativos com diferentes imagens, contendo diferentes posturas, distância, quantidade de faces, iluminações. Os dois detectores faciais conseguiram detectar com sucesso as faces em imagens com pouca quantidade de rostos, mas conforme ocorre o aumento de rostos, a diferença dos detec-

¹<https://github.com/ipazc/mtcnn>

tores surge. Esta diferença pode ser visualizada na Figura 4.2.



Figura 4.2: Comparativo Cascade Classifier e Multi-task CNN

Na Figura 4.2, o Cascade Classifier não conseguiu detectar todas as faces na imagem, detectando 15 de 18 faces, enquanto o Multi-task CNN identificou todas as faces da imagem, incluindo uma mão que foi considerada uma face, totalizando 19 caixas delimitadoras. Diante desta situação, o Multi-task CNN foi o detector facial escolhido para o *SmartGlass*, pois considerando a segurança, é preferível reconhecer todas as faces em uma imagem contendo apenas uma classificação errônea do que não ser possível identificar todos os rostos na imagem.

4.2 Abordagem 1:FaceNet

A primeira abordagem de treinamento considerou o uso dos modelo FaceNet de maneira canônica, isto é, tais como são definidos na literatura com os conjunto de dados *Labeled Faces in the Wild dataset* (LFW) e *Youtube Faces DB* (YTDB). Adotou-se em todas as camadas de ativação internas a função de ativação não-linear ReLU por ser simples de calcular e por satisfazer os critérios de continuidade e diferenciação, requeridos pelo algoritmo de *backpropagation* e para

a função de ativação da camada de saída foi utilizada a função *Softmax*.

As camadas densas que compõem os últimos blocos da arquitetura são compostas por 1, 1, 128 neurônios. O *optimizer* utilizado para o algoritmo de *backpropagation* foi o *solver SGD*, fornecido pela biblioteca Keras, considerando a taxa de aprendizagem no valor de 0.05, com o tamanho de *batch* 1800, do mesmo modo que a FaceNet (Schroff; Kalenichenko; Philbin, 2015).

As imagens da base de dados foram normalizadas antes de serem apresentadas às redes, como estratégia de pré-processamento apresentada na Seção 3.2.7 , isto é, todos os valores dos *pixels* das imagens foram escalonados para o intervalo [0,1] por meio de uma divisão por 255.

Após o treinamento, foi então realizada a fase de testes com os dados reservados para este fim, conforme o método de validação *holdout*. Os resultados desta abordagem encontram-se sintetizados na Tabela 4.1.

Tabela 4.1: Resultados da FaceNet na Abordagem 1

Modelo	Função de ativação	Dataset	Tamanho do Dataset	Acurácia
FaceNet	<i>ReLU</i>	LFW	13.233 imagens	0.98
FaceNet	<i>ReLU</i>	YTDB	3.425 vídeos	0.95

A partir da análise dos resultados obtidos e a utilização dos conjunto de dados LFW e YTDB, foi possível verificar que a normalização dos dados de entrada garantiram um aprendizado efetivo que refletiu positivamente na etapa de testes, alcançando altos níveis de acurácia nos conjunto de dados com a função de ativação ReLU.

4.3 Abordagem 2: ResNet-50 com VGGFace2

A segunda abordagem adotou uma nova rede, baseada na arquitetura canônica ResNet-50, as camadas internas com a função de ativação *ReLU*, treinadas apenas com as imagens da base de dados normalizadas. A arquitetura da ResNet-50 foi utilizada como modelo pré-treinado, pois o modelo foi treinado anteriormente em um conjunto de dados e contém os pesos e viés que representam os recursos de qualquer conjunto de dados.

Os recursos aprendidos geralmente são transferíveis para dados diferentes. Por exemplo,

um modelo treinado em um grande conjunto de dados de imagens de aves conterá recursos aprendidos, como arestas ou linhas horizontais, às quais podem ser transferidos para um próprio conjunto de dados. O principal motivo para a escolha de um modelo pré-treinado foi a economia de tempo, pois os autores da ResNet-50 computaram recursos para aprender muitas *features*, que podem ser reaproveitadas, beneficiando assim, o modelo.

