UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

Visão Computacional versus Vision-Language Models em Edge AI

Fine-Tuning, Quantização e Trade-offs de Desempenho

Hugo Rodrigues Pessoni

UFG
UNIVERSIDADE FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

HUGO RODRIGUES PESSONI

Visão Computacional versus Vision-Language Models em Edge AI

Fine-Tuning, Quantização e Trade-offs de Desempenho

Goiânia
2025

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): HUGO RODRIGUES PESSONI

Título do trabalho: Visão Computacional versus Vision-Language Models em Edge AI

Fine-Tuning, Quantização e Trade-offs de Desempenho

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Hugo Rodrigues Pessoni**, **Discente**, em 05/02/2026, às 09:07, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Fernando Marques Federson**, **Professor do Magistério Superior**, em 13/03/2026, às 11:34, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

HUGO RODRIGUES PESSONI

Visão Computacional versus Vision-Language Models em Edge AI

Fine-Tuning, Quantização e Trade-offs de Desempenho

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Orientador: Prof. Dr. Fernando Marques Federson

Goiânia

2025

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

PESSONI, HUGO RODRIGUES
Visão Computacional versus Vision-Language Models em Edge AI [manuscrito]: Fine-Tuning, Quantização e Trade-offs de Desempenho / HUGO RODRIGUES PESSONI. - 2025.
56 f.: 2025

Orientador: Prof. Dr. Fernando Marques Federson
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2025.

1. Inteligência Artificial. 2. Modelos de Visão-linguagem. 3. Dispositivos de Borda.

I. Federson, Fernando Marques , orient. II. Título.

CDU 004

HUGO RODRIGUES PESSONI

Visão Computacional versus Vision-Language Models em Edge AI

Fine-Tuning, Quantização e Trade-offs de Desempenho

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Data da Aprovação: 09 de dezembro de 2025.

Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)

Prof. Dr. Aldo André Díaz Salazar
Coordenador de TCC do BIA (INF-UFG)

Prof. Dr. Anderson da Silva Soares
Coordenador do BIA (INF-UFG)

Prof. Dr. Sávio Salvarino Teles de Oliveira
(INF-UFG)

HUGO RODRIGUES PESSONI

Visão Computacional versus Vision-Language Models em Edge AI

Fine-Tuning, Quantização e Trade-offs de Desempenho

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Otimização de Modelos VLM**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: Inteligência artificial; Modelos de visão-linguagem; Dispositivos de borda.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **VLM Model Optimization**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

Keywords: Artificial intelligence; Vision-language models; Edge devices.

