

UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**IA E COMPUTAÇÃO AFETIVA: UMA ABORDAGEM DE
APERFEIÇOAMENTO DO DIALOGO HUMANO-COMPUTADOR**

THÁRCIO THALLES ALMEIDA SILVA

VITÓRIA DA CONQUISTA – BA
DEZEMBRO/2021

IA E COMPUTAÇÃO AFETIVA: UMA ABORDAGEM DE APERFEIÇOAMENTO DO DIÁLOGO HUMANO-COMPUTADOR

THÁRCIO THALLES ALMEIDA SILVA
ALEXSANDRA OLIVERA ANDRADE

Monografia apresentada ao Curso de Graduação em Ciência da Computação da Universidade Estadual do Sudoeste da Bahia, como requisito parcial para obtenção do Grau de Bacharel em Ciência da Computação. Orientadora: Profa. Dr^a. Alexsandra Oliveira Andrade

VITÓRIA DA CONQUISTA – BA
DEZEMBRO/2021

DEDICATÓRIA

Dedico esse trabalho a minha família e meus amigos que sempre estiveram comigo e me apoiaram durante essa difícil jornada.

AGRADECIMENTOS

Primeiramente, agradeço a minha família. Meus pais Adeir e Lucenilza e meus irmãos Dean, Dayane e Paloma que sempre me apoiaram nessa jornada e foram essenciais para que eu pudesse me tornar quem eu sou.

Aos meus amigos que tanto me apoiaram durante os períodos difíceis que passei, tenho muito orgulho de mim por ser o primeiro aluno da turma a se formar, mas sei que isso não seria possível sem meus queridos colegas e amigos que me ajudaram. Deixo aqui uma passagem para cada um de vocês que marcaram minha vida acadêmica e são amizades que irei levar para a vida. Obrigado Natália e Vinicius, por sempre estarem presentes me dando base emocional, conhecimento e sabedoria, tenho certeza que nossa amizade me tornou uma pessoa melhor e sem vocês esse trabalho não teria sido concluído. Obrigado Ricardo, Vitor e Thiago por me acompanharem desde o início do curso, tirando minhas dúvidas de programação, me passando muito conhecimento, me fazendo companhia e deixando essa jornada mais leve. Agradeço a todos os outros amigos que fiz durante o curso e que além do diploma são conquistas que tem valor inestimável.

A minha orientadora, Alexsandra, que foi minha segunda mãe durante o curso, me apoiando, dando conhecimento, acreditando em mim, dando vida aos tantos projetos que atuamos juntos e sendo uma grande amiga durante todo esse período.

Agradeço os professores por todo conhecimento passado, Celina, e todos os outros que me ajudaram durante esse período, fica aqui o meu muito obrigado a todos os alunos da minha turma de 2017, todos vocês foram importantes para a conclusão do fim dessa etapa.

RESUMO

A inteligência artificial (I.A), é um campo da Ciência da Computação que tem o intuito de simular ações específicas de humanos e animais com uma performance melhorada. Apesar de tal objetivo, essa área ainda possui muitos obstáculos para concretizar tais feitos.

Além disso existem ainda muitos mitos e incertezas que circundam a área e fazendo que se torne confuso o seu estado da arte, entretanto em meio a tantos rumores, existem trabalhos e pesquisas atuais que estão dando grandes passos para conseguir tal exibição performática, alguns desses trabalhos utilizam conceitos bastante promissores como a computação afetiva por exemplo que visa estudar a emoção humana envolvida no uso de um software.

Baseando-se no ideal da computação afetiva e pensando na potencial ferramenta que pode se tornar um software inteligente capaz de analisar emoções e também dialogar, este trabalho propôs uma Inteligência Artificial que seja capaz de reconhecer macro expressões faciais em seres humanos. Utilizando visão computacional, tal inteligência artificial, irá se conectar com um chatbot, para que ele possa utilizar as expressões detectadas com o fim de contextualizar a conversa e dessa forma tornar o dialogo o mais natural possível.

Palavras-chave: Computação afetiva, Inteligência artificial, Visão computacional.

ABSTRACT

Artificial intelligence (A.I) is a field of Computer Science that aims to simulate specific actions of humans and animals with an improved performance. Despite this objective, this area still has many obstacles to achieve such feats.

In addition, there are still many myths and uncertainties surrounding the area and making its state of the art confused, however amidst so many rumors, there are current works and research that are taking great steps to achieve such a performance exhibition, some of these works use very promising concepts such as affective computing, for example, which aims to study the human emotion involved in the use of software.

Based on the ideal of affective computing and thinking about the potential tool that can become an intelligent software capable of analyzing emotions and also dialoguing, this work proposed an Artificial Intelligence capable of recognizing macro facial expressions in human beings. Using computer vision, such artificial intelligence will connect with a chatbot, so that it can use the detected expressions in order to contextualize the conversation and thus make the dialogue as natural as possible.

Keywords: Affective computing, Artificial intelligence, Computer vision.

SUMÁRIO

OBJ1 - INTRODUÇÃO	8
2- FUNDAMENTAÇÃO TEÓRICA.....	11
2.1 - Inteligência Artificial	11
2.2 - Tensorflow e funções nativas.....	12
2.2.1 – Mapa de características	12
2.2.2 – Kernel size e regularizer.....	13
2.2.3 - Função de ativação e padding.....	13
2.3 - Arquitetura CNNs.....	13
2.4 - Computação afetiva	14
2.5 - Chatbot	15
2.7 - Visão computacional.....	17
3.1- Equipamentos e softwares.....	19
3.2- Código da arquitetura e explicação.....	20
3.2.1 – Código parte 2.....	20
3.3 – Processamento de imagem e código	21
4- EXPERIMENTOS E RESULTADOS	24
4.1 – Resultado dos experimentos da visão computacional	24
4.2 – Construção do chatbot e integração da I.A.....	31
OBJ5 - CONCLUSÃO.....	33
OBJ6- TRABALHOS FUTUROS	35
OBJ REFERÊNCIAS.....	36

1 - INTRODUÇÃO

Woebot é uma inteligência artificial (I.A) de chatbot desenvolvida pela Universidade de Stanford com o objetivo de tornar a saúde mental mais acessível. A I.A está disponível em formato de aplicativo, no qual o usuário pode ter conversas direcionadas a respeito de algum problema que ele esteja vivendo.

Demirci (2018) reconhece o êxito da woebot e atribui a esta ótimas escolhas tomadas no design e na implementação, tais como: a boa escolha de vocabulário que gera conforto no usuário, a simplicidade de uso e a utilização da TCC (Terapia Cognitivo Comportamental), que é uma ótima escolha para o tratamento a distância, uma vez que acentua a participação do paciente no processo terapêutico. O fato de o sistema não fazer a análise das expressões durante seu diálogo torna a conversação mais suscetível a falhas em relação ao contexto em que a conversa se insere, o que torna o diálogo menos natural.

Por esse motivo, esse trabalho propõe o desenvolvimento de uma I.A com o objetivo de analisar as expressões do usuário no decorrer da conversa, para que a I.A não emita uma resposta descontextualizada em relação ao sentimento do usuário. Para isso, será utilizado a computação afetiva, que trata da interação do usuário com a máquina levando em conta as emoções presentes durante a interação.

Dentro deste contexto, o objetivo geral desse trabalho se trata de desenvolver uma I.A que apresenta uma melhor eficiência de resposta contextualizada através do reconhecimento de macro expressões faciais do usuário utilizando a computação afetiva.