O modelo proposto nesta abordagem foi a ResNet-50 treinada com o conjunto de dados VGGFace2, o formato de entrada da rede com dimensões de 224x224 *pixels* e definindo a função *average* para os processos de *pooling*, de modo que os *features maps* na saída do modelo sejam reduzidos a um vetor utilizando a função *global average pooling*. Vale ressaltar que o VGGFace2 contém 3.31 milhões de imagens de 9131 categorias, quantidade essa maior que a junção dos conjuntos de dados *Labeled Faces in the Wild dataset* e *Youtube Faces DB*, viabilizando formas do modelo aprender novas *features*.

O treinamento da rede se deu de forma análoga a Abordagem 1. Os resultados obtidos pelo treinamento e teste encontram-se detalhados na Tabela 4.2.

Tabela 4.2: Resultados da ResNet-50 com VGGFace2 na Abordagem 2

Modelo	Função de ativação	Dataset	Tamanho do Dataset	Acurácia
ResNet-50	<i>ReLU</i>	VGGFace2	3.31 M imagens	0.996

Apesar dos resultados próximos aos encontrados na Abordagem 1, a ResNet-50 com VGGFace 2 apresentou ser o melhor modelo com acurácia de 0.996 e função de ativação *ReLU*. Este modelo foi elencado para a detecção de face em uma imagem na aplicação *SmartGlass*.

Foram consideradas duas métricas de desempenho para encontrar a distância de uma face para outra: a Equação 3.1 Euclidiana e a Equação 3.2 Similaridade de Cossenos. O resultado será menor para faces semelhantes e maior para diferentes faces, caso a distância seja menor que o *Threshold* (definido inicialmente como 0.68), a imagem será reconhecida, caso contrário será retornado que não foi possível reconhecer a face. Após realizado testes com voluntários, a Similaridade de Cossenos obteve melhor desempenho na tarefa de reconhecimento com o valor do *Threshold* definido como 0.5 e obtendo 3 segundos para reconhecer a face, alcançando a melhor performance dos modelos propostos para a tarefa de reconhecimento de faces.

Capítulo 5

Considerações Finais

Este trabalho de conclusão de curso consistiu em propor o desenvolvimento de uma interface de Realidade Aumentada (AR) inteligente para o Reconhecimento Facial (RF) de alunos da Escola Superior de Tecnologia da Universidade do Estado do Amazonas. Para este fim, foi proposto, treinado e testado modelos de Rede Neural Convolucional (CNN) utilizando as técnicas de *Deep Learning*. Este RF auxilia o controle da entrada de alunos na universidade, visando o aumento da segurança dentro da universidade.

A abordagem utilizada para o RF, foi o *One-shot Learning* implementada em uma rede siamesa, contendo duas CNNs idênticas totalmente conectadas. A escolha do *One-shot Learning* foi uma alternativa à modelos tradicionais de *Machine Learning* que necessitam de treinamentos com grandes conjuntos de dados, contendo centenas ou milhares de imagens, em contrapartida, o *One-shot Learning* visa aprender informações das entradas a partir de uma ou algumas imagens de treinamento.

Considerando o modelo adotado neste trabalho, foram conduzidas duas abordagens de treino e teste utilizando diferentes arquiteturas de CNNs: FaceNet e ResNet-50 com VGGFace2, esta última, sendo desenvolvida para este projeto. Não houve uma variação no parâmetro de função de ativação, fazendo uso da *ReLU*, descrita na literatura como a mais adequada ao problema. Considerou-se também técnicas de normalização da entrada, imagens no modo RGB. Estas redes, parâmetros, hiperparâmetros e técnicas foram explorados de maneira sistemática em dois cenários diferentes.

Para a detecção de faces, foram avaliados o Cascade Classifier e Multi-task Cascade CNN (MTCNN), devido a complexidade da arquitetura deste último e por apresentar um melhor desempenho para fins de comparação, elencou-se o MTCNN responsável por essa atividade.

De maneira geral, foi possível constatar uma progressiva melhora ao passo que as abordagens foram conduzidas durante o processo de treino e teste dos modelos de aprendizado apresentados. Os resultados obtidos mostraram que a melhor CNN para esta tarefa foi a ResNet-50 com VGGFace2 sujeita à abordagem 2, a qual considerou um largo conjunto de dados e função de ativação *ReLU*, obtendo-se uma acurácia de 0.996.

