UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

Carlos Gabriel Farias Da Silva

Análise de desempenho do Gemini na Estimativa de Peso de Alimentos por Imagem

Recife

Agosto de 2025

Carlos Gabriel Farias Da Silva

Análise de desempenho do Gemini na Estimativa de Peso de Alimentos por Imagem

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE
Departamento de Estatística e Informática
Curso de Bacharelado em Sistemas de Informação

Orientador: Cícero Garrozi

Recife
Agosto de 2025

Análise de desempenho do Gemini na Estimativa de Peso de Alimentos por Imagem

Carlos Gabriel [1], **Cícero Garrozi** [1]

[1]Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco
Rua Dom Manuel de Medeiros, s/n, - CEP: 52171-900 – Recife – PE – Brasil

carlos.gabrielsilva@ufrpe.br, cicero.garrozi@ufrpe.br@ufrpe.br

Resumo. *Com o avanço das inteligências artificiais multimodais, cresce o interesse em sua aplicação na área da saúde para facilitar a análise nutricional e auxiliar no combate à obesidade. No entanto, a confiabilidade desses modelos para identificar alimentos e estimar porções a partir de imagens ainda é incerta, sendo fundamental mensurar seu desempenho de forma objetiva.*
Este trabalho avalia a capacidade do modelo Gemini de classificar ingredientes e estimar seus respectivos pesos (em gramas) a partir de fotografias de refeições. Para isso, foi desenvolvido um sistema automatizado que envia requisições à API do Gemini, utilizando um prompt textual padronizado, elaborado com técnicas de engenharia de prompt, e uma lista de ingredientes de referência. As respostas do modelo, obtidas em formato JSON, foram comparadas com dados reais para análise de desempenho.
Os resultados obtidos nos experimentos indicaram um baixo desempenho geral. Na classificação de ingredientes, o modelo apresentou baixa precisão e sensibilidade (recall), com dificuldade em detectar itens como temperos e condimentos (por exemplo, azeite e sal) que estavam misturados a outros alimentos, embora tenha obtido altas taxas de acerto para ingredientes visualmente distintos, como morangos e ovos mexidos. Na estimativa de peso, o desempenho também foi insatisfatório, com altos valores de erro (MAE e RMSE) e coeficiente de determinação (R^2) negativo, evidenciando tendência à superestimação e desempenho inferior a uma simples predição pela média.

Abstract. *With the rapid development of multimodal artificial intelligence systems, there is growing interest in their application in healthcare to support nutritional analysis and help combat obesity. However, the reliability of these models in identifying foods and estimating portion sizes from images remains uncertain, making it essential to objectively assess their performance.*
This study evaluates the ability of the Gemini model to classify ingredients and estimate their respective weights (in grams) from meal photographs. An automated system was developed to send requests to the Gemini API, using a standardized text prompt, designed with prompt engineering techniques, along with a reference list of ingredients. The model's responses, provided in JSON format, were compared against ground truth data to assess performance.
The results showed generally low performance. In ingredient classification, the model exhibited low precision and recall, struggling to detect items such as condiments and seasonings (e.g., olive oil and salt) mixed with other foods, although it achieved high detection rates for visually distinct ingredients such

as strawberries and scrambled eggs. In weight estimation, performance was also poor, with high error values (MAE and RMSE) and a negative coefficient of determination (R^2), indicating a tendency to overestimate weights and performance worse than a simple mean-based prediction.

1. Introdução

Nos últimos anos, a inteligência artificial (IA) tem experimentado um crescimento técnico acelerado, especialmente com o surgimento e popularização de ferramentas generativas, como o ChatGPT, Gemini, DeepSeek e Grok. Esse avanço tem impulsionado a incorporação da IA em múltiplas esferas da vida cotidiana, no ambiente de trabalho, à educação, passando pelo entretenimento e pela saúde [Ipsos 2024], com modelos cada vez mais multimodais, ou seja, capazes não apenas de gerar textos, mas também de criar e analisar imagens, vídeos e áudios.

O relatório da [Ipsos 2024] indica que 66% das pessoas acreditam que a IA mudará profundamente suas vidas nos próximos três a cinco anos, e metade sente que isso já ocorreu. Além disso, cresce a percepção de que a IA afetará significativamente o mundo do trabalho, com 60% prevendo mudanças na forma como desempenham suas funções e 36% temendo que seus empregos sejam substituídos por máquinas.

Um dos setores que mais pode se beneficiar dessa tecnologia é o da saúde pública, em especial no que diz respeito à nutrição e ao controle da obesidade. Projeções indicam que, até 2030, quase 3 bilhões de adultos, cerca de 50% da população adulta mundial, viverão com alto índice de massa corporal (IMC), sendo o Brasil um dos países afetados, com estimativa de mais de 119 milhões de adultos nessa condição [World Obesity Federation 2025].

Nesse contexto, a utilização da IA generativa pode ser aplicada como uma das estratégias para melhorar a alimentação da população, ao facilitar a identificação e contagem calórica das refeições. Tal abordagem contribui para uma redução da carga cognitiva aos métodos tradicionais (como o registro manual do consumo diário em aplicações web ou mobile) consequentemente aumentando as chances de sucesso na adaptação de novos regimes alimentares.

No entanto, a confiabilidade desses modelos para tarefas como a identificação de alimentos e a estimativa precisa de porções ainda é incerta. Para que essas ferramentas sejam efetivamente utilizadas em escala, é fundamental compreender suas limitações e mensurar, de forma objetiva, seu desempenho.

1.1. Formulação do problema

Neste trabalho, o problema consiste em avaliar a capacidade do modelo Gemini de, a partir de uma imagem I representando uma refeição composta por K ingredientes, identificar corretamente cada ingrediente c_k pertencente a um conjunto C de possíveis ingredientes, bem como estimar o peso w_k (em gramas) de cada um deles. A identificação correta de um ingrediente presente na imagem é considerada um Verdadeiro Positivo (TP); a falha em detectar um ingrediente presente, um Falso Negativo (FN); e a identificação de um ingrediente ausente, um Falso Positivo (FP).

Podemos definir matematicamente o problema como:

$$f_{Gemini}(I) = \{(c_k, w_k)\}_{k=1}^{K} \quad | \quad c_k \in \mathcal{C}, w_k \in \mathbb{R}^+ \tag{1}$$

1.2. Objetivos

O principal objetivo deste trabalho é avaliar, de forma quantitativa e qualitativa, o desempenho do modelo de inteligência artificial generativa Gemini, desenvolvido pelo Google, na tarefa de estimar o peso (em gramas) dos ingredientes presentes em imagens de refeições. A avaliação é conduzida em duas etapas: a primeira consiste na verificação da correta identificação dos ingredientes; a segunda, na estimativa precisa de seus respectivos pesos.

1.2.1. Objetivos específicos

Para atingir o objetivo proposto, são definidos os seguintes objetivos específicos:

1. Selecionar um conjunto de dados com imagens reais de pratos contendo anotações dos ingredientes e seus respectivos pesos.
2. Elaborar o prompt a ser enviado ao modelo LLM, seguindo princípios de engenharia de prompt.
3. Automatizar o envio de requisições e o salvamento das respostas do modelo para posterior agregação.
4. Calcular métricas de desempenho, incluindo métricas de regressão (como erro médio absoluto e raiz do erro quadrático médio) e métricas de classificação (como precisão e *recall*), com base nas respostas fornecidas pelo modelo.
5. Validar os resultados obtidos por meio de análise estatística e comparação com os dados reais.