A inteligência artificial (I.A) vem ganhando um espaço cada vez maior no mercado, sua atuação inclui vendas online (chatbot), marketing digital e assistentes pessoais. Além dessas aplicações, a I.A também está começando ser inserida no tratamento de doenças psicológicas, como ansiedade e depressão, um exemplo da aplicação nessas áreas é o woebot, um aplicativo que tem o intuito de auxiliar pessoas que possuem ou estão passando por algum problema pessoal.

O woebot atua utilizando a TCC (Terapia Cognitiva Comportamental), e através dela induz os usuários do aplicativo a pensamentos positivos da situação. O aplicativo, apesar de ser bastante efetivo segundo relatos na plataforma de download, que se deve ao fato de possuir uma base sólida de profissionais (foi desenvolvido em um laboratório da Stanford), ele somente analisa o usuário pelo texto, ou seja, o

sistema ignora fatores externos que influenciam a interpretação de um diálogo, como os sinais de expressões faciais do interlocutor. De acordo com Silva (2020), isso prejudica a análise emocional do indivíduo, uma vez que mensagens de texto sozinhas podem não fornecer o contexto necessário para a interpretação.

Dessa forma pode-se afirmar que aplicações como o woebot já vêm ajudando diversas pessoas pelo mundo e poderiam auxiliar ainda mais as pessoas e empresas que utilizam de bots de resposta automatizada caso pudessem analisar também as expressões faciais dos usuários. Por isso, foi proposta neste trabalho uma inteligência artificial capaz de dialogar com pessoas (processamento de texto), e em paralelo analisar suas macros expressões faciais (processamento de imagem), dessa forma deixando o diálogo mais conciso.

Logo, unir esses dois aprendizados para que a I.A de processamento de texto seja capaz de ter melhores respostas para o usuário, melhorará a eficiência do software pois irá juntar dados relacionados ao texto com dados das macros expressões e assim contextualizar a conversa, tornando-a mais dinâmica, interativa e funcional no sentido de auxiliar o usuário, ou simplesmente ter uma outra função a depender do sistema implementado em que irá ser feito a aderência ao chatbot.

De acordo com Barbosa (2017), a separação física entre o profissional de saúde e o paciente nas consultas pode comprometer a riqueza e a complexidade que o contato visual, o olhar, as expressões faciais, ou seja, de pistas que o paciente pode deixar para o profissional da psicologia que poderiam modificar o significado de expressões.

No caso do woebot, tais pistas são escritas, e é válido ressaltar que em um diálogo escrito, em que ambas as partes não tenham contato visual, é muito diferente de um diálogo em que ambos estejam mantendo tal contato, pois a partir de determinados gestos a conversa pode mudar o seu rumo e dar novos significados e interpretação por parte do profissional.

Dessa forma, visto a deficiência existente no diálogo entre um ser humano e um computador, em que pode ser apresentado falhas pelo chatbot, devido a sua não análise contextual da emoção do usuário naquele momento, como a computação afetiva pode auxiliar o diálogo entre máquina e homem?

Como o presente trabalho tem o objetivo desenvolver uma inteligência artificial que dialogue e analise as expressões durante uma conversa, com o intuito de melhorar a eficácia da interação entre um humano e um computador, deixando-a mais

contextualizada de acordo com a emoção. Pode-se definir alguns objetivos específicos como: desenvolver uma I.A para reconhecer macro expressões, desenvolver um chatbot que possui inteligência artificial, unir os dois aprendizados para melhor eficiência da resposta ao usuário: analisar os resultados dessa I.A. com a computação afetiva.

As pesquisas desenvolvidas neste trabalho foram divididas nas vertentes formal e experimental e, com isso, a metodologia teve de ser adequada à vertente empregada. Utilizou-se o método hipotético-dedutivo para a vertente formal. Esta metodologia procura uma solução através do estabelecimento de conjecturas, hipóteses e teorias, e realização de testes de falseamento para eliminação de erros. Ela se desenvolve através de raciocínios e trabalha do geral para o particular, levando o pesquisador do conhecido para o desconhecido.

As vertentes formal e experimental são um processo cíclico que evolui à medida em que as hipóteses são reformuladas e as técnicas para realizar observações e experimentos melhoram. (MARCONE; LAKATOS, 2003). Para a vertente experimental, foi utilizada a metodologia incremental-iterativa. Nessa abordagem, várias partes do problema são desenvolvidas em paralelo, integradas quando completas e reelaboradas quando necessário, até que o objetivo seja atingido. É uma estratégia de planejamento escalonado onde o tempo de revisão e melhorias de partes do problema é pré-definido (PRESSMAN; SOMMERVILLE, 2002, 2003).

Este trabalho está organizado da seguinte forma: O capítulo 2 apresenta conceitos e definições para a compreensão do escopo do trabalho; O capítulo 3 apresenta as ferramentas utilizadas e os códigos do desenvolvimento da I.A. No capítulo 4 são apresentados os experimentos, resultados obtidos e as principais dificuldades encontradas. O capítulo 5 conclui o trabalho, apresentando as contribuições e no capítulo 6 é apresentado os trabalhos futuros.

2- FUNDAMENTAÇÃO TEÓRICA

2.1 - Inteligência Artificial

Conforme Dias (2017), Inteligência Artificial, ou I.A, é uma inteligência que é capaz de ser produzida artificialmente, o que deixa em aberto para que essa inteligência será usada, ou seja, sua capacidade de adaptação é enorme, podendo realizar várias tarefas que os humanos fazem, sendo ela especialista ou básica. Ela ganhou força no século XXI com a invenção dos computadores modernos.

A I.A teve seu o início do seu desenvolvimento após a segunda guerra mundial, em que um artigo publicado pelo matemático Alan Turing, *Computing Machinery and Intelligence*, despertou o interesse da comunidade científica por máquinas inteligentes. Entretanto, foi somente com o advento das máquinas modernas que a Inteligência Artificial, ganhou formas de se estabelecer como uma ciência, visto que existiam algoritmos, mas não existiam hardwares capazes de executar tais algoritmos (BURKE; ISLA; DOWNIE, 2001).

Com o avanço dos computadores, com hardwares mais potentes, a computação se tornou uma área diversificada e campos como a visão computacional, análise e síntese de voz e a computação afetiva, ganharam espaço e seu próprio nicho de pesquisa, e hoje pode-se dizer que cada uma dessas áreas são um pequeno mundo de ideias promissoras.

A inteligência artificial possui vários ramos que podem ser seguidos, como por exemplo ao treinar uma determinada I.A a partir do machine learning, teremos que escolher entre um tipo de aprendizagem supervisionada e não supervisionada, sendo que na aprendizagem supervisionada, teremos uma base de dados que possui o padrão a se reconhecer (a resposta da I.A, será baseada nesse padrão), e que será utilizada para o treinamento da I.A. Já no aprendizado não supervisionado é realizado uma busca por itens ou elementos semelhantes, ou seja, o padrão é estabelecido durante o treinamento da I.A sobre a base de dados (FERREIRA, 2021).

É importante citar tais campos, pois eles são responsáveis por estudar o comportamento humano durante o uso de uma tecnologia, e é pelos poderes dessas áreas que a computação afetiva pode atuar estudando o comportamento humano na interação com o computador.

