



Trabalho Prático

Inteligência Artificial – GCC128

Ana Beatriz Rodrigues Torres - 201920241

Beatriz Fernandes Teixeira - 202110220

Felipe Carvalho Godoi - 201920237

Lavras - MG

2023

Ana Beatriz Rodrigues Torres - 201920241
Beatriz Fernandes Teixeira - 202110220
Felipe Carvalho Godoi - 201920237

Trabalho Prático

Inteligência Artificial – GCC128

Trabalho apresentado à disciplina Inteligência Artificial – GCC128, como parte das exigências apresentadas no plano de curso.

Lavras - MG

2023

Sumário

1. Introdução	4
2. Arquitetura	5
2.1 Protótipo da idealização do robô	5
2.2 Estrutura do código fonte	5
2.3 Banco de imagens	6
3. Implementação	7
3.1 Recursos utilizados	7
3.2 Aquisição de dados e imagens	7
3.3 Algoritmos utilizados	7
3.4 Treinamento do Modelo de Reconhecimento	7
3.5 Testes e Validação	7
4. Resultados	8
4.1 Desempenho do Robô	8
5. Conclusão	9
5.1 Desafios	9
5.2 Futuras Melhorias	9
6. Referências	10

1. Introdução

Este relatório apresenta um projeto de desenvolvimento de um Robô Social com Interação Humana, baseado no Modelo Arquitetônico Sentir-Pensar-Agir. O objetivo fundamental deste projeto de Inteligência Artificial (IA) é a criação de um robô social autônomo e natural, capaz de interagir de forma significativa com seres humanos em ambientes do mundo real.

O projeto se baseia em uma estrutura sólida, dividida em três componentes essenciais: Percepção Sensorial, Processamento de Dados, Tomada de Decisão e Resposta, além de focar na interatividade entre o robô e os usuários. A integração desses elementos é fundamental para alcançar a capacidade desejada de comunicação e entendimento do robô ao longo do tempo.

O projeto tem como objetivo atender às necessidades de pessoas com deficiência visual, oferecendo um assistente robótico capaz de auxiliá-las na identificação de produtos em supermercados. Para atingir esse propósito, o robô foi projetado para responder a comandos de voz e direcionar sua câmera para os produtos desejados, fornecendo informações detalhadas sobre esses itens, tornando a experiência de compra mais acessível e informativa para esse público.

Neste relatório, faremos uma explanação minuciosa das etapas do projeto, desde a concepção da arquitetura até a implementação, enfrentando e superando desafios técnicos e operacionais, e, por fim, apresentando os resultados conquistados. A avaliação deste projeto se concentrará na eficácia do robô em interagir naturalmente com seres humanos e na qualidade da implementação técnica, bem como na capacidade de documentação e apresentação dos alunos.

2. Arquitetura

2.1 Protótipo da idealização do robô



2.2 Estrutura do código fonte

O código foi separado pelo Sentir-Pensar-Agir utilizando os diretórios de Model-View-Controller (MVC).

Model (Modelo): responsável por representar os dados e a lógica de negócios do sistema. É responsável por armazenar informações sensoriais, ou seja, os dados da câmera e os dados de áudio. Nele também contém o reconhecimento de objetos, processamento da fala. Por fim, é feita a tomada de decisão com base nas informações dos sensores e na interpretação da linguagem humana

View (Visualização): responsável por apresentar as informações ao usuário e receber suas interações. Inclui elementos de interface do usuário, no caso, um sistema de comandos de voz, que permite aos usuários interagirem com o robô. Também é responsável por emitir as respostas do robô.

Controller (Controlador): atua como uma camada intermediária entre o Modelo e a View, controlando o fluxo de dados e lógica de negócios. Ele recebe os comandos de voz e as imagens da câmera. Assim, com base nos comandos recebidos, o Controlador chama os

métodos no Modelo para realizar ações, como identificar produtos ou processar comandos de voz, refletindo as mudanças na View.