1.3. Abordagem proposta

Neste trabalho, propomos como abordagem o uso do modelo de IA generativa Gemini, acessado por meio de uma API, para estimar o peso e os ingredientes de refeições a partir de imagens. A proposta envolve o envio automático das imagens e de um arquivo com informações sobre todos os ingredientes ao modelo, com o objetivo de obter suas previsões. Os resultados serão organizados e analisados estatisticamente, com foco na avaliação da precisão das estimativas fornecidas pelo modelo.

1.4. Contribuições

Este trabalho contribui com a análise da capacidade do modelo Gemini em realizar estimativas de ingredientes e dos respectivos pesos presentes em fotografias de refeições. A partir dessa análise, buscamos compreender as limitações do modelo e o grau de confiabilidade de suas respostas, considerando seu uso em aplicações como assistentes virtuais e sistemas nutricionais automatizados baseados em inteligência artificial.

1.5. Organização do trabalho

O artigo está organizado nas seguintes seções:

- Seção 2 – Referencial Teórico: Apresenta os principais conceitos e fundamentos relacionados ao tema do trabalho, incluindo inteligência artificial e modelos generativos.

- Seção 3 – Trabalhos Relacionados: Revisa os estudos mais relevantes e recentes da literatura, com foco em abordagens semelhantes para estimativa de alimentos.
- Seção 4 – Abordagem Proposta: Detalha os métodos, ferramentas e estratégias adotadas, incluindo a estrutura do sistema implementado e o uso da API do modelo Gemini.
- Seção 5 – Experimentos: Descreve o passo a passo da execução experimental, incluindo a preparação dos dados, definição dos prompts e critérios de avaliação.
- Seção 6 – Resultados: Apresenta os resultados obtidos, análises estatísticas e interpretações dos desempenhos observados.
- Seção 7 – Conclusão: Resume as principais descobertas, discute as limitações do estudo e propõe possíveis direções para trabalhos futuros.

2. Referencial Teórico

Esta seção estabelece as bases conceituais para a compreensão do tema. Inicialmente, são abordados os conceitos de Inteligência Artificial e Visão Computacional, seguidos pelos paradigmas de aprendizado de máquina, como regressão e classificação. Posteriormente, apresentam-se as métricas estatísticas para a avaliação de modelos e, por fim, exploram-se os Modelos de Linguagem de Grande Escala (LLMs), detalhando suas extensões multimodais e o papel da Engenharia de Prompts na interação com esses sistemas.

2.1. Inteligência Artificial (IA)

A IA é um vasto campo da ciência da computação e engenharia que busca construir mecanismos que simulem a capacidade humana de pensar e tomar decisões. Seu propósito é desenvolver e empregar máquinas para realizar atividades humanas de maneira autônoma, percebendo variáveis, raciocinando, aprendendo e resolvendo problemas [Norvig and Russell 2014].

A IA abrange uma enorme variedade de subcampos, que vão desde o geral, como aprendizado e percepção, até tarefas específicas, como jogar xadrez, demonstrar teoremas matemáticos, criar poesia e diagnosticar doenças. Uma abordagem moderna da IA foca na ideia de agentes inteligentes, definindo a IA como o estudo de agentes que recebem percepções do ambiente e executam ações. Esses agentes implementam funções que mapeiam sequências de percepções em ações, podendo ser representadas por sistemas de produção, agentes reativos, planejadores condicionais em tempo real, redes neurais e sistemas de teoria da decisão. A concepção de um agente inteligente ideal, ou seja, aquele que adota a melhor ação possível em uma situação, é o foco principal. A aprendizagem é um aspecto crucial que permite aos agentes operar em ambientes inicialmente desconhecidos e melhorar sua competência [Norvig and Russell 2014].

2.1.1. Visão Computacional

A Visão Computacional é um ramo da inteligência artificial que capacita máquinas a "enxergar" e interpretar o mundo ao seu redor, extraindo informações significativas a partir de imagens capturadas por diversos dispositivos, como câmeras de vídeo, sensores e scanners. O objetivo é permitir que computadores reconheçam, manipulem e processem informações sobre os objetos em uma imagem, simulando e aproximando-se da inteligência humana [de Milano and Honorato 2014].

A organização de um sistema de Visão Computacional não segue um modelo padrão rígido, pois muitas aplicações são desenvolvidas como sistemas especialistas para resolver problemas específicos. Contudo, a maioria desses sistemas envolve o reconhecimento de objetos em imagens e a transformação dessas informações para processamento posterior. As funcionalidades comuns incluem a aquisição de imagem, pré-processamento, extração de características, detecção e segmentação e processamento de alto nível [de Milano and Honorato 2014].

2.1.2. Métricas de Avaliação

A avaliação do desempenho de modelos de aprendizagem é fundamental para determinar a eficácia de um algoritmo e sua capacidade de generalizar para dados não vistos. Um dos principais desafios é garantir que a hipótese produzida pelo algoritmo irá prever o valor correto para novas entradas [Norvig and Russell 2014].

Diversas métricas e conceitos são utilizados para a avaliação do desempenho do modelo:

- **Precisão (Precision)**: Mede a proporção de previsões positivas corretas em relação ao total de previsões positivas realizadas, sendo dada por $\frac{TP}{TP+FP}$ [Mariano 2021].
- **Curvas de Aprendizagem**: Representam a evolução do desempenho do modelo conforme aumenta o tamanho do conjunto de treinamento, permitindo avaliar se o modelo está sofrendo de *underfitting* ou *overfitting* [Norvig and Russell 2014].
- **Funções de Perda**: Quantificam o erro entre os valores previstos e os reais [Norvig and Russell 2014]. As principais utilizadas foram:
  - Erro Absoluto Médio (MAE): $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
  - Raiz do Erro Quadrático Médio (RMSE): $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$

  .
- **Sensibilidade (Recall)**: Avalia a capacidade do modelo de identificar corretamente os casos positivos, sendo dada por $\frac{TP}{TP+FN}$ [Mariano 2021].
- **F1-Score**: Corresponde à média harmônica entre precisão e sensibilidade, especialmente útil em cenários com classes desbalanceadas. É calculado como $2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}}$ [Mariano 2021].
- **Coeficiente de Determinação (R^2)**: Mede o grau de ajuste do modelo aos dados observados, sendo calculado por $1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}i)^2}{\sum i=1^{n}(y_i - \bar{y})^2}$. Seus valores variam de 0 a 1, indicando a proporção da variabilidade dos dados que é explicada pelo modelo, quanto mais próximo de 1 melhor o ajuste [Molnar 2025]. No entanto, se o modelo não possuir um valor inicial constante e apresentar desempenho inferior ao de um modelo que simplesmente prevê a média da variável de resposta, o valor de R^2 pode ser negativo [Barten 1987].

2.2. Modelos de Linguagem de Grande Escala (LLMs)

Large Language Models (LLMs), são sistemas avançados de IA que se destacam por sua capacidade de processar e gerar texto de forma coerente. Eles representam um avanço significativo na área de processamento de linguagem natural (NLP), evoluindo de modelos de linguagem estatísticos e neurais, e posteriormente de modelos de linguagem pré-treinados

(PLMs). A característica distintiva dos LLMs é a sua escala massiva, contendo dezenas a centenas de bilhões de parâmetros, e sendo pré-treinados em vastos volumes de dados textuais [Naveed et al. 2025].