2.2 - Tensorflow e funções nativas

Com a crescente da inteligência artificial algumas ferramentas foram implementadas para apoiar o desenvolvedor em algumas tarefas repetitivas ou que já possuem um padrão estabelecido de desenvolvimento. Uma das principais bibliotecas de I.A atualmente, é o tensorflow, que inclusive foi utilizado para o treinamento da I.A e que possui uma gama de ferramentas. Como diz Zeng (2019),

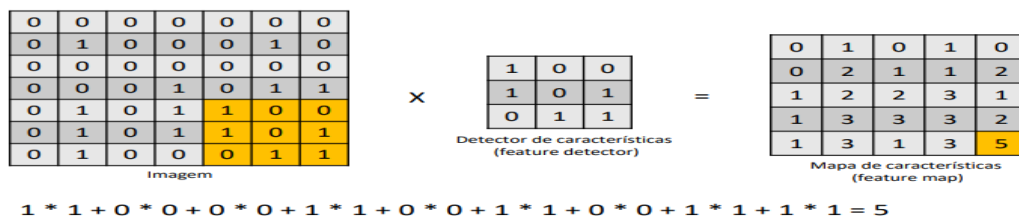
...O Tensorflow é uma plataforma completa de código aberto para machine learning.

Ele tem um ecossistema abrangente e flexível de ferramentas, bibliotecas e recursos da comunidade que permite aos pesquisadores levar adiante machine learning de última geração e aos desenvolvedores criar e implantar aplicativos com tecnologia de machine learning. O tensorflow fornece de forma nativa algumas funções que são muito importantes, nos próximos tópicos será explicado o funcionamento de algumas dessa funções.

2.2.1 – Mapa de características

O mapa de características ou feature map, conforme Bullinaria (2005), é um tipo de saída dos atributos de entrada que possibilita um número muito menor de dimensões (melhorando dessa forma o desempenho de processamento), ou seja, ele consegue entregar uma representação discreta dos dados de entrada, é possível ver uma demonstração do feature map na figura 1.

Figura 1 - Demonstração do mapa de características.



Fonte: livro - An introduction to neural computing

2.2.2 – Kernel size e regularizer

O kernel size, segundo DE CASTRO (1999) é o tamanho da matriz que se usa gerar o feature map, nesse caso como está destacado em laranja na figura 1, utiliza-se um kernel size de 3x3. Já o kernel regularizer é responsável por fazer a “penalização” da I.A, ou seja, ele identifica o erro, e aumenta o seu valor conforme o erro for sendo identificado ao decorrer do treinamento.

2.2.3 - Função de ativação e padding

A função de ativação (activation), chamada “relu”, responsável por avaliar o retorno do mapa de características, caso o valor retornado seja menor do que 0, ele é usado para zerar o valor, e caso contrário ele passar o valor para frente. Essa função de ativação é importante pois ela será usada para identificar as principais partes da imagem, e realça-las, dessa forma otimizando a identificação da emoção. Já a função de padding utilizando o parâmetro “same”, é responsável por preservar algumas características menos importantes da imagem, dessa forma não possibilitando que a imagem perca tantas informações.

2.3 - Arquitetura CNNs

Para que a I.A fosse treinada com o objetivo de reconhecer emoções, foi utilizado uma arquitetura do tipo CNNs (Redes Neurais Convolucionais).

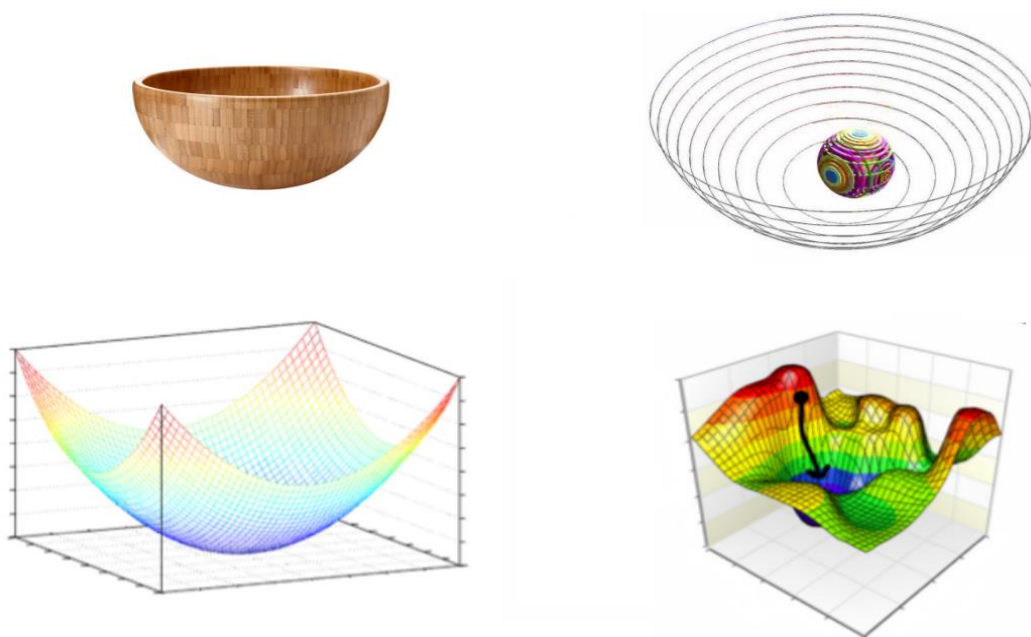
As Redes Neurais Convolucionais (ConvNets ou CNNs) são redes neurais artificiais profundas que podem ser usadas para classificar imagens, agrupá-las por similaridade (busca de fotos) e realizar reconhecimento de objetos dentro de cenas. São algoritmos que podem identificar rostos, indivíduos, sinais de rua, cenouras, animais e muitos outros aspectos dos dados visuais (AGHDAM; HERAVI, 2018).

As redes convolucionais realizam o reconhecimento óptico de caracteres (OCR), para digitalizar texto e tornar possível o processamento de linguagem natural em documentos analógicos e manuscritos, onde as imagens são símbolos a serem transcritos. CNNs também podem ser aplicadas a arquivos de áudio quando estes são representados visualmente como um espectrograma. Mais recentemente, as redes convolucionais foram aplicadas diretamente à análise de texto, bem como dados gráficos (BIZOTTO; LOUISE, 2021).

Para se pensar em redes neurais convolucionais, deve-se saber que ela não percebe imagens como seres humanos, ou seja, quando uma imagem é processada

por uma rede convolucional, a rede irá perceber a imagem não se baseando em altura e largura, mas sim como volumes conforme a figura 2, ou seja, objetos tridimensionais, isso acontece, pois, as imagens possuem uma codificação RGB (vermelho, verde e azul), e ao misturar essas três cores pode-se obter o espectro que os seres humanos percebem. A rede convolucional irá receber imagens como três estratos separados de cores empilhados um em cima do outro (AGHDAM; HERAVI, 2018).

Figura 2 - Representação da forma como a CNN irá enxergar uma imagem.



Fonte: Próprio autor

2.4 - Computação afetiva

Como diz Ma (2021),

...a computação afetiva é um campo da Inteligência Artificial que tem como propósito entender as emoções e o pensamento humano durante o uso de um sistema, e utilizar desse entendimento para algum fim, uma dessas explicações para tal entendimento é conhecer o próprio modo de pensar humano, resultando dessa forma em um desenvolvimento tanto da I.A, como também de ciências que estudam as faculdades mentais.

Entre essa relação entre o modo de pensar humano e a lógica da I.A, se encaixa o campo que estuda as emoções envolvidas do usuário ao utilizar um determinado sistema, que se chama computação afetiva. A computação afetiva além

de estudar como as expressões emitidas pelo usuário podem influenciar no uso do sistema, também irá levar isso em conta no desenvolvimento de um software baseado nesse campo de estudo (PICARD, 1997).