Para avaliar o desempenho do modelo para encontrar a distância de um *embedding* face para outro, foi utilizada a Equação de Similaridade de Cossenos, com o *Threshold* de 0.5. Isto resultou em um tempo de reconhecimento rápido, levando em consideração o envio da imagem através do servidor, o processamento na rede e a resposta, resultando em um tempo resposta padrão de 3 segundos.

Desta feita, os resultados obtidos decorrentes deste trabalho de conclusão de curso permitiram com sucesso que o *SmartGlass* alcançasse as três características para o melhor funcionamento de uma aplicação de AR: combinar o real e o virtual; alinhar os objetos reais e virtuais entre si; e execução interativamente em tempo real, este último, foi a característica mais beneficiada devido à alta performance da CNN proposta.

Em trabalhos futuros, podem dar continuidade ao estudo aqui realizado, sendo desenvolvidos com o intuito de poder reconhecer pessoas em tempo real (sem a necessidade de pressionar um botão), atingir melhores métricas através da utilização de diferentes arquiteturas, novas técnicas e outras ferramentas disponíveis no mercado.

Referências Bibliográficas

- ACQUISTI, A. et al. Face recognition and privacy in the age of augmented reality. In: . [S.l.: s.n.], 2014.
- AMOS, B.; LUDWICZUK, B.; SATYANARAYANAN, M. *OpenFace: A general-purpose face recognition library with mobile applications*. [S.l.], 2016.
- ARAÚJO, F. H. et al. Redes neurais convolucionais com tensorflow: Teoria e prática. *SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. III Escola Regional de Informática do Piauí. Livro Anais-Artigos e Minicursos*, Sociedade Brasileira de Computação, v. 1, p. 382–406, 2017.
- ARÁUJO, G. A. d. S. Uma abordagem de reconhecimento de gestos aplicado à língua brasileira de sinais (libras). 2018. Disponível em: <<http://dspace.bc.uepb.edu.br/jspui/handle/123456789/19030>>.
- Azuma, R. et al. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, v. 21, n. 6, p. 34–47, Nov 2001. ISSN 0272-1716.
- AZUMA, R. T. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, v. 6, n. 4, p. 355–385, 1997. Disponível em: <<https://doi.org/10.1162/pres.1997.6.4.355>>.
- BARAKONYI, I.; FAHMY, T.; SCHMALSTIEG, D. Remote collaboration using augmented reality videoconferencing. In: *Graphics Interface*. [S.l.: s.n.], 2004.
- BELLINI, H. et al. Profiles in innovation - virtual augmented reality - understanding the race for the next computing platform. 2016. Disponível em: <<https://www.goldmansachs.com/insights/pages/technology-driving-innovation-folder/virtual-and-augmented-reality/report.pdf>>.
- BENGIO, Y. Learning deep architectures for ai. *Foundations*, v. 2, p. 1–55, 01 2009.
- BIEDERMAN, I. Recognition-by-components: A theory of human image understanding. *Psychological review*, v. 94, p. 115–47, 05 1987.
- BILLINGHURST, M. Introduction to augmented reality. In: . [S.l.: s.n.], 2004. p. 266.
- BOOK, D. L. *O Perceptron Parte 2*. 2016. Disponível em: <<http://deeplearningbook.com.br/o-perceptron-parte-2/>>.