Essa escala colossal lhes confere habilidades emergentes que não são observadas em modelos menores. Entre essas habilidades, destacam-se o raciocínio, o planejamento, a tomada de decisões, o aprendizado em contexto e a capacidade de responder a consultas sem exemplos explícitos. A relevância dos LLMs reside na sua aptidão para lidar com tarefas complexas de linguagem, como tradução, sumarização, recuperação de informações e interações conversacionais, aproximando-se do desempenho humano em diversas atividades. [Naveed et al. 2025]

2.2.1. Modelos de Linguagem Grande Multimodais (MLLMs)

Enquanto as LLMs são primariamente projetadas para e treinadas em dados textuais, os modelos de Linguagem Grande Multimodais são sistemas projetados para processar e integrar diversos tipos de dados, incluindo texto, imagens, vídeos, áudio e sequências fisiológicas. Ao combinar e sintetizar informações dessas diferentes modalidades, os MLLMs alcançam uma compreensão e geração de informações mais abrangente e precisa. Eles surgem para abordar a complexidade das aplicações do mundo real, indo muito além das capacidades dos sistemas de modalidade única [Wang et al. 2024].

A arquitetura de um MLLM geralmente pode ser categorizada em três componentes principais: o codificador de entrada multimodal, o mecanismo de fusão de características e o decodificador de saída multimodal. O codificador transforma os dados brutos de várias modalidades em um formato estruturado que o modelo pode processar. O mecanismo de fusão integra as características de diferentes modalidades, o que pode ocorrer em várias fases. Por fim, o decodificador reconverte as informações multimodais integradas de volta para uma forma utilizável, adaptada a tarefas específicas, como geração de legendas de imagens ou resumos de vídeo [Wang et al. 2024].

2.3. Engenharia de Prompt

Prompts podem assumir diferentes formas, desde instruções em linguagem natural que fornecem contexto para orientar a resposta do modelo, até representações vetoriais aprendidas que ativam conhecimentos específicos armazenados no modelo. A engenharia de prompt surgiu como uma técnica essencial para ampliar as capacidades dos LLMs e MLLMs. Trata-se de um processo estratégico de design e refinamento de instruções adaptadas à tarefa, com o objetivo de direcionar a saída do modelo sem a necessidade de modificar seus parâmetros internos [Sahoo et al. 2025].

Um prompt é, essencialmente, uma entrada textual fornecida a um modelo de linguagem com o objetivo de orientar sua resposta. A qualidade desse prompt influencia diretamente a eficácia da saída gerada, o que torna a engenharia de prompt uma competência essencial para pesquisadores, desenvolvedores e profissionais que trabalham com IA tradicional e conversacional. Essa prática oferece uma estrutura sistemática para a criação e documentação de padrões de prompts, permitindo sua adaptação a diferentes domínios e a combinação de estratégias para aprimorar os resultados dos LLMs. A

construção de prompts eficazes envolve etapas como a definição clara do objetivo, a compreensão das capacidades e limitações do modelo, a escolha do formato adequado, o fornecimento de contexto relevante e a realização de testes com refinamentos iterativos [Marvin et al. 2024].

Algumas das técnicas utilizadas no engenharia de prompt são:

- Zero-Shot Prompting: O modelo recebe uma descrição da tarefa no prompt, mas não possui dados rotulados para treinamento. Ele utiliza seu conhecimento pré-existente para gerar previsões [Sahoo et al. 2025].
- Few-Shot Prompting: Fornece ao modelo alguns exemplos de entrada-saída para induzir a compreensão de uma tarefa [Sahoo et al. 2025].
- Chain-of-Thought (CoT) Prompting: Uma técnica que facilita processos de raciocínio coerentes e passo a passo em LLMs. Ela mostra o processo de raciocínio e a resposta final, imitando como os humanos resolvem problemas complexos [Sahoo et al. 2025].
- Contextual Prompting: Fornece contexto adicional ao modelo para ajudá-lo a gerar respostas mais coerentes e relevantes [Marvin et al. 2024].
- Multi-modal Prompting: Utiliza múltiplas modalidades, como texto e imagens, para fornecer prompts mais ricos e detalhados ao modelo de linguagem [Marvin et al. 2024].

3. Trabalhos Relacionados

Considerando a atualidade, relevância metodológica e a contribuição direta ao escopo deste projeto, esta seção apresenta estudos sobre a utilização de inteligência artificial e as diferentes abordagens existentes para o reconhecimento de alimentos e seus valores nutricionais.

3.1. Revisão sistemática de abordagens para estimativa nutricional a partir de imagens de alimentos

O trabalho [Sultana et al. 2023] realiza uma revisão sistemática da literatura com foco em técnicas de aprendizado profundo, comparando-as com algoritmos tradicionais de aprendizado de máquina na estimativa de valores nutricionais a partir de imagens de alimentos. Os autores identificam um fluxo padrão para essa tarefa, composto pelas etapas de segmentação e classificação dos alimentos, estimativa de volume ou peso e, por fim, a estimativa do valor nutricional, conforme ilustrado na Figura 1.

Adicionalmente, o estudo discute os desafios presentes em cada etapa do processo. A segmentação de alimentos apresenta alta complexidade devido a fatores como formas irregulares, oclusão parcial e mistura de ingredientes. Alguns resultados obtidos nessa etapa podem ser observados na Tabela 1. A estimação de volume a partir de fotografias é apontada como um dos maiores obstáculos, sendo exploradas estratégias como o uso de objetos de referência, captura por múltiplas vistas ou utilização de câmeras de profundidade, cada uma com limitações práticas específicas. A Tabela 2 apresenta alguns dos resultados alcançados por diferentes abordagens. Por fim, a estimação nutricional integra os dados de classificação e volume a bancos de dados alimentares, como o USDA (Departamento de Agricultura dos Estados Unidos), permitindo o cálculo do valor calórico e de outros nutrientes.

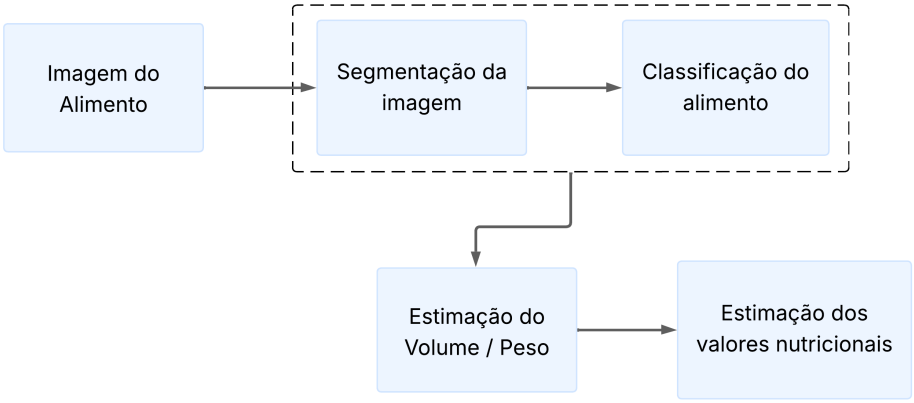

Figura 1. Fluxo geral para a estimativa nutricional de imagens de alimentos, reproduzido de [Sultana et al. 2023].