Portanto tais softwares baseados na computação afetiva serão capazes de produzir sistemas computacionais capazes de interagir com o usuário e dialogar de forma inteligente levando em conta seus estados emocionais ao longo da sessão de uso do software, dessa forma tornando a experiência com o software mais completa.

Alguns exemplos podem ser citados como aplicações recentes da computação afetiva e que representam essa experiência mais sofisticada da máquina com o homem, como é o caso do trabalho feito por Jaques, Nunes, Isotani e Bittencourt (2012). Nesse trabalho é relatado a importância de tutores inteligentes, que sejam capazes de reconhecer emoções e utiliza-las para um melhor aprendizado do aluno/usuário, sendo uma aplicação que utiliza dos fundamentos da computação afetiva para melhorar a eficiência de um determinado seguimento da sociedade que é o ensino e como é ressaltado no artigo, tal ferramenta melhoraria principalmente o ensino a distância, em que o professor não está a todo tempo junto ao aluno, para saber por exemplo quando ele está deprimido por não ter aprendido algum assunto.

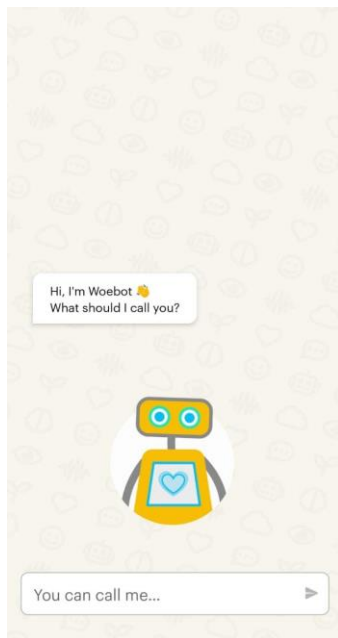
2.5 - Chatbot

O chatbot é um software desenvolvido com o intuito de dialogar por meio de linguagem escrita ou falada com o usuário de forma automatizada, dessa forma ele tem o objetivo de passar a sensação de que o usuário está dialogando com um ser humano. O nível de "inteligência" entre os chatbots varia muito. Enquanto alguns chatbots têm um entendimento bastante básico da linguagem, outros empregam algoritmos sofisticados de inteligência artificial (IA) e aprendizado de máquina (Machine Learning), para alcançar um nível de conversação quase humano (MCTEAR, 2020).

Essa aplicação visa interagir com o usuário de forma mais sofisticada. Como diz Cabot (2021), o nome chat (bate-papo) + bot (robô), da origem aos chatbot's que são robôs de conversa, e entre suas aplicações estão por exemplo os assistentes virtuais, que possibilitam as empresas conversarem com seus clientes 24 horas por dia, 7 dias por semana, os exemplos de sua atuação podem ser sanando dúvidas rápidas, agendando consultas, confirmando entregas, além de outras funcionalidades.

Além disso podemos ter chatbot's de respostas simples e objetivas que não envolvem inteligência artificial, como também pode-se ter chatbot's capazes de interagir de forma inteligente e inclusive aprender durante esse processo de conversa com o usuário, estes são mais ideais para tarefas mais sofisticadas, como é o exemplo do woebot mostrado na figura 3, que é um robô de conversação que tenta ajudar usuários com problemas relacionados a baixo estima por exemplo.

Figura 3 - Interface inicial do woebot.



Fonte: Aplicativo do woebot disponível no play store e app store

2.6 - Interação Humano Computador

Pode-se dizer que a interação é algo que advém dos primórdios da humanidade, sempre existiu a necessidade de se interagir, seja através de desenhos em rochas ou formas geométricas, o ser humano sempre buscou formas de viver em sociedade e se comunicar de alguma forma. Como tudo no mundo, a comunicação evoluiu e alcançou um patamar que não envolve mais somente dois humanos, mas pode ocorrer entre máquinas e homens, ou, até mesmo entre máquinas e máquinas, e é a partir dessa ideia de comunicação entre homem e máquina que surgiu a interação humano computador (IHC), que visa estudar tal comunicação, abordando várias outras áreas que irão se relacionar com a ciência da computação, como ergonomia, design, artes, semiótica, psicologia, sociologia entre outras áreas relacionadas (BENYON, 2015).

Como dizia Picard (1997),

...caso máquinas (incluindo softwares e hardwares), fossem capazes de reconhecer, expressar e desenvolver emoções, seria possível emular a vida como um todo, a partir de robôs, tal desejo é muito antigo pelo ser humano, entender a si, para que se possa replicar em máquinas cada vez mais inteligentes, que possam interagir da melhor forma com usuários e tornar a experiência humano-computador, cada vez mais sofisticada e acessível.

A partir desse estudo da IHC, é visado resultados que proporcionem maximizar as habilidades de quem usa a máquina, tornando o uso de dispositivos tecnológicos (que possuam uma interface), mais simples, agradáveis e seguros (BARBOSA; SILVA, 2010), tornando os softwares mais sofisticados, pois ele é pensado tanto em questão de desempenho como em questão de usabilidade.

2.7 - Visão computacional

A visão computacional é uma área da inteligência artificial que visa estudar e implementar um sistema que seja capaz de simular a visão humana. O sistema munido dessa capacidade, irá interpretar um conjunto de dados e esses dados normalmente carregam muitos ruídos provenientes de fatores como o local de captura da imagem, equipamentos utilizados, objetos em foco, entre outros. O processamento de imagens visa a adequação necessária para atenuar essas variações. Para tanto, como a imagem é vista pelo computador como uma função, existem inúmeras técnicas matemáticas para modelar os dados conforme o necessário, uma das muitas bibliotecas de desenvolvimento utilizadas para isso é a OpenCV que se destaca por sua robustez de funções e velocidade de processamento (KOVASNAY, 1955).

A biblioteca OpenCV, é utilizada para processamento de imagem e visão computacional e é uma ferramenta de código aberto, dessa forma possibilitando personalização de suas funções e código fonte. Por conta da sua alta gama de funções, que funcionam de forma muito eficaz, e o fato dela ser de código aberto, acaba a tornando a biblioteca favorita de desenvolvedores do campo de visão computacional (ESCRIVÁ, 2019).

3- IMPLEMENTAÇÃO E DESENVOLVIMENTO

3.1- Equipamentos e softwares

O hardware utilizado para implementação e treinamento da I.A. será um notebook com processador Intel Core i7 de oitava geração, 8gb de RAM e 1TB de armazenamento. Para o treinamento da I.A. será utilizada uma câmera com boa qualidade em ambientes claros ou escuros. Como os ambientes em que o usuário pode utilizar a I.A. são muito variados será utilizado a câmera ZED, uma câmera estéreo de alta qualidade, que possibilitou uma maior versatilidade de ambientes em que a I.A seria capaz de reconhecer as expressões.

A I.A. de visão computacional será treinada utilizando uma arquitetura de rede neural convolucional conhecida como CNN (Convolutional Neural Network). Segundo Demirci (2018), as redes neurais convolucionais são inspiradas nos processos biológicos, nesse tipo de arquitetura a conectividade entre os neurônios é inspirada na organização do córtex visual dos animais, ou seja, existem neurônios específicos que irão responder a estímulos apenas em uma região restrita do campo de visão conhecida como campo receptivo. A CNN demanda um nível de pré-processamento menor quando comparado a outros tipos de algoritmos de classificação de imagens, ou seja, ela “aprende” os filtros que teriam que ser implementados manualmente, que é uma das suas grandes vantagens, pois necessita de um menor conhecimento a priori e também menor esforço humano no desenvolvimento de suas funcionalidades básicas.