- BOQUIMPANI, A. R.; FILHO, S. F. A realidade aumentada como novo paradigma da interface homem-máquina: um caso de estudo aplicado à leitura de rótulos nutricionais. Universidadse Federal Fluminense, 2017.
- BRAGA, A. de P.; CARVALHO, A. de L. F.; LUDELMIR, T. *Redes neurais artificiais: teoria e aplicações*. 2. ed. [S.l.]: LTC Editora, 2007.
- CAO, Q. et al. Vggface2: A dataset for recognising faces across pose and age. *CoRR*, abs/1710.08092, 2017. Disponível em: <<http://arxiv.org/abs/1710.08092>>.
- CARMIGNIANI, J.; FURHT, B. Augmented reality: An overview. In: _____. [S.l.: s.n.], 2011. p. 3–46.
- CARUSO, G. Augmented reality. p. 7, 2017. Disponível em: <<http://www00.unibg.it/dati/corsi/38001/53141-AR\lesson.pdf>>.
- CHOI, J. Y. et al. Multi-categorical deep learning neural network to classify retinal images: A pilot study employing small database. *PLOS ONE*, v. 12, p. e0187336, 11 2017.
- CHOLLET, F. *Deep Learning with Python*. 1st. ed. Greenwich, CT, USA: Manning Publications Co., 2017. ISBN 1617294438, 9781617294433.
- CRAIG, A. B. *Understanding Augmented Reality: Concepts and Applications*. 1st. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013. ISBN 9780240824086, 9780240824109.
- DOLAN, H.; KURIAN, M. J.; WYLLIE, G. M. *Facilitating Digital Data Transfers Using Augmented Reality Display Devices*. [S.l.]: Google Patents, 2018. US Patent App. 15/353,005.
- FUCHS, H. et al. Augmented reality visualization for laparoscopic surgery. In: . [S.l.: s.n.], 1998. p. 934–943.
- GONÇALVES, G.; BASTOS, P.; OLIVEIRA, D. O uso de websockets no desenvolvimento de sistemas baseados em uma arquitetura front-end com api. In: *Anais da I Escola Regional de Sistemas de Informação do Rio de Janeiro*. Porto Alegre, RS, Brasil: SBC, 2014. p. 57–63. ISSN 0000-0000. Disponível em: <<https://portaldeconteudo.sbc.org.br/index.php/ersi-rj/article/view/5785>>.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>.
- GUPTA, H. *One Shot Learning with Siamese Networks in PyTorch*. 2017. Disponível em: <<https://hackernoon.com/one-shot-learning-with-siamese-networks-in-pytorch-8ddaa10340e>>.
- HAYKIN, S. S. *Redes neurais : princípios e práticas*. [S.l.: s.n.], 2001.
- HE, K. et al. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Disponível em: <<http://arxiv.org/abs/1512.03385>>.
- HEBB, D. O. *The organization of behavior: A neuropsychological theory*. New York: Wiley, 1949. Hardcover. ISBN 0-8058-4300-0.

- HERREMA, J. Evaluation of graphical user interfaces for augmented reality based manual assembly support. In: . [S.l.: s.n.], 2013.
- HJELMĀS, E.; LOW, B. K. Face detection: A survey. *Computer Vision and Image Understanding*, v. 83, n. 3, p. 236 – 274, 2001. ISSN 1077-3142. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S107731420190921X>>.
- HOOKEY, A. *Google Glass - The Smart Wearable Intelligence*. 2015. Disponível em: <<https://theogm.com/2015/09/17/google-glass-the-smart-wearable-intelligence/>>.
- KARPATHY, A. *Convolutional Neural Networks for Visual Recognition*. 2017. Disponível em: <<http://cs231n.github.io/convolutional-networks/>>.
- KARUNAKARAN, D. *One shot learning explained using FaceNet*. 2018. Disponível em: <<https://medium.com/intro-to-artificial-intelligence/one-shot-learning-explained-using-facenet-dff5ad52bd38>>.
- KATO, H. et al. Virtual object manipulation on a table-top ar environment. In: . [S.l.: s.n.], 2000. p. 111 – 119. ISBN 0-7695-0846-4.
- KHAN, S. et al. A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision*, v. 8, p. 1–207, 02 2018.
- KIRNER, C. Realidade virtual e aumentada: Conceitos, projetos e aplicações. In: . [S.l.: s.n.], 2007. p. 02–04.
- KIRNER, C.; TORI, R. Fundamentos de realidade aumentada. In: . [S.l.: s.n.], 2006.
- KUNKEL, N. et al. *Augmented and virtual reality go to work*. 2016. Disponível em: <<https://www2.deloitte.com/insights/us/en/focus/tech-trends/2016/augmented-and-virtual-reality.html>>.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, p. 436–44, 05 2015.
- LEE, J.-Y. et al. Design and implementation of an augmented reality system using gaze interaction. *Multimedia Tools and Applications*, v. 68, 04 2011.
- LI, F. F.; FERGUS, R.; PERONA, P. A bayesian approach to unsupervised one-shot learning of object categories. In: . [S.l.: s.n.], 2003. v. 2:, p. 1134–1141.
- LI, R.; CAI, S.; SAXBERG, T. Login authentication with facial gesture recognition. Santa Clara: Santa Clara University, 2018., 2018. Disponível em: <<https://scholarcommons.scu.edu/cseng\senior/111>>.
- LOPES, M.; REITER, R. F.; REIS, D. Animar: um aplicativo para criação de animações com realidade aumentada e interface tangível. In: . [S.l.: s.n.], 2018. p. 983.
- MARTINS-FILHO, L. S.; MOL, A. A.; ROCHA, R. Desenvolvimento de ferramenta computacional para auxílio ao projeto de gemas lapidadas. *Rem: Revista Escola de Minas*, scielo, v. 58, p. 367 – 373, 12 2005. ISSN 0370-4467. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0370-44672005000400011&nrm=iso>.