Os autores concluem que, apesar dos avanços significativos, o campo ainda enfrenta grandes desafios para desenvolver e generalizar modelos mais robustos. Evidenciando a escassez de grandes conjuntos de dados públicos e com anotações para todas as etapas, a dependência de objetos de referência para estimativa do volume, a influência de variáveis visuais (iluminação, ângulo de captura) na acurácia dos modelos e a dificuldade de diferenciação nutricional entre alimentos visualmente semelhantes.

3.2. Utilização do ChatGPT para avaliação nutricional de imagens de alimentos

[O'Hara et al. 2025] realiza uma avaliação da precisão do modelo de linguagem ChatGPT-4 para estimar o conteúdo nutricional de refeições a partir de fotos. O objetivo central era verificar a capacidade do modelo em identificar alimentos, estimar o peso das porções e calcular o valor de 16 nutrientes diferentes.

A metodologia envolveu a análise de 114 imagens de 38 refeições distintas, cada uma apresentada em três tamanhos de porção (pequeno, médio e grande). Além das comparações entre os valores estimados e reais das refeições, para um subconjunto de 38 imagens da porção média, a performance do modelo foi comparada a valores estimados por nutricionistas nos atributos de energia, proteína e carboidratos.

Os resultados indicaram que o ChatGPT-4 teve um bom desempenho na identificação de alimentos, alcançando 93,0% de precisão e um F1-score de 88,6%. Contudo, a estimativa do tamanho das porções mostrou-se precisa apenas para refeições pequenas, com o modelo subestimando significativamente o peso de porções médias e grandes. Apesar da imprecisão nos valores absolutos, observa-se boas correlações indicando que o modelo é eficaz para classificar as refeições de acordo com seu conteúdo nutricional.

[O'Hara et al. 2025] destaca que, embora o ChatGPT-4 demonstre potencial para uso na avaliação dietética, especialmente na identificação de alimentos e na classificação de refeições, ele ainda não possui a precisão necessária para estimar o conteúdo nutricio-

Pesquisador(es) e Ano	Abordagem/Método Utilizado	Descrição Resumida do Método	Resultado na Classificação de Alimentos
Pouladzadeh et al. (2014, *apud* [Sultana et al. 2023])	ML Tradicional (SVM) com características *handcrafted*	Utilizou GraphCut para segmentação e SVM com características visuais (cor, tamanho, forma, textura) para classificar 5 classes de alimentos.	95% de acurácia
Matsuda et al. (2012, *apud* [Sultana et al. 2023])	ML Tradicional (MKL-SVM) com múltiplas características	Empregou detecção de círculo, DPM e segmentação JSEG, usando SIFT, CSIFT, HOG, Gabor texture e cor como características para classificar alimentos.	21% de acurácia (foco principal na segmentação)
Kong et al. (2012, *apud* [Sultana et al. 2023])	ML Tradicional (KNN) com SIFT e Visual Words	Extraiu características SIFT e utilizou *K-means clustering* de Visual Words, aplicando KNN para classificação de 5 classes de alimentos de *dataset in-house*.	92% de acurácia
Tan e Le (2019, *apud* [Sultana et al. 2023])	Deep Learning (EfficientNet)	Implementou a arquitetura EfficientNet para classificação de alimentos no *dataset* de benchmark Food-101 (101.000 imagens de 101 classes).	93% de acurácia
Akhi et al. (2018, *apud* [Sultana et al. 2023])	Híbrido (CNN para extração de *features* + Fisher Vector)	Utilizou arquiteturas CNN pré-treinadas para extração de *features* e codificação com Fisher Vector.	99,13% de acurácia para Bar-Food101 e 95,79% para PFID

Tabela 1. Resumo dos métodos e resultados de classificação de alimentos [Sultana et al. 2023].

nal de forma confiável. Assim, para fins de consulta populacional, ainda há a necessidade de mais desenvolvimento, como o treinamento específico do modelo e sua integração com bancos de dados de composição de alimentos.

4. Abordagem proposta

A abordagem proposta neste trabalho consiste na utilização do modelo Gemini (versão 2.5 Flash), escolhido por sua disponibilidade pública e acessibilidade via API, para estimar o peso de alimentos e identificar ingredientes presentes a partir de imagens.

Para viabilizar a análise em larga escala (em comparação ao processo manual via aplicação web) foi desenvolvido um código em Python que automatiza o envio de requisições à API do Gemini. A cada requisição, são enviados:

- uma imagem do alimento;
- um prompt textual padronizado com instruções específicas ao modelo;
- um arquivo `.csv` contendo uma lista de ingredientes com identificadores únicos.

O conjunto de dados utilizado como base para a extração das imagens empregadas no projeto inclui, adicionalmente, informações detalhadas sobre os ingredientes presentes em cada refeição, as quais foram utilizadas como referência para a avaliação das previsões geradas pelo modelo. Além disso, o conjunto disponibiliza um arquivo contendo a lista completa de ingredientes possíveis, informação que foi incorporada às requisições enviadas à API, para viabilizar a correta comparação das previsões.

O prompt utilizado em todas as requisições é definido no Quadro 1. Para garantir melhores resultados e a padronização das previsões, o prompt foi escrito em inglês e

Pesquisador(es) e Ano	Abordagem/Método Utilizado	Descrição Resumida do Método	Resultado na Estimação de Volume/Peso
Pouladzadeh et al. (2014, *apud* [Sultana et al. 2023])	Objetos de Referência (polegar) e Múltiplas Vistas (superior e lateral)	Utilizou o polegar como objeto de referência e capturou vistas superior e lateral para reconstruir a imagem 3D e estimar o volume.	Erros de estimação de volume variaram entre 1% (melhor caso) e 10% (pior caso) para um *dataset* de alimentos não mistos.
Chae et al. (2011, *apud* [Sultana et al. 2023])	Modelos de Formas Específicas para Alimentos (*Templates*)	Reconstruiu imagens 3D de alimentos a partir de imagens 2D usando *templates* de formas específicas para cada tipo de alimento (ex.: cilindro para bebidas).	Erro relativo médio de volume de 11% para bebidas (17 itens) e 8% para fatias de pão.
Dehais et al. (2017, *apud* [Sultana et al. 2023])	Múltiplas Vistas de Imagens de Alimentos	Propôs a reconstrução de uma imagem 3D de alimento a partir de duas imagens 2D para estimar o volume.	Obteve um Erro Percentual Médio Absoluto (MAPE) entre 8.2% e 9.8% para 45 e 14 pratos em dois *datasets* distintos.
Yang et al. (2019, *apud* [Sultana et al. 2023])	Dados de Sensor de Movimento do Smartphone	Técnica sem marcador fiducial que usou dados do sensor de movimento do smartphone para detectar a orientação da câmera na estimação de volume a partir de imagens 2D.	A estrutura de estimação de volume alcançou um erro absoluto de 16.65% para 10 classes de alimentos.
Rahman et al. (2012, *apud* [Sultana et al. 2023])	Imagens Estéreo de Alimentos	Utilizou imagens estéreo de alimentos para reconstruir imagens 3D, visando estimar o volume de alimentos.	Alcançou um erro médio de 7.7% para seis classes de frutas.