O software utilizado para escrever o algoritmo será o Pycharm, que possui uma interface simples e completa, possibilitando que o desenvolvedor possa fazer download de bibliotecas diretamente da sua interface e também é conhecido por ser uma das melhores IDE'S (Integrated Development Environment ou Ambiente de Desenvolvimento Integrado), para desenvolvimento utilizando a linguagem Python.

A linguagem utilizada para o desenvolvimento do algoritmo será o Python 3.6, pois esta é de fácil integração com as bibliotecas da I.A. como a Tensorflow da Google, além disso o Python é reconhecido atualmente como sendo a melhor linguagem para desenvolvimento de algoritmos voltados para ciência, pois possui baixa curva de aprendizado focando suas funções na maior produtividade do desenvolvedor.

O *dataset* com as imagens das expressões que será utilizada para treinar a rede neural deste projeto foi fornecida de forma pública pela Kaggle, comunidade online da Google de cientistas de dados, que promove constantemente desafios que possam utilizar redes neurais de forma prática para resolver problemas propostos pela comunidade ou que a sociedade vive no momento. (FER2013)

3.2- Código da arquitetura e explicação

Nesta seção será mostrado os principais trechos do código, utilizado na arquitetura do modelo que foi utilizada para o treinamento da I.A, e a explicação do seu funcionamento.

Figura 4 - Definição das variáveis da arquitetura da I.A

```
num_features = 64
num_labels = 7
batch_size = 64
epochs = 100
width, height = 48, 48

model = Sequential()
```

Fonte: Próprio autor

No código mostrado na figura 4, é feita a definição das variáveis e a inicialização do modelo. O **num_features** é o número de características que foi definido em 64, logo teremos 64 mapa de características, o **num_labels** que é referente a quantidade de classes de emoções que iremos trabalhar, no caso 7 (raiva, nojo, medo, neutro, feliz, triste, surpreso), o **batch_size** serve para indicar de quantos em quantos registros irá ser feito a atualização dos pesos na rede neural, a **epochs** que irá dizer por quantas épocas irá ser executado o algoritmo, no caso 100 e por último, o **width** e o **height** que é a largura e a altura que as imagens serão processadas. Após a definição das variáveis iniciamos nosso modelo, gerando uma sequência de camadas através da função **Sequential()**.

3.2.1 – Código parte 2

No código demonstrado na figura 5 será possível observar algumas das funções e variáveis citadas anteriormente, e também a adição de mais camadas no treinamento da I.A.

Figura 5 - Adição de camadas na arquitetura da I.A.

```
model.add(Conv2D(num_features, kernel_size=(3,3), activation='relu',
                input_shape=(width, height, 1),
                kernel_regularizer = l2(0.01)))
model.add(Conv2D(num_features, kernel_size=(3,3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))
model.add(Dropout(0.5))

model.add(Conv2D(2*num_features, kernel_size=(3,3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Conv2D(2*num_features, kernel_size=(3,3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))
model.add(Dropout(0.5))

model.add(Conv2D(2*2*num_features, kernel_size=(3,3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Conv2D(2*2*num_features, kernel_size=(3,3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))
model.add(Dropout(0.5))

model.add(Conv2D(2*2*2*num_features, kernel_size=(3,3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Conv2D(2*2*2*num_features, kernel_size=(3,3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))
model.add(Dropout(0.5))

model.add(Flatten())

model.add(Dense(2*2*2*num_features, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(2*2*num_features, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(2*num_features, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(num_labels, activation = 'softmax'))

model.summary()
```

Fonte: Próprio autor

3.3 – Processamento de imagem e código

O processamento das imagens, assim como funções de visão computacional e carregamento do classificador em cascatas, foram feitos com o auxílio da biblioteca OpenCV. No código que pode ser vista na figura 6, é mostrado algumas definições iniciais, como a variável condição, que estabelece se o algoritmo irá entrar no loop de reconhecimento de emoções que irá ser mostrado em um trecho posterior de código.

Também pode-se ver definições de diretórios para simplificar onde irá ser encontrado materiais importantes para o processo de reconhecimento, como o classificador, que é o arquivo gerado após o treinamento sobre a base de dados, que é responsável por ter matrizes que representam os padrões da base de dados em

que ele foi treinado, também é definido o caminho do modelo de arquitetura, é iniciado a captura de vídeo e definição do tamanho da face que irá ser processado pelo algoritmo na captura do vídeo.

Por último tem-se a variável “imagemFace”, que é a que diz para o algoritmo se alguma face foi detectada, para que caso não tenha sido detectado nenhuma face, ele não faça um processamento de verificação da emoção, dessa forma aumentando a eficiência do algoritmo.

Figura 6 – Início do código do arquivo que detecta a emoção.

```
def run(self):  
    condicao = True  
    diretorio = "../MaterialInteligenciaArt/"  
    classificador = "D:\\Program Files\\pycharmprojects\\IC\\MaterialInteligenciaArt\\haarcascade_frontalface_default.xml"  
    face_detection = cv2.CascadeClassifier(classificador)  
    caminho_modelo = "D:\\Program Files\\pycharmprojects\\IC\\MaterialInteligenciaArt\\modelo_02_expressoes.h5"  
  
    camera = cv2.VideoCapture(0)  
  
    largura, altura = 200, 200 # tamanho que ficará a imagem da face  
    imagemFace = False
```

Fonte: Próprio autor

Já neste outro trecho de código exibido na figura 7, é definido as emoções que irão ser detectadas, além de personalização de como a face irá ser processada na variável faces. Já na variável cinza, a imagem irá receber uma transformação de cor para cinza, isso pois a cor cinza, como cor única, facilita o reconhecimento de um determinado padrão, visto que a diversidade de cores, pode acabar confundindo a I.A durante o treinamento. Após isso é verificado se alguma face foi detectada, no trecho de código onde tem o comando “for”, caso tenha sido detectado alguma face, é feito uma série de processamentos sobre a imagem da face detectada, incluindo o seu redimensionamento (irá ficar no tamanho 48x48), para que ela possa ficar dentro do mesmo padrão de dimensão das imagens que o classificador foi treinado. Se nenhuma face é detectada, o algoritmo retorna “nenhuma face detectada”.

Figura 7 – Definição das expressões e algumas funções do OpenCV.

```
expressoes = ["Raiva", "Nojo", "Medo", "Feliz", "Triste", "Surpreso", "Neutro"]

maiorEmoc = 0
#emocPred = ""

original = imagemFace.copy()
faces = face_detection.detectMultiScale(original, scaleFactor=1.1, minNeighbors=3, minSize=(20, 20))
cinza = cv2.cvtColor(original, cv2.COLOR_BGR2GRAY)

if len(faces) > 0:
    for (fX, fY, fW, fH) in faces:
        roi = cinza[fY:fY + fH, fX:fX + fW]
        roi = cv2.resize(roi, (48, 48))
        roi = roi.astype("float") / 255.0
        roi = img_to_array(roi)
        roi = np.expand_dims(roi, axis=0)
        preds = classificador_emocoes.predict(roi)[0]
        #print(preds)
        emotion_probability = np.max(preds)
        label = expressoes[preds.argmax()]
        cv2.putText(original, label, (fX, fY - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.65, (0, 0, 255), 2, cv2.LINE_AA)
        cv2.rectangle(original, (fX, fY), (fX + fW, fY + fH), (0, 0, 255), 2)
else:
    print('Nenhuma face detectada')
```

Fonte: Próprio autor

Já no final do algoritmo que está representado na figura 8 é feita a verificação da quantidade de faces encontradas e só irá executar as partes posteriores se a quantidade de faces for 1.