2.3 Banco de imagens

Para o projeto de reconhecimento de produtos do supermercado, foi necessário criar um banco de dados de imagens para treinar e alimentar um modelo de reconhecimento de objetos. Esse banco de dados de imagens servirá como base de dados para o algoritmo de visão computacional do robô.

A construção do banco de dados foi feita a partir de imagens obtidas pela webcam do computador de forma a capturar todos os ângulos e detalhes dos produtos bolacha, livro, um cubo mágico, chiclete Trident e pipoca.

3. Implementação

3.1 Recursos utilizados

1. TensorFlow e Keras:

Uso no Model: A biblioteca TensorFlow, juntamente com a API Keras, foi utilizada na camada do Model. É responsável pelo carregamento e predição do modelo de aprendizado de máquina (`ia.py`). O Model faz predições com base nas imagens capturadas pela câmera.

2. GPT-3.5-turbo (GPT-3):

Uso na camada Model: O GPT-3 foi utilizado para expandir o dataset de interações do robô (`dataset.py`). Ele é chamado pelo Model para gerar respostas dinâmicas e adaptáveis, fornecendo uma funcionalidade adicional ao conjunto de dados.

3. OpenCV (Visão):

Uso na **camada View**: A biblioteca OpenCV foi usada para a captura e exibição de vídeo. A classe `Vision` interage diretamente com a câmera para obter os quadros de vídeo, redimensioná-los e exibi-los em uma janela.

4. SpeechRecognition (Áudio):

Uso na camada View: A biblioteca SpeechRecognition é utilizada na View para converter áudio em texto. A classe `Audio` possui métodos para aceitar comandos de voz, permitindo a interação intuitiva do usuário com o robô.

5. gTTS (Google Text-to-Speech) (Áudio):

Uso na camada View: A biblioteca gTTS é utilizada na View para converter texto em fala. É empregada pela classe `Audio` para permitir que o robô responda verbalmente às consultas e comandos do usuário.

3.2 Aquisição de dados e imagens

A aquisição das imagens para treinamento visando popular a base de dados da IA foi utilizada uma webcam e a plataforma Teachable Machine. Cinco diferentes produtos foram escolhidos para compor esse conjunto de dados: um pacote de pipoca, um chiclete, um pacote de biscoito, um cubo mágico e um livro.

Por meio da webcam, capturaram-se uma série de imagens representativas de cada um desses produtos em diferentes ângulos e iluminações. Esse processo permitiu a obtenção de uma variedade de perspectivas, enriquecendo a capacidade da IA em reconhecer esses objetos em ambientes variados.

Adicionalmente, o Teachable Machine foi empregado para capturar fotos específicas para treinamento. Esta plataforma facilitou a criação de um conjunto de dados equilibrado e enriquecido com exemplos adicionais, garantindo uma maior abrangência na identificação dos produtos pelo sistema. Os dados adquiridos foram processados e utilizados para treinar o modelo de IA, que foi salvo em um arquivo denominado "keras_model.h5". Esse arquivo encapsula o aprendizado do modelo, permitindo sua posterior utilização para inferência em tempo real. Além disso, um arquivo denominado "labels.txt" foi criado para mapear as classes associadas aos diferentes produtos. Esse arquivo contém os rótulos correspondentes a cada categoria, o que facilita a interpretação das previsões feitas pelo modelo.

3.3 Algoritmos utilizados

O algoritmo utilizado para produzir o modelo de reconhecimento de produtos (salvo como `keras_model.h5`) é um modelo de aprendizado de máquina baseado em redes neurais convolucionais (CNNs). Mais especificamente, a biblioteca Keras com backend TensorFlow foi empregada para construir e treinar esse modelo. A arquitetura da rede neural é essencialmente composta por várias camadas convolucionais e de pooling, que são responsáveis por extrair características relevantes das imagens. As camadas convolucionais aplicam um conjunto de filtros a uma entrada para extrair características relevantes. Esses filtros são aprendidos durante o treinamento da rede e ajudam a identificar padrões visuais complexos. Em seguida, as características extraídas são alimentadas em camadas densas para realizar a classificação final dos produtos.