Tabela 2. Resumo de métodos e resultados para estimação de volume/peso de alimentos [Sultana et al. 2023].

aprimorado com o auxílio do próprio Gemini, seguindo práticas de engenharia de prompt como:

- Contextual prompting: instruir o modelo a atuar como um especialista em análise nutricional e reconhecimento de imagens;
- Multi-modal prompting: fornecer simultaneamente a imagem e o arquivo CSV para ampliar a compreensão do modelo;
- Chain-of-thought prompting: decompor a tarefa em etapas sequenciais para facilitar o raciocínio;
- Definição e estruturação do formato de resposta esperado, em JSON;
- Inclusão de regras para casos especiais, definindo a resposta quando o modelo não for capaz de identificar os alimentos presentes na imagem.

Quadro 1: Prompt utilizado

```
You are an expert in nutritional analysis and ingredient
   image recognition. Your task is to analyze the provided
   image of a dish to identify its ingredients and estimate
   their individual weights in grams.

Follow these steps precisely:

1. Identify all visible ingredients present in the dish.
2. For each identified ingredient:
   a. Assign its corresponding ID based on the provided CSV
      reference data.
```

```
   b. Estimate its weight (in grams) using visual cues (such
       as volume) and known density values.
3. Calculate the total estimated weight of the dish by
   summing the weights of all identified ingredients.
4. For each ingredient, and for the overall dish, compute a
   confidence score (in percentage) based on image clarity,
   visibility, and recognition accuracy.
5. If an ingredient cannot be confidently identified, set
   its name to "undefined", its ID to null, its weight to 0,
    and its confidence to 0%.
6. Always include the dish_id: {id} in the output.
7. Return only and exclusively a JSON object with the
   following exact structure:

{
  "dish_id": "dish_id_value",
  "total_weight": "total_weight_in_grams",
  "confidence": "overall_confidence_percentage",
  "ingredients": [
    {
      "id": "ingredient_csv_id",
      "name": "ingredient_name",
      "weight": "weight_in_grams",
      "confidence": "ingredient_confidence_percentage"
    }
  ]
}

Ensure the JSON is syntactically correct and contains no
   additional commentary, metadata or explanations.
```

A resposta da API, recebida em formato JSON, contém as estimativas dos ingredientes identificados, seus respectivos pesos e níveis de confiança. Cada resposta é salva integralmente em um arquivo `.csv` e, posteriormente, analisada individualmente. Os ingredientes previstos e reais são, então, comparados a fim de classificá-los, considerando a categorização TP, FP e FN previamente apresentada.

Cada ingrediente é representado como [Categoria, P_r, P_p], onde P_r é o peso real e P_p o peso previsto. Para FP, define-se $P_r = 0$; para FN, $P_p = 0$. Os dados são salvos em um arquivo `.csv` adicional para análise estatística posterior.

A proposta visa avaliar a precisão do Gemini em tarefas de estimativa visual de alimentos, por meio de métricas como *Root Mean Square Error* (RMSE), Precisão e revocação (*recall*), calculadas com scripts em Python utilizando a biblioteca pandas. O fluxo geral da abordagem pode ser visualizado na Figura 2.

O código-fonte, scripts auxiliares e o prompt utilizado estão disponíveis no repositório: `https://github.com/carlosgabriel311/gemini-food-weight-estimation`.

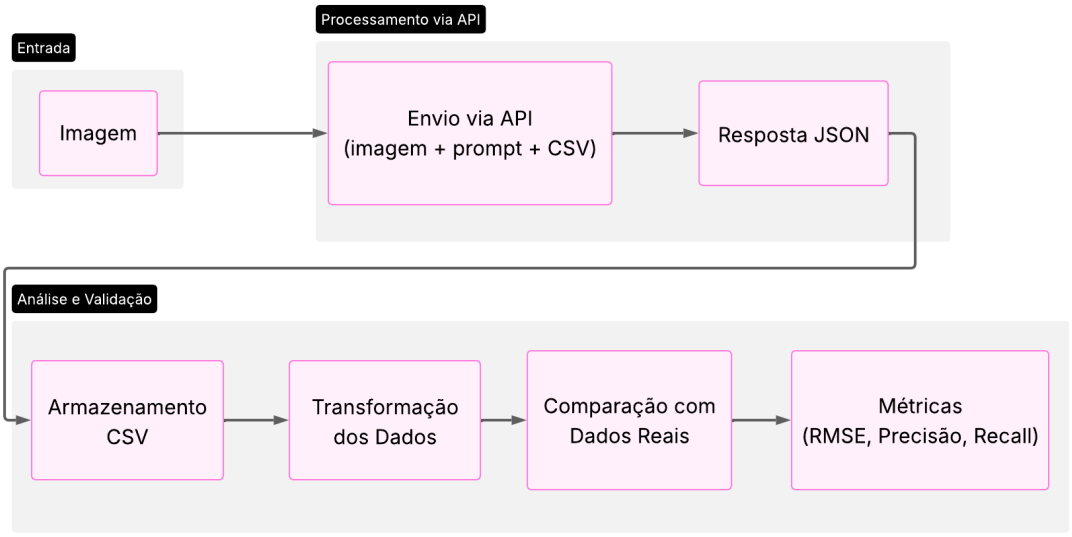

Figura 2. Fluxo de trabalho, Gerado pelo autor.

5. Experimentos

Os experimentos foram conduzidos de maneira a alcançar os objetivos definidos para este projeto. Contudo, devido à limitação da API do Gemini, que permite um máximo de 250 requisições por dia na versão gratuita, os experimentos precisaram ser distribuídos ao longo de vários dias. Essa restrição influenciou a quantidade de imagens analisadas por dia e, consequentemente, o número total de pratos selecionados para a validação do modelo. No total, foram executados 5 testes, seguindo a abordagem proposta e com pratos diferentes do mesmo conjunto de dados, a fim de tornar mais robusta a análise final. Cada etapa é explicada em sua respectiva subseção.

5.1. Conjunto de dados

Para a realização dos experimentos, foi utilizado o conjunto de dados Nutrition5k disponível em `https://github.com/google-research-datasets/Nutrition5k`. O conjunto contém 20 mil vídeos curtos, em 4 diferentes ângulos, de aproximadamente 5 mil pratos servidos nas cafeterias do Google, com 555 classes de ingredientes no total [Thames et al. 2021]. Inclui anotações detalhadas de cada prato, com o peso exato de cada ingrediente e sua composição nutricional (calorias, gorduras, carboidratos e proteínas) obtida a partir do banco de dados USDA. Além disso, cerca de 3.500 pratos possuem dados de profundidade, consistindo em imagens RGB-D capturadas de uma visão superior.

Segundo [Thames et al. 2021], a base de dados foi construída de forma incremental, com cada ingrediente sendo adicionado individualmente ao prato. Após cada adição, o prato era pesado em uma balança de precisão e escaneado por diversos sensores, incluindo câmeras RGB e uma câmera de profundidade Intel RealSense, resultando em múltiplas versões incrementais do mesmo prato.