Goiânia

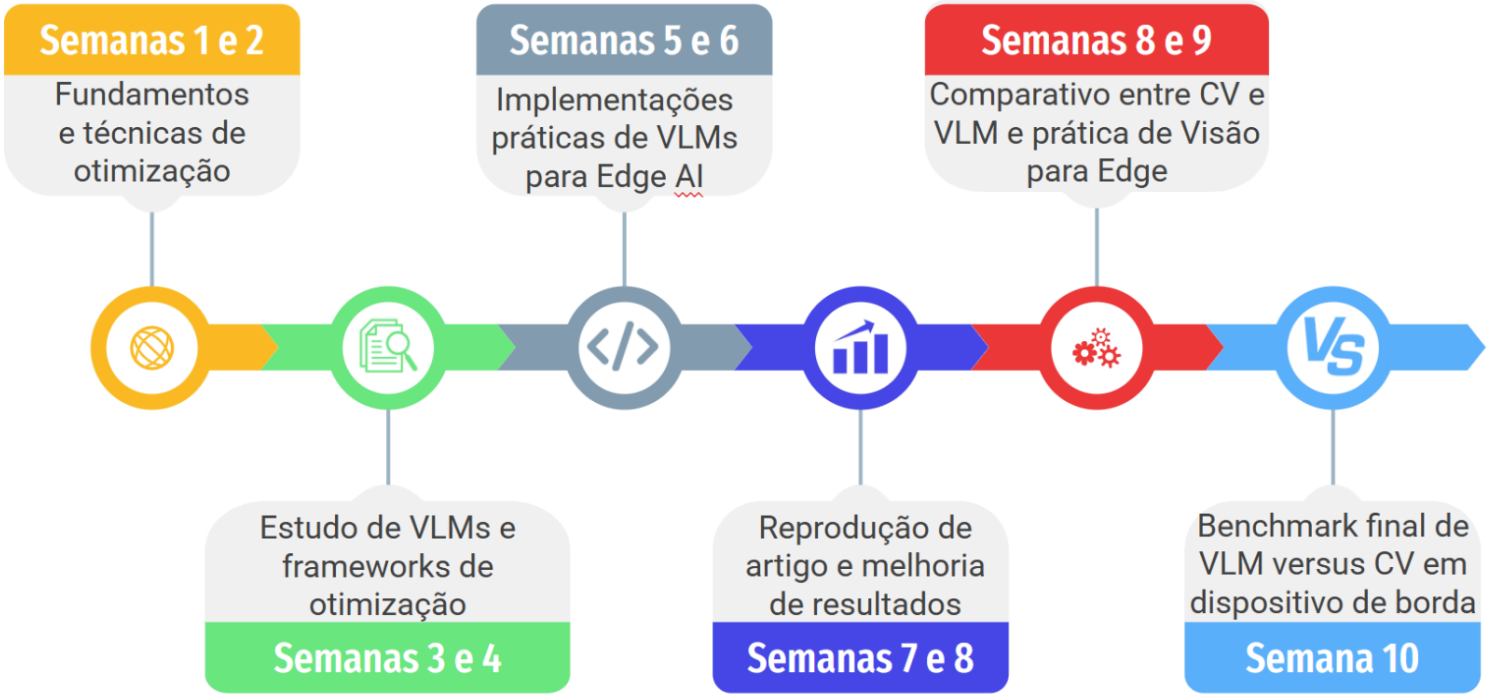
2025

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Minha Jornada

Hugo Rodrigues Pessoni

Especialista em: Otimização de Modelos VLM



Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

MINHA JORNADA

Nome: **Hugo Rodrigues Pessoni**

Especialidade: **Otimização de Modelos VLM**

Objetivo deste documento

Durante o processo da disciplina Residência em IA[1], foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

Minha jornada começou na **Semana 1** com uma dúvida: “Sobre o que eu realmente gosto e gostaria de me aprofundar?”. Dentre dezenas de ideias, algumas desafiadoras e outras nem tanto, realizei uma busca exploratória para definir o tema da minha especialização. Acabei percebendo a paixão que tenho em hardwares e dispositivos eletrônicos desde muito jovem. Mas como unir essa paixão com o bacharelado em IA? A resposta era óbvia, mas não tão clara no começo: IoT e Edge Computing. Inicialmente, dediquei-me a entender o panorama da Otimização de Modelos para Sistemas Embarcados, consumindo desde vídeos introdutórios sobre o tema, blogs, repositórios e surveys, até documentações técnicas de grandes players como OpenAI, Google e Tensorflow. Foi um período de descoberta para mim, em que notei que é uma área que já existia a muito tempo e que as pessoas inovam a cada dia para todo tipo de modelo, seja LLM, seja Visão Computacional ou até mesmo algoritmos clássicos, tornando mais eficientes em relação a: tempo de inferência, consumo energético e/ou uso de recurso computacional. Essa percepção amadureceu na **Semana 2**, onde percebi que o termo genérico de "otimização"

[1] Dez Semanas, entre setembro de 2025 e dezembro de 2025.

era um “guarda-chuva” vasto e que cobre desde compressão de modelos, arquiteturas compactas, otimização de software, otimização de hardware até fine-tuning. Após a leitura crítica de surveys como "*Optimizing Edge AI" e “Vision-Language Models for Edge Networks*”, compreendi a necessidade de refinar o escopo. Ao invés de me perder na generalidade ou focar apenas em LLMs ou Visão Computacional isoladamente, como eu pensava anteriormente, decidi direcionar meus esforços para a interseção dessas duas áreas: os *Vision-Language Models* (VLMs). Essa decisão me direcionou a mudar a minha metodologia de busca de referências para artigos principalmente. Todas as anotações sobre os artigos aqui relatados, referências, vídeos e resumos estão diretamente no **Apêndice 1.**