MATTERPORT. *How do I use Virtual Reality?* 2018. Disponível em: <<https://support.matterport.com/hc/en-us/articles/224736667-How-do-I-use-Virtual-Reality->>.

MCCULLOCH, W.; PITTS, W. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, p. 127–147, 1943.

MENESCAL, R. C.; MELO, A. M. Biometria facial para avaliação de competências essenciais em um ambiente educacional: Avaliação do caso de sala de aula nas universidades. *Revista de Tecnologia da Informação e Comunicação da Faculdade Estácio do Pará*, v. 1, n. 2, p. 66–74, 2018. ISSN 2595-8798. Disponível em: <<http://www.revistasfap.com/ojs3/index.php/tic/article/view/241>>.

MICROSOFT. *HoloLens 2 - A realidade misturada está pronta para os negócios.* 2019. Disponível em: <<https://www.microsoft.com/pt-br/hololens>>.

MILGRAM, P. et al. Augmented reality: A class of displays on the reality-virtuality continuum. *Telemanipulator and Telepresence Technologies*, v. 2351, 01 1994.

MINSKY, M.; PAPERT, S. *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA, USA: MIT Press, 1969.

MISSFELDT, M. *How does Google glass work?* 2017. Disponível em: <<https://www.varifocals.net/google-glass/>>.

MOREIRA, S. *Rede Neural Perceptron Multicamadas*. 2018. Disponível em: <<https://medium.com/ensina-ai/rede-neural-perceptron-multicamadas-f9de8471f1a9>>.

NECHYBA, M. C.; SCHNEIDERMAN, H. Pittpatt face detection and tracking for the clear 2006 evaluation. In: STIEFELHAGEN, R.; GAROFOLI, J. (Ed.). *Multimodal Technologies for Perception of Humans*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 161–170. ISBN 978-3-540-69568-4.

NEWTRADE. *Indústria passa a investir mais em realidade virtual e aumentada.* 2017. Disponível em: <<https://newtrade.com.br/tecnologia/industria-passa-investir-mais-em-realidade-realidade-virtual-e-aumentada/>>.

RAWAT, W.; WANG, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, v. 29, p. 1–98, 06 2017.

REIS, B. *Redes Neurais - Funções De Ativação*. 2016. Disponível em: <<http://www.decom.ufop.br/imobilis/redes-neurais-funcoes-de-ativacao/>>.

RIBO, M. et al. Hybrid tracking for outdoor augmented reality applications. *Computer Graphics and Applications, IEEE*, v. 22, p. 54 – 63, 12 2002.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, p. 65–386, 1958.

Schmalstieg, D.; Höllerer, T. Augmented reality: Principles and practice. In: *2017 IEEE Virtual Reality (VR)*. [S.l.: s.n.], 2017. p. 425–426. ISSN 2375-5334.

- Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 815–823. ISSN 1063-6919.
- SILVA, A. P. S. Previsão do preço de liquidação das diferenças por meio de redes neurais artificiais. 2018. Disponível em: <<http://www.monografias.ufop.br/handle/35400000/1289>>.
- SILVA, E. M. Padrões mapeados localmente em multiescala aplicados ao reconhecimento de faces. Universidade Estadual Paulista (UNESP), 2018.
- ULLMER, B.; ISHII, H. Emerging frameworks for tangible user interfaces. *IBM Systems Journal*, v. 39, p. 915 – 931, 02 2000.
- Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. [S.l.: s.n.], 2001. v. 1, p. I–I.
- ZHANG, K. et al. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, Institute of Electrical and Electronics Engineers (IEEE), v. 23, n. 10, p. 1499â1503, Oct 2016. ISSN 1558-2361. Disponível em: <<http://dx.doi.org/10.1109/LSP.2016.2603342>>.
- ZHOU, F.; DUH, H.; BILLINGHURST, M. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, v. 2, p. 193–202, 09 2008.