Embora os dados estivessem originalmente em formato de vídeo, o conjunto disponibiliza alguns scripts úteis para a extração e conversão dos frames em imagens no formato .jpeg, conforme exemplificado no Quadro 2. Considerando a limitação quanto

à quantidade de imagens que poderiam ser utilizadas, optou-se por extrair apenas o primeiro frame de cada vídeo. A Figura 3 apresenta exemplos das imagens finais obtidas a partir da base de dados e utilizadas na execução dos testes.

Quadro 2: Script de extração do frame [Thames et al. 2021]

```
dirs=($(find "./imagery/dataset_unificado" -type d))
for dir in "${dirs[@]:1}"; do
  echo "$dir"
  mkdir ${dir}/frames_sampled$1/
  for camera in {A..D}; do
   ffmpeg -i ${dir}/camera_${camera}.h264 -vf "select=not(
      mod(n\,${1})),scale=640:-1" -fps_mode vfr ${dir}/
      frames_sampled$1/camera_${camera}_frame_%03d.jpeg
  done
done
```

Figura 3. Imagens extraídas da base de dados [Thames et al. 2021], licenciada sob CC BY 4.0.

5.1.1. Filtragem e divisão do conjunto de dados

Para definir quais pratos seriam utilizados na etapa experimental, foram estabelecidos os seguintes critérios de filtragem:

- Seleção dos pratos que possuem dados de profundidade no formato de imagem RGB, bem como os frames dos vídeos curtos capturados lateralmente;

- Inclusão apenas dos pratos que apresentam imagens obtidas de 4 ou 5 ângulos diferentes.

A partir dos dados que atenderam aos critérios definidos, realizou-se uma seleção aleatória de 1.000 pratos, representando aproximadamente 20% do conjunto original antes da filtragem. Esses pratos foram organizados em cinco subconjuntos de 200 itens cada, preservando os 4 ou 5 ângulos distintos por prato. Cada subconjunto resultou em cerca de 1.000 imagens, contemplando uma quantidade variável de classes selecionadas dentre as 555 existentes na base, conforme apresentado na Tabela 3. Considerando as limitações computacionais previamente mencionadas, a geração de previsões para cada teste com esse volume de dados demanda, em média, quatro dias de processamento.

Subconjunto	Quantidade de classes
1	160
2	162
3	159
4	162
5	165

Tabela 3. Quantidade de classes de cada subconjunto

5.2. Execução dos testes

O script Python desenvolvido é responsável por selecionar um dos subconjuntos gerados na etapa de filtragem para a realização dos testes. Após essa seleção, todas as imagens presentes no subconjunto escolhido são carregadas em uma lista. Em seguida, o script realiza, individualmente, uma requisição à API para cada imagem, utilizando os parâmetros previamente definidos.

As respostas recebidas são verificadas quanto à validade do formato JSON. Uma vez validadas, essas respostas são armazenadas em um arquivo no formato `.csv`, com o objetivo de viabilizar análises posteriores. Além disso, após o sucesso de cada requisição, a imagem correspondente é movida para um diretório distinto. Essa etapa visa evitar que a mesma imagem seja processada novamente, caso o script seja interrompido durante a execução.

5.3. Pós-processamento e comparação

Ao término da análise de todas as imagens de um subconjunto, o arquivo `.csv` com as previsões geradas é carregado por outro script em Python. Nesse processo, cada resposta em formato JSON é convertida em um dicionário para facilitar sua manipulação. O script também carrega as informações reais de cada prato, permitindo, em seguida, a execução das comparações entre os dados reais e os previstos. Cada prato comparado pode conter ingredientes que foram corretamente previstos, não previstos ou previstos incorretamente (não existentes no prato real), conforme definido na seção de Introdução.

Para cada ingrediente, são acumuladas as quantidades de Verdadeiros Positivos, Falsos Positivos e Falsos Negativos observadas nas comparações, além de uma lista com todos os pesos previstos. No caso de FP, assume-se peso real igual a zero; para FN, o peso previsto é considerado como zero. Ao final, todas essas informações são salvas em um

novo arquivo `.csv`, no formato descrito no Quadro 3, que será utilizado para o cálculo das métricas.

Quadro 3: Formato do `.csv` de pós-processamento

$INGR, TP, FP, FN, LIST\_WEIGHT$
"$nome\_ingrediente$", x, y, z, [["$Categoria$", P_r, P_p], ["$Categoria$", P_r, P_p]...]

Em que P_r representa o peso real e P_p o peso previsto pelo modelo.

5.3.1. Métricas de Avaliação

Na etapa final do experimento, foi realizada a avaliação de desempenho do modelo. Para isso, calcularam-se as métricas clássicas de classificação, Precisão, *Recall* e F1-Score, conforme definidas no referencial teórico, agregando os resultados de todos os ingredientes para obter uma medida global de desempenho.

Para a avaliação da estimativa dos pesos, utilizaram-se as métricas tradicionais de regressão: MAE, RMSE e R^2, também previamente definidas. Diferentemente da avaliação de classificação, as métricas de regressão foram calculadas em dois momentos: primeiramente, considerando todas as estimativas de peso, e em seguida, apenas aquelas associadas a classificações corretas.

Além das métricas quantitativas, foi conduzida uma análise qualitativa para melhor interpretação dos resultados. Consideraram-se apenas ingredientes com ao menos 10 ocorrências (soma de verdadeiros positivos, falsos positivos e falsos negativos) por estarem presentes em mais de dois pratos distintos. Foram listados os 10 ingredientes mais e menos detectados, ordenando-os pelo *recall* (arredondado para duas casas decimais) e, em caso de empate, pelo total de ocorrências. Para os 10 mais detectados, foi gerado um gráfico com os valores de RMSE, permitindo visualizar a média dos erros de cada ingrediente.

Por fim, foi produzido um gráfico de dispersão com os resultados obtidos, consolidando a análise. Todas as etapas foram executadas por meio de códigos desenvolvidos em Python, utilizando os dados disponíveis em arquivos `.csv` com os valores das métricas.

6. Resultados

Em contexto geral, os resultados obtidos evidenciam que o modelo apresenta um baixo desempenho na resolução do objetivo definido para este projeto. Com uma média aproximada entre todos os 5 testes realizados de **51,80%** de precisão, **22,90%** de *recall* e **31,74%** de F-score, o sistema não consegue identificar muitos dos ingredientes presentes no prato e o que ele prevê são poucos confiáveis. A Tabela 4 exibe as métricas de classificação de cada teste realizado.

Ao analisar os 10 ingredientes menos detectados em cada teste, observou-se que o modelo apresenta grande dificuldade em classificar corretamente temperos ou líquidos como azeite de oliva, sal, alho, vinagre, pimenta, limão, mostarda, chalotas e tomilho. Por estarem frequentemente misturados ou incorporados a outros alimentos, a detecção desses itens é desafiadora mesmo para humanos, o que em parte justifica os baixos valores de

Tabela 4. Métricas de classificação

Número do Teste	Total de TP	Total de FP	Total de FN	Precisão	*Recall*	F-score
1	1716	1625	5644	51,36%	23,32%	32,07%
2	1752	1621	6188	51,94%	22,07%	30,97%
3	1591	1653	5922	49,04%	21,18%	29,58%
4	1678	1476	5078	53,20%	24,84%	33,86%
5	1763	1536	5878	53,44%	23,07%	32,23%

recall. A Tabela 5 ilustra, como exemplo, os dez alimentos menos detectados no primeiro teste (todos com recall aproximado igual a 0%), ordenados de acordo com a quantidade decrescente de ocorrências.