Com o tema ajustado para **“**Otimização por compactação de modelos VLM voltado para Computação de Borda**”**, as **Semanas** seguintes foram dedicadas a “desmistificar” e entender o funcionamento dessas arquiteturas complexas, uma vez que, eu nunca havia trabalhado com elas antes. Na **Semana 3**, me aprofundei na matemática e nos mecanismos por trás dos VLMs, estudando conceitos fundamentais como CLIP, ALIGN e Vision Transformers, além de pré-treino desses modelos, benchmarks atuais e as tasks para VLMs. Esses estudos e testes práticos realizados, em especial, com infográficos do funcionamento dos VLMs estão presentes no **Apêndice 2**. O objetivo não era apenas usar ou ver as ferramentas em ação, mas entender como esse alinhamento entre texto e imagem ocorre de verdade no espaço latente. Esse embasamento teórico foi crucial para a **Semana 4**, onde iniciei uma busca por trabalhos que falassem sobre VLM para dispositivos de borda, além de trabalhos que comparassem diretamente VLMs com modelos tradicionais de Visão Computacional. Os resultados desta busca também estão reunidos no **Apêndice 2**. Foi nesse momento que percebi uma lacuna entre a literatura e a necessidade prática: eu precisaria estruturar o conhecimento sobre as frameworks disponíveis para realizar a otimização de modelos em geral. Como resposta, iniciei uma busca para criar uma tabela comparativa, dos principais frameworks de otimização e quantização que encontrei no GitHub, facilitando as decisões técnicas nas etapas práticas que com certeza estavam por vir. Nesse momento também fiz um refinamento do tema da Residência para “Otimização de Modelos VLM”.

A transição da teoria para a implementação prática marcou a fase intermediária da minha jornada na Residência. Na **Semana 5**, iniciei primeiro adquirindo conhecimento através de vídeos explicativos sobre os processos internos de modelos VLM, além da criação de um diagrama (presente no **Apêndice 3**) sobre os frameworks obtidos na **Semana 4** com o intuito de criar uma visualização dessas ferramentas. Em seguida acompanhei tutoriais avançados, linha a linha, para a construção de um VLM do zero, mostrando como modelos multimodais combinam a Visão Computacional e o Processamento de Linguagem Natural (NLP) para a task de resposta visual a pergunta (Vision QA), saindo assim da abstração dos artigos para a realidade do código. A jornada experimental se intensificou na **Semana 6**, onde implementei um *Vision Transformer* (ViT) do zero, além de seu treinamento para aplicação de classificação de imagens no dataset MNIST. Ao invés de apenas importar bibliotecas prontas para esse processo, reconstruí a arquitetura linha a linha, compreendendo o funcionamento dos *patches*, *embeddings* e mecanismos de atenção. Essa etapa foi fundamental para “desmistificar” os termos e me deu a confiança necessária para dar continuidade às fases seguintes. Ao longo das **Semanas,** continuei a leitura de artigo, além da busca e preenchimento da tabela e do diagrama de frameworks. Todas as etapas aqui realizadas estão presentes no **Apêndice 3**.

Dando continuidade na jornada experimental, na **Semana 7**, realizei o Fine-tuning e a otimização de um modelo VLM (Qwen2-VL-7B) voltado para interpretação de gráficos. O objetivo é que o modelo pudesse responder perguntas quantitativas e qualitativas a partir de imagens de gráficos. Conforme apresentado no **Apêndice 4,** os resultados não foram bons, mas ficou claro que isso aconteceu por causa do tempo de treinamento adotado para o teste. Durante o processo de treinamento, lembrei do artigo lido anteriormente "*A Comparative Study of CNNs and Vision-Language Models for Chart Image Classification*" e identifiquei uma oportunidade de intervenção: os autores compararam CNNs treinadas (com Fine-tuning) contra VLMs em modo zero-shot em um hardware de alta capacidade. Esta comparação me pareceu uma “injusta”. Com isso, surgiu uma pergunta: “seria possível melhorar o resultado do modelo VLM com Fine-tuning ao ponto de ser melhor que um modelo de Visão Computacional também com um Fine-tuning aplicado?”. Motivado por essa hipótese, na **Semana 8**, comecei a criação/adaptação do *dataset* original do artigo

disponibilizado no repositório do GitHub para reproduzir o experimento com o modelo com os piores resultados do artigo, o modelo PaliGemma (Gemma 2B como decoder + SigLIP-So400m). Além disso, busquei entender um pouco mais esse modelo através do artigo que o originou: "*PaliGemma: A versatile 3B VLM for transfer*". Todas as anotações sobre esta etapa estão reunidas no **Apêndice 4**. Não me limitei apenas a rodar o código original do artigo, construí um *dataset* adaptado e executei um processo de Fine-tuning utilizando técnicas de LoRA *(Low-Rank Adaptation*) em camadas específicas da arquitetura do modelo. Os resultados, presentes no **Apêndice 4**, validaram minha tese: o modelo VLM ajustado superou significativamente o desempenho zero-shot de todos os outros modelos relatados no artigo original, provando o potencial dessas arquiteturas VLM quando devidamente especializadas para funções específicas.