3.4 Treinamento do Modelo de Reconhecimento

O treinamento da rede neural do Teachable Machine utiliza o algoritmo de otimização de descida de gradiente estocástico (SGD) com backpropagation, e a função de perda utilizada é a entropia cruzada categórica (categorical cross-entropy), adequada para problemas de classificação multi-classe.

Os parâmetros de aprendizado de máquina utilizados para treinar o modelo de reconhecimento foram tamanho do *Batch* e *Epochs*. O primeiro (cujo valor utilizado foi 16) refere-se à quantidade de exemplos de treinamento que são processados de uma só vez durante uma iteração do processo. O segundo Epochs se refere ao número de vezes que o conjunto completo de dados de treinamento é passado pela rede neural durante o processo de treinamento. Cada *epoch* permite que o modelo ajuste seus pesos com base nos gradientes calculados a partir da diferença entre as previsões do modelo e os rótulos reais. Por exemplo, se definirmos 10 *epochs*, significa que o conjunto de treinamento será utilizado 10 vezes para atualizar os pesos da rede. Cada passagem por todos os dados é uma época.

3.5 Testes e Validação

- **Teste da Webcam e Captura de Imagens:**

A funcionalidade de captura de imagens pela webcam foi testada e validada para garantir que o sistema fosse capaz de obter imagens nítidas e bem definidas dos produtos em diferentes cenários.

- **Avaliação do Reconhecimento de Produtos:**

Foram realizados testes específicos para cada categoria de produto, onde o sistema foi exposto a vários cenários para as classes bolacha, livro, cubo mágico, trident e pipoca. O reconhecimento foi avaliado quanto à precisão da resposta

- **Integração com o Chat GPT:**

O sistema foi testado em cenários de diálogo com o usuário. Foram feitas solicitações para identificar produtos e, em seguida, o Chat GPT gerou respostas corretas na maioria das vezes com base nas categorias reconhecidas.

4. Resultados

4.1 Desempenho do Robô

O desempenho do robô no reconhecimento de produtos foi satisfatório, demonstrando uma taxa de acerto consistente durante os testes. A integração da webcam e o modelo de aprendizado de máquina permitiram identificar com precisão os produtos pré-definidos, como bolacha, livro, cubo mágico, trident e pipoca. No entanto, é importante destacar que há margem para melhorias adicionais. Uma expansão do conjunto de dados de treinamento e o refinamento dos algoritmos de processamento de imagem poderiam aprimorar ainda mais o desempenho do robô.

5. Conclusão

5.1 Desafios

Durante o desenvolvimento e teste do projeto, observamos que o Chat GPT 3.5 por vezes apresentou respostas que não correspondiam inteiramente às solicitações dos usuários de modo a provocar uma certa confusão no fluxo da conversa e do programa. Isso destacou a necessidade de uma integração mais avançada e precisa para proporcionar uma experiência ainda mais refinada.

5.2 Futuras Melhorias

Para futuras iterações do projeto, recomenda-se a expansão do conjunto de dados de treinamento e a implementação do Chat GPT 4. O primeiro para aprimorar ainda mais a precisão do reconhecimento e o segundo a fim de melhorar a experiência do usuário na interação com o Chat GPT. Além disso, a incorporação de feedback do usuário mais detalhada a partir do reconhecimento de um produto como busca de preços em tempo real e em supermercados próximos pode ser considerada para ampliar a gama de funcionalidades do robô.

6. Referências

<https://www.tensorflow.org/about/bib?hl=pt-br>
https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
<https://keras.io/api/>
<https://gtts.readthedocs.io/en/latest/>