Tabela 5. Top 10 ingredientes menos detectados do teste 1

Ingrediente	Total de TP	Total de FN	Total de FP
azeite de oliva	1	472	1
sal	1	460	0
alho	0	298	2
pimenta	1	260	1
vinagre	0	242	0
suco de limão	0	178	0
chalota	0	100	0
acelga	0	84	5
tomilho	0	80	0
lima	0	68	1

Em contraste, ingredientes maiores e menos propensos à mistura, como morango, bacon, ovos mexidos, couve-de-bruxelas, couve-flor e melancia, apresentaram altas taxas de detecção (*recall* acima de 90%) e, na maioria dos casos, precisão superior a 80%. A Tabela 6 apresenta, como exemplo, as métricas obtidas para os dez alimentos mais detectados no primeiro teste, ordenados pelo *recall* e pelo número total de ocorrências.

Além disso, observou-se que, em determinados casos, o modelo realizou uma decomposição dos ingredientes compostos, tentando prever individualmente os componentes de misturas ao invés de classificá-las como um único item agregado. Por exemplo, no caso de um "mix de frutas vermelhas", o modelo previu separadamente ingredientes como morango, mirtilo, amora e framboesa. Esse comportamento gerou divergências em relação aos rótulos de referência, impactando negativamente as métricas de precisão.

Já ao que se refere à estimativa de pesos, o Gemini obteve uma média aproximada entre todos os 5 testes de MAE de **32,50** gramas, erro de **62,53** gramas para o RMSE e um R^2 de **-1,0397**, quando consideramos todas as previsões. Já ao considerar apenas o que foi classificado corretamente (verdadeiros positivos, TP) obtemos uma média entre todos os testes de MAE de **44,80** gramas, RMSE de **63,78** gramas e um R^2 de **-0,6116**. A Tabela 7 mostra as métricas obtidas na estimativa de peso de cada teste.

A performance do modelo na estimativa de peso dos ingredientes mostrou-se con-

Tabela 6. Top 10 ingredientes mais detectados do teste 1

Ingrediente	Total de TP	Total de FN	Total de FP	Recall	Precisão
Ovos mexidos	75	0	14	100%	84%
Uvas	42	0	8	100%	84%
Bacon	66	1	13	99%	84%
Couve de Bruxelas	78	2	10	98%	89%
Melancia	24	1	5	96%	83%
Morango	14	1	48	93%	23%
Couve-flor	31	3	12	91%	72%
Salada Caesar	32	3	2	91%	94%
Bife	18	2	43	90%	30%
Batata frita	22	3	2	88%	92%

Tabela 7. Métricas da estimativa de peso

Número do Teste	MAE Geral	RMSE Geral	R^2 Geral	MAE TP	RMSE TP	R^2 TP
1	31,71g	60,70g	-0,9077	47,65g	67.79g	-0,6362
2	31,73g	62,49g	-0,9059	41,52g	59.25g	-0,4607
3	32,13g	61,94g	-1,0541	45,39g	65.86g	-0,7706
4	34,01g	64,73g	-1,0564	43,02g	61.31g	-0,4189
5	32,94g	62,80g	-1,2744	46,40g	64,70g	-0,7718

sistentemente baixa. Em todos os testes, o modelo apresentou uma alta média de erro, com valores estimados que frequentemente excediam o dobro do peso real. A baixa capacidade preditiva do modelo é validada pelo R^2, que se mostrou negativo em todas as avaliações, indicando que sua performance foi inferior à de uma simples predição baseada na média dos pesos reais. Adicionalmente, a análise dos gráficos de dispersão revela uma tendência sistemática do modelo em superestimar os pesos, conforme ilustrado na Figura 4 para os testes 2 e 4.

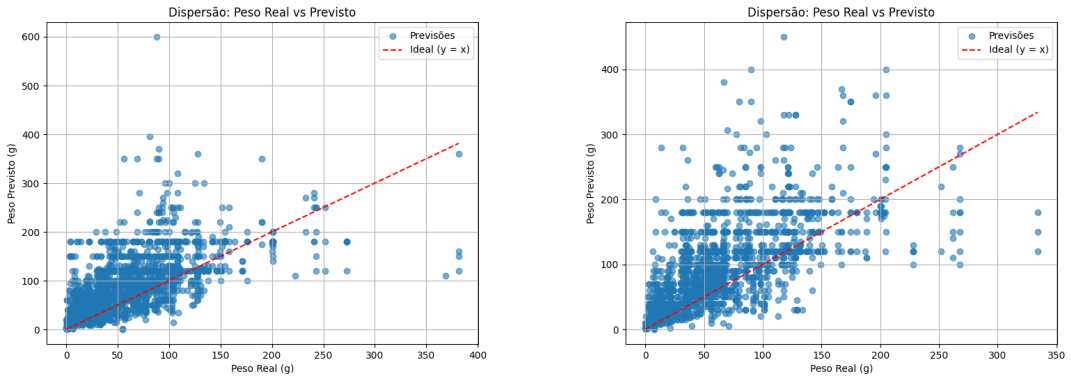

Figura 4. Gráfico de dispersão do teste 2 e 4

A análise gráfica do RMSE, focada nas previsões corretas dos 10 alimentos mais reconhecidos pelo modelo, revela uma variabilidade considerável na média de erro de

cada ingrediente, sendo menor para alimentos como bacon e morango, conforme exibido na figura 5.

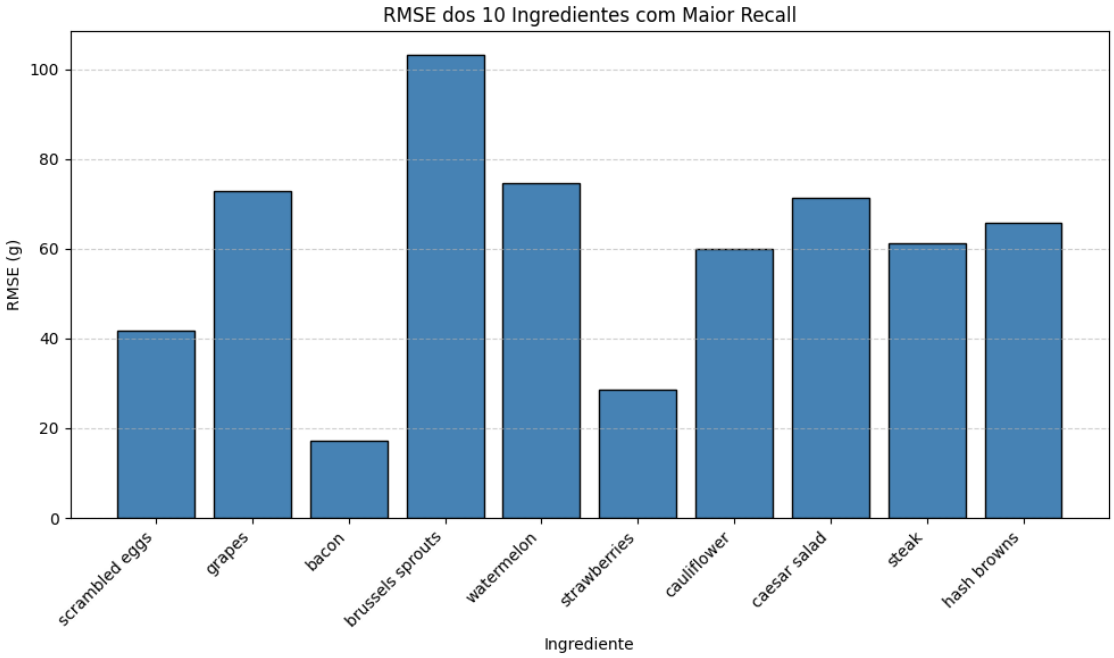

Figura 5. Gráfico de RMSE do teste 1

7. Conclusões

Com o objetivo central de realizar uma avaliação quantitativa e qualitativa da capacidade do modelo generativo Gemini, especificamente a versão 2.5 Flash, em identificar ingredientes e estimar seus respectivos pesos em gramas a partir de imagens de refeições. Foi desenvolvida uma abordagem automatizada que enviou imagens do conjunto de dados Nutrition5k, juntamente com um prompt estruturado, para a API do modelo, analisando estatisticamente as respostas em formato JSON.