Após isso é feito um **for** que percorre cada emoção atribuindo a ela o resultado de um cálculo de probabilidade que sinaliza o quanto aquela emoção pode ser a emoção predominante da imagem.

Dentro de cada iteração do **for**, é verificado no **if** se a variável “maiorEmoc” possui a emoção com maior probabilidade na imagem até o momento, essa variável é inicializada com o valor zero, e ao longo das iterações ela é substituída com o valor da probabilidade (que foi definida como “prob”), da emoção que está dentro da variável “emotion”, caso esse valor de “prob” seja maior que o valor presente em “maiorEmoc” no momento da iteração, dessa forma é sempre guardado na variável “maiorEmoc”, o valor de probabilidade da emoção predominante naquela iteração.

Caso o **if** seja adentrado, é guardado na variável “emocPred”, a emoção que possuiu maior probabilidade até o momento, caso não entre no **if** o valor de “maiorEmoc” e de “emocPred” continuam os mesmos. Após isso é utilizado duas funções do OpenCV, uma para desenhar uma retângulo envolta da face detectada e outra para montar a “barra” de probabilidade de cada emoção no gráfico construído.

Figura 8 – Cálculo da probabilidade e atribuição da emoção de maior predominância a uma variável.

```
if len(faces) == 1:
    for (i, (emotion, prob)) in enumerate(zip(expressoes, preds)):
        # Nome das emoções
        text = "{}: {:.2f}%".format(emotion, prob * 100)
        w = int(prob * 300)
        if (maiorEmoc < (prob * 100)):
            maiorEmoc = prob * 100
            self.emocPred = emotion
            self.emocpred2 = emotion
            #print(emotion + " entrou na emocao")

        cv2.rectangle(probabilidades, (7, (i * 35) + 5),
                      (w, (i * 35) + 35), (200, 250, 20), -1)
        cv2.putText(probabilidades, text, (10, (i * 35) + 23),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.45,
                    (0, 0, 0), 1, cv2.LINE_AA)

time.sleep(1)
```

Fonte: Próprio autor

4- EXPERIMENTOS E RESULTADOS

4.1 – Resultado dos experimentos da visão computacional

Algumas estatísticas podem ser realizadas, sobre a base de dados FER2013, pode ser visto na figura 9, demonstra a quantidade de imagens disponíveis de cada emoção, onde a emoção nojo é a que possui menor quantidade, logo será a que terá menor taxa de acerto também.

Figura 9 - Gráfico da quantidade de imagens para cada tipo de emoção

- 0: 4593 imagens- *Angry (Raiva)*
- 1: 547 imagens- *Disgust (Nojo)*
- 2: 5121 imagens- *Fear (Medo)*
- 3: 8989 imagens- *Happy (Feliz)*
- 4: 6077 imagens- *Sad (Triste)*
- 5: 4002 imagens- *Surprise (Surpresa)*
- 6: 6198 imagens- *Neutral (Neutro)*



Fonte: Kaggle

Os experimentos relatados foram feitos com imagens do conjunto FER2013 e foram submetidos ao algoritmo desenvolvido, onde as barras representam a quantidade de cada emoção que foi detectada nos gráficos apresentados. Todas as imagens exibidas, apresentam a assertividade da I.A em relação a emoção, isso é representado através da maior porcentagem da emoção que é detectada.

A emoção **Felicidade** obteve uma grande porcentagem, a maior entre os testes exibidos, que pode ser observado na figura 10, isso aconteceu devido a grande quantidade de dados disponíveis (a maior entre as 7 emoções, possuindo 8989 imagens), para treinamento de reconhecimento dessa expressão, e isso deixa claro que, quanto maior a quantidade de imagens disponíveis para treinamento, melhor será a assertividade da I.A. Além disso, a imagem também possui uma boa reação da expressão por parte da pessoa que é exibida na imagem.

Figura 10 - Detecção da emoção feliz, em uma imagem da base de dados FER2013.



Fonte: Próprio autor

A emoção **Medo** que é observada na figura 11 apresenta uma maior distribuição entre os palpites da I.A se comparado a emoção feliz da figura 10 por exemplo, o que é justificável também pela posição que essa emoção se encontra, quando é usado como base a quantidade de imagens, a emoção medo se configura na quarta posição das emoções com mais imagens, possuindo 5121 imagens. A segunda maior porcentagem dessa imagem é a emoção surpresa, o que é justificável devido as semelhanças que podem existir entre essas emoções.

Figura 11 - Detecção da emoção medo, em uma imagem da base de dados FER2013.

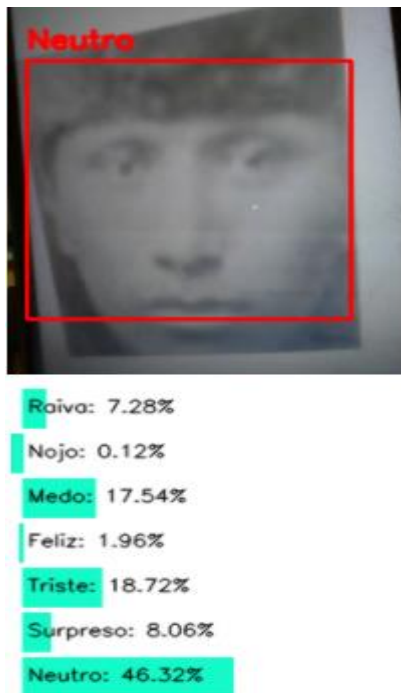


Fonte: Próprio autor

Já a emoção **Neutro** mostrada na figura 12 apresenta a maior porcentagem, possuindo na base de dados de treinamento, 6198 imagens, se comparado a emoção medo da figura 11. Podemos ver que as disparidades entre as porcentagens de cada emoção na figura 12 quando comparado a emoção neutro, são grandes, pois a porcentagem da emoção neutro se sobressai bastante sobre as outras emoções inclusive sobre a emoção que teve a segunda maior porcentagem que foi a emoção triste.

Já na figura 11 que possui a maior porcentagem na emoção medo, demonstrou uma disparidade diferente em questão de assertividade, comparando a emoção predominante medo da figura 11, com a segunda mais predominante surpreso também da figura 11, pode-se ver que a emoção surpreso se aproxima um pouco mais da porcentagem da emoção predominante medo, isso contribui com o ponto de que a quantidade das imagens, influencia diretamente na assertividade, mesmo quando a diferença entre a base de dados de cada emoção não é tão grande (diferença de 1077 imagens), como foi este caso, pois a emoção neutro, possui uma maior base de dados para treinamento e logo apresentou uma melhor acurácia (maior diferença na porcentagem entre a emoção de maior predominância comparado com a segunda emoção de maior predominância).

Figura 12 - Detecção da emoção neutro, em uma imagem da base de dados FER2013.



Fonte: Próprio autor

A emoção **Nojo** é detectada com uma boa assertividade, mesmo possuindo a menor base de dados entre todas as emoções (547 imagens), mas isso acontece devido a uma segunda e terceira variável que influencia no processo, que é a intensidade que a expressão está sendo exibida, e também a quantidade de imagens de determinada emoção que apresentam tal intensidade demonstrada na figura 13, o ideal para um bom treinamento, seria ter diversidade nesse ponto, possuindo vários tipos de intensidade de uma mesma emoção.

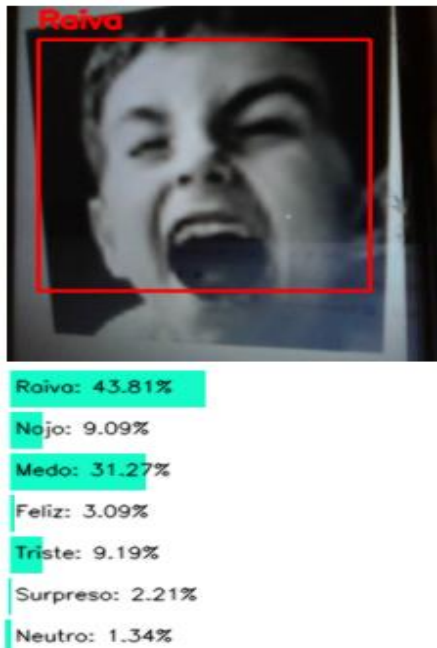
Figura 13 - Detecção da emoção nojo, em uma imagem da base de dados FER2013.



Fonte: Próprio autor

A emoção de **Raiva** possui 4593 imagens na base de dados é detectada de forma predominante e pode ser vista na figura 14. Existe uma segunda grande predominância no medo, o que pode ser explicado pela relação que ambas as emoções possuem e também pela quantidade de imagens na base de dados serem parecidas, o mesmo não aconteceu na emoção medo da figura 11, pois a expressão possui mais dados que a expressão raiva, além das variáveis de intensidade da emoção estarem mais bem distribuídas na categoria medo, logo mesmo a emoção medo e raiva possuindo similaridades, esses outros pontos influenciaram para que a I.A tivesse mais certeza na hora de fazer inferências.

Figura 14 - Detecção da emoção raiva, em uma imagem da base de dados FER2013.



Fonte: Próprio autor

A emoção **Surpresa** possui 4002 imagens na base treinamento que pode ser observada na figura 15, obteve uma boa assertividade, não demonstrando muita dúvida da I.A em relação a qual expressão está sendo exibida. A expressão medo acaba aparecendo em algumas delas, como a segunda maior porcentagem, isso está ocorrendo devido a diversidade de intensidade de imagens na base de dados da emoção medo, e isso é um ponto a se questionar quanto a um melhor refinamento dessa variável intensidade, para que isso não atrapalhe na acurácia da detecção de cada expressão.

Cada base deve possuir uma diversidade nem muito abrangente, que acabe possibilitando afetar a detecção de outras expressões com melhor acurácia, e também não deve possuir uma base de dados “viciada”, para que não ocorra de uma expressão só ser detectada quando ela for exibida exatamente como é mostrada nas figuras do treinamento.

Figura 15 - Detecção da emoção surpresa, em uma imagem da base de dados FER2013.



Fonte: Próprio autor

A emoção **Triste** mostrado na figura 16 entre todos os testes apresentados nesse trabalho, possuiu a 2º melhor assertividade da emoção detectada. A sua posição em relação a quantidade de imagens na base de treinamento é a 3º maior base de dados, com 6077 imagens. A emoção triste apresentou uma grande diferença de assertividade na porcentagem em relação a emoção neutro da figura 12, mesmo as duas bases possuindo uma quantidade semelhante de imagens, isso ressalta novamente a questão da diversidade de imagens de cada base e as variantes da emoção que estão presentes em cada uma, sendo que na figura 16, a emoção triste possui melhor diversidade de intensidade da emoção e mais variantes, dessa forma, apresentando melhores resultados.

Figura 16 - Detecção da emoção triste, em uma imagem da base de dados FER2013.



Fonte: Próprio autor

4.2 – Construção do chatbot e integração da I.A

O chatbot foi construído com o objetivo de consumir a visão computacional da I.A em forma de API (Application Programming Interface), dessa forma as macros expressões são analisadas pela parte de visão computacional a cada 1 segundo e retornadas para o chatbot para que ele possa responder tendo “consciência” da emoção do usuário naquele momento.

Uma instância não treinada do chatbot começa sem conhecimento de como se comunicar. Cada vez que o usuário entra com uma afirmação, a biblioteca salva o texto que foi inserido e em seguida a afirmação foi respondida. Conforme o chatbot recebe mais entradas o número de respostas que ele pode responder e a precisão de suas respostas em relação a afirmação de entrada cresce. O programa seleciona a resposta mais precisa procurando pela resposta mais próxima que combina com a afirmação de entrada, então ele retorna a resposta mais provável para a afirmação baseada na frequência que esta resposta é emitida pelo usuário que está se comunicando com o bot.

Dessa forma, a precisão das respostas aumenta de acordo com a quantidade de comunicação com o chatbot, e essa precisão também será influenciada pela emoção do usuário, pois a resposta pode ser modificada de acordo com cada emoção a depender da forma como foi configurada o chatbot. Essa implementação de como

o chatbot irá responder, pode ser modificada e personalizada para cada tipo de aplicação que ela pode ser utilizada.

Nesse trabalho não foi especificado uma aplicação final para o chatbot, ou seja, a aplicação faz o reconhecimento das expressões faciais, integra isso com um chatbot que consome as expressões, como se tivesse consumindo uma API, e retorna uma resposta personalizada para o usuário, mas sem um determinado fim, sendo que essa resposta fica mais precisa, quanto mais se comunica com o chatbot.

5 - CONCLUSÃO

A interação de áreas do conhecimento com a inteligência artificial, permite que os resultados obtidos a partir de aplicações sejam mais sólidos e eficientes, e este foi o propósito desse trabalho. Pois ao interagir a visão computacional com o chatbot, é possível obter resultados melhores no que tange a eficiência da comunicação e contextualização da mesma, pois assim as respostas serão personalizadas de acordo com a emoção que o usuário sente naquele momento.

Fatores externos podem influenciar no não cumprimento do objetivo definido para um chatbot, como por exemplo a venda de um plano ou o não cancelamento do mesmo, esse fator externo pode ser a emoção, pois um usuário pode aderir ou não a uma oferta a depender de como ele é tratado e qual sua emoção no atual momento da conversa.

A emoção pode influenciar também em um chatbot que somente quer dialogar para melhorar o dia de alguém, pois somente o diálogo sem levar em conta, o contexto emocional daquela conversa, nem sempre vai resolver.

Partindo desse princípio, construiu-se o “HUE”, que é uma inteligência artificial integrada com um chatbot, capaz de dialogar reconhecendo os sentimentos através das macro expressões exibidas pelo usuário ao longo do diálogo, permitindo uma melhor contextualização conversacional e personalização de respostas, visto que o software foi construído de forma a consumir a API de visão computacional, deixando para quem for utilizar a livre personalização de qual será o objetivo a se alcançar pelo chatbot, já que ele irá aprender de acordo com a frequência de conversas ou a base de dados utilizada para treinamento do chatbot.

O software construído segue o modelo de desenvolvimento de licenciamento livre open-source, o que torna o conhecimento sobre eles abertos, sendo possível a sua continuidade, modificação e expansão.