Os resultados alcançados indicam um baixo desempenho do modelo Gemini para a tarefa proposta. Na classificação de ingredientes, o sistema demonstrou ser pouco confiável, com métricas de precisão e *recall* consideravelmente baixas em todos os cinco testes realizados, indicando que muitos ingredientes presentes nos pratos não foram identificados (alto número de falsos negativos). Qualitativamente, observou-se que o modelo falha em detectar itens que estão misturados ou servem como tempero, como azeite, sal, alho e pimenta , mas possui uma alta taxa de acerto para ingredientes maiores e visualmente distintos, como frutas e ovos mexidos.

Uma observação relevante foi a tendência do modelo em decompor ingredientes compostos, como mix de frutas vermelhas, em seus componentes individuais, o que, embora demonstre uma capacidade de análise detalhada, gerou divergências em relação aos dados de referência e impactou negativamente as métricas.

No que tange à estimativa de peso, a performance foi igualmente insatisfatória. O modelo apresentou erros médios elevados, tanto no MAE quanto no RMSE. De forma

mais contundente, o R^2 se mostrou consistentemente negativo em todas as análises, o que significa que o desempenho do modelo foi inferior a uma simples estimativa pela média dos pesos reais, indicando sua baixa capacidade preditiva. Os gráficos de dispersão corroboram essa conclusão, revelando uma tendência sistemática do modelo em superestimar o peso dos alimentos.

Comparando estes achados com trabalhos relacionados, as dificuldades encontradas são consistentes com as de outras pesquisas. Por exemplo, o estudo de [O'Hara et al. 2025] sobre o ChatGPT-4 também apontou que, embora a identificação de alimentos seja razoável, a estimativa de porções é imprecisa, especialmente para porções maiores. Da mesma forma, os desafios na segmentação de ingredientes misturados e na estimativa de volume a partir de imagens 2D, como destacados na revisão de [Sultana et al. 2023], refletem-se diretamente nos baixos resultados deste trabalho.

Para aprimorar os resultados obtidos, propõe-se, em trabalhos futuros, restringir a análise a refeições cujos ingredientes não estejam excessivamente misturados, bem como realizar a categorização prévia dos alimentos e excluir os itens líquidos ou considerados temperos, como sal e pimenta, que podem introduzir ruído na análise. Além disso, recomenda-se substituir a utilização de apenas uma imagem por requisição pela captura de múltiplas imagens, obtidas de diferentes ângulos da mesma refeição, ou até mesmo pela gravação de um vídeo curto que apresente todo o prato. Essa abordagem pode facilitar a reconstrução tridimensional dos ingredientes pelo modelo, aumentando a precisão da identificação.

Outra possibilidade é a experimentação com diferentes modelos de IA generativa, incluindo versões pagas do GPT (OpenAI) e do Grok (xAI), bem como outras variantes do Gemini, como a versão 2.5 Pro, a fim de realizar comparações de desempenho. Também se sugere investigar o impacto de diferentes técnicas de engenharia de prompt, avaliando como variações no enunciado influenciam a qualidade e a consistência dos resultados obtidos.

Referências

Barten, A. P. (1987). *The coefficient of determination for regression without a constant term*, pages 181–189. Springer Netherlands, Dordrecht. https://doi.org/10.1007/978-94-009-3591-4_12.

de Milano, D. and Honorato, L. B. (2014). Visão computacional. *UNICAMP Universidade Estadual de Campinas FT Faculdade de Tecnologia*. https://www.academia.edu/9621896/VISO_COMPUTACIONAL_Palavras_Chaves.

Ipsos (2024). *THE IPSOS AI MONITOR 2024*. Ipsos Global Advisor. Pesquisa em 32 países sobre atitudes globais em relação à inteligência artificial. https://www.ipsos.com/pt-br/ipsos-ai-monitor-2024.

Mariano, D. (2021). Métricas de avaliação em machine learning: acurácia, sensibilidade, precisão, especificidade e f-score. In *BIOINFO – Revista Brasileira de Bioinformática*, chapter 15. Alfahelix. doi:10.51780/978-6-599-275326-15.

Marvin, G., Hellen, N., Jjingo, D., and Nakatumba-Nabende, J. (2024). Prompt engineering in large language models. In Jacob, I. J., Piramuthu, S., and Falkowski-Gilski,

P., editors, *Data Intelligence and Cognitive Informatics*, pages 387–402, Singapore. Springer Nature Singapore.

Molnar, C. (2025). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Christoph Molnar, 3 edition. https://christophm.github.io/interpretable-ml-book.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., and Mian, A. (2025). *A Comprehensive Overview of Large Language Models*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3744746.

Norvig, P. and Russell, S. (2014). *Inteligência artificial: Tradução da 3a Edição*. Elsevier Brasil. isbn: 9788535251418. https://books.google.com.br/books?id=BsNeAwAAQBAJ.

O'Hara, C., Kent, G., Flynn, A. C., Gibney, E. R., and Timon, C. M. (2025). An evaluation of chatgpt for nutrient content estimation from meal photographs. *Nutrients*, 17(4):607. https://www.mdpi.com/2072-6643/17/4/607.

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., and Chadha, A. (2025). A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*. https://arxiv.org/abs/2402.07927.

Sultana, J., Ahmed, B. M., Masud, M. M., Huq, A. K. O., Ali, M. E., and Naznin, M. (2023). A study on food value estimation from images: Taxonomies, datasets, and techniques. *IEEE Access*, 11:45910–45935. doi: 10.1109/ACCESS.2023.3274475.

Thames, Q., Karpur, A., Norris, W., Xia, F., Panait, L., Weyand, T., and Sim, J. (2021). Nutrition5k: Towards automatic nutritional understanding of generic food. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8899–8907.

Wang, J., Jiang, H., Liu, Y., Ma, C., Zhang, X., Pan, Y., Liu, M., Gu, P., Xia, S., Li, W., et al. (2024). A comprehensive review of multimodal large language models: Performance and challenges across different tasks. *arXiv preprint arXiv:2408.01319*. https://arxiv.org/abs/2408.01319.

World Obesity Federation (2025). *World Obesity Atlas 2025*. World Obesity Federation, Londres. https://www.worldobesity.org/resources/resource-library/world-obesity-atlas-2025.