Após a construção do software, testes foram aplicados sobre a I.A de visão computacional para que fosse possível mensurar sua precisão sobre cada emoção detectada e também sobre as emoções como um todo. Os resultados obtidos foram satisfatórios, sendo que a I.A obteve uma acurácia de 62,44% de precisão ao detectar uma emoção, o que está dentro dos parâmetros de reconhecimento de expressão facial que foram definidos pela Kaggle em uma de suas competições.

A I.A também mostrou ótimos resultados em imagens mais escuras e em diferentes ângulos e isso foi possível devido à base de dados possuir uma grande diversidade de imagens e a arquitetura ter uma ótima construção, definindo parâmetros para que fosse reduzido a quantidade de erros da I.A.

As principais dificuldades no desenvolvimento desse projeto foram:

- Diversidade de ângulos, cores e luz das imagens obtidas
 - A diversidade de características em que uma imagem pode estar foi um fator problemático para o desenvolvimento do software, pois para que isso não seja um fator que piore a acurácia da I.A, foi necessário também ter uma base de dados bastante diversa.
- Intensidade das emoções detectadas
 - As emoções podem ser detectadas em diferentes intensidades, mesmo que seja a mesma emoção, as intensidades dificultam o reconhecimento, sendo que as emoções que apresentam uma maior intensidade serão mais simples de serem reconhecidas, ao contrário das que possuem menos.
- Multidisciplinaridade
 - Para que esse software fosse desenvolvido foi necessário conhecimentos teóricos e práticos de diversas áreas como inteligência artificial, computação afetiva, interação humano-computador, psicologia, sociologia, teleatendimento, programação concorrente, aprendizado de máquina, redes neurais, banco de dados, robótica e arquitetura de computadores.

Dentro das propostas do início do projeto os objetivos gerais e específicos foram atingidos e obtiveram-se os resultados esperados, fomentando o futuro desse trabalho e o que ainda pode ser desenvolvido. Os resultados deixam expectativas otimistas para dar continuidade ao projeto com alguns aprimoramentos.

6- TRABALHOS FUTUROS

Nessa seção são mostradas algumas sugestões para o aperfeiçoamento do sistema proposto.

- Reconhecer as emoções também através do texto e da voz, dessa forma aumentando a acurácia do reconhecimento, pois irá utilizar várias fontes de reconhecimento e elas irão se validar.
- Construir um sintetizador de voz para a I.A, dessa forma deixando-a mais humanizada e encurtando os laços com o usuário.
- Construir uma face para a I.A, de forma que ela possa mover a cabeça e lábios de acordo com a resposta que ela irá dar.
- Personalizar a face da I.A, para que ela expresse emoções com o seu “rosto”, de forma condizente com a sua resposta dada para o usuário.
- Fornecer uma base de dados prontas e diversas para o treinamento do chatbot, isso irá diminuir o trabalho de treinamento de quem irá utilizar a ferramenta, dessa forma ao fazer uma personalização na I.A, ela poderá ser mais específica.
- Embarcar o sistema em algum dispositivo.
- Fornecer o sistema em formato de aplicativo.

REFERÊNCIAS

AGHDAM, H.H; HERAVI, E.J. **Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification**. Springer, ago. 2018.

BARBOSA, I. A. **Cuidado de enfermagem por Telessaúde: qual a influência da distância na comunicação?**. Rev. Bras. Enferm, Brasília, v. 70, n. 5, p. 928-934, out. 2017.

BARBOSA, S.D.J.; Silva, B.S. **Interação Humano-Computador**. Série SBC, Editora Campus-Elsevier, Rio de Janeiro, jan. 2010.

BENYON, D. **Interação Humano-Computador**. Pearson, Londres, 2º edição, p. 46-74, out. 2015;

BIZOTTO, F.; LOUISE, M.S.C.D.A. **Systematic Mapping on Convolutional Neural Networks Applied in Air Image Segmentation**. EasyChair Preprint, v. 6870, out. 2021.

BULINARIA, J. A. **Introduction to Neural Networks**. Self Organizing Maps: Fundamentals. Birmingham, fev. 2005, Disponível em <https://www.cs.bham.ac.uk/~jxb/NN/l16.pdf> Acesso em: 15 mar. 2021.

BURKE, R.; ISLA, D.; DOWNIE, M.; IVANOV, Y.; BLUMBERG, B. **Creature Smarts: The Art and Architecture of a Virtual Brain**. Synthetic Character Group, MIT Media Laboratory, Massachusetts, out. 2001.

CABOT, J. **The Software Challenges of Building Smart Chatbots**. IEEE *Xplore*, Barcelona, mai. 2021, Disponível em <https://ieeexplore.ieee.org/document/9402515> Acesso em: 22 jun. 2021.

DE CASTRO, L. N. & VON ZUBEN, F. J. **Neural Networks with Adaptive Activation Functions: A Second Order Approach**, Proc. do SCI/ISAS'99, p. 574-581, mar. 1999.

DEMİRÇİ, H. M. **User experience over time with conversational agents: case study of woebot on supporting subjective wellbeing.** Middle East Technical University, Ankara-Eskişehir, p. 135-144, set. 2018.

DIAS, C. R. **Qual a relação entre Deep Learning e Machine Learning?**, IGTI, São Paulo, p.10-15, nov. 2017.

ESCRIVÁ, D.M. **OpenCV 4 Computer Vision Application Programming Cookbook: Build complex computer vision applications with OpenCV and C++.** Packt Publishing, Montreal, p. 30 – 63, mai. 2019.

FERREIRA, A.C.P.D.L. **Inteligência Artificial – Uma Abordagem de Aprendizado de Máquina.** LTC, São Paulo, p.70-100, mar. 2021.

G. Kovasznay, L., & Joseph, H. **Image Processing. Proceedings of the IRE**, 43(5), 560–570. doi:10.1109/jrproc.1955.278100. fev. 1955.

JAQUES, P.A.; NUNES, M.A.S.N.; ISOTANI, S.; BITTENCOURT L.G. **Computação Afetiva aplicada à Educação: Dotando Sistemas Tutores Inteligentes de Habilidades Sociais.** Desafie, jul. 2012.

MA, H; YAROSH S. **A Review of Affective Computing Research Based on Function-Component-Representation Framework.** IEEE *Xplore*, Minneapolis, ago. 2021. Disponível em <https://ieeexplore.ieee.org/document/9512296> Acesso em: 29 jul. 2021.

MARCONE, M. d. A.; LAKATOS, E. M. **Fundamentos da Metodologia Científica.** 5o edição. São Paulo: Editora Atlas SA, 2003.

MCTEAR, M. **Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots.** Morgan & Claypool, San Rafael, out. 2020.

PICARD, Rosalind. **Affective computing.** MIT Press, Massachusetts Ave, Cambridge, p. 217-233, out. 1997

PRESSMAN, R. S. **Engenharia de Software**. [S.l.]: McGraw Hill Brasil, 2002.

SILVA, T. T. A.; SILVA, N.P.; ANDRADE, A.O. **System for Recognition of Facial Expressions Using Machine Learning**. IEEE *Xplore*, Natal, jan. 2020. Disponível em <https://ieeexplore.ieee.org/document/9307127> Acesso em: 27 jul. 2021.

SOMMERVILLE, I. **Engenharia de software**. ed. Adison Wesley, 6a. Edição, 2003.

ZENG Z; Gong Q.; Zhang J. **CNN Model Design of Gesture Recognition Based on Tensorflow Framework**. IEEE *Xplore*, Xi'an, mar. 2019. Disponível em <https://ieeexplore.ieee.org/document/8729185> Acesso em: 25 jul. 2021.