Na **Semana 9,** decidi introduzir restrições reais de hardware configurando um ambiente de *Edge Computing* utilizando uma *Single Board Computer* (SBC) Orange Pi 3B. Dest forma, realizei o mesmo teste de comparação entre VLM e Visão Computacional, porém em um hardware bem menor ao utilizado no artigo. O desafio consistia em estabelecer uma linha de base sólida para a tarefa de contagem de objetos. Para isso, implementei e testei modelos de Visão Computacional tradicionais, especificamente o YOLOv8 (versão nano) e o MobileNetV3, aplicando técnicas de quantização para FP16 e INT8 via *ONNX Runtime*. Ao todo, foram testados 6 modelos sendo: 2 de base FP32, 2 em FP16, e 2 em INT8, em 3 diferentes thresholds (0.3, 0.5 e 0.7), totalizando 18 modelos (testes). Todos os testes foram realizados no mesmo hardware e com o mesmo *dataset*. Todos os resultados estão reunidos em uma planilha no **Apêndice 5**. Além disso, foi levantado uma lista de possíveis VLMs que poderiam rodar na SBC para fim de comparação com os modelos de visão tradicionais.

Na **Semana 10**, realizei o *benchmark* final entre a Visão Computacional e a nova geração de modelos multimodais no dispositivo de borda. Consegui executar modelos VLMs de diferentes escalas na SBC Orange Pi: Qwen2-VL-2B, MobileVLM-v2, SmolVLM-500M e SmolVLM-256M. Um dos desafios era conseguir alocar modelos de bilhões de parâmetros em um hardware modesto. Esses modelos foram então quantizados em FP16 e INT8

utilizando o framework *Bitsandbytes* com objetivo que a comparação fosse equilibrada e "justa". A análise cruzada dos dados entre os resultados de um total de 12 modelos VLM revelou um trade-off claro e valioso quando comparado aos 18 modelos de Visão Computacional. Enquanto os modelos de detecção tradicionais (YOLO) ofereceram velocidade superior para aplicações em tempo real, os VLMs demonstraram uma capacidade de compreensão e precisão de contagem "drasticamente" superior, ainda que com maior latência. Essa conclusão sintetiza um dos meus aprendizados da Residência, demonstrando que a escolha entre arquiteturas depende intrinsecamente dos requisitos exigidos como velocidade de inferência, profundidade semântica e hardware disponível. Todos os testes e resultados estão disponíveis no **Apêndice 6.**

A conclusão do processo da Residência em IA não representa apenas o fim de uma série de experimentos acadêmicos e estudos aprofundados, mas a consolidação de uma fase importantíssima na minha vida. Ao navegar desde a escolha inicial do tema, a fundamentação teórica até a execução de *benchmarks* em hardwares de borda, desenvolvi uma visão holística sobre os desafios reais da implementação de Inteligência Artificial. A capacidade de transitar entre a complexidade dos *Vision Language Models* e as restrições físicas do *Edge Computing* me permitiu compreender que a verdadeira inovação não está apenas no tamanho do modelo, mas na sua aplicabilidade principalmente. Sinto-me, portanto, preparado para atuar como um Especialista em Otimização de VLMs para Edge Computing, apto a transformar o potencial de grandes modelos em soluções tangíveis e limitadas, conectando a inovação da pesquisa acadêmica à viabilidade técnica exigida pelo mundo real.

Por fim, gostaria de expressar minha gratidão primeiramente à minha família, cujo apoio incondicional foi o que me sustentou em um dos momentos mais difíceis da minha vida. Vocês me deram todo o suporte e a coragem necessários para iniciar esta nova carreira a qual eu amo. Aos meus amigos, tanto os de longa data quanto os novos que esta jornada me trouxe, obrigado por não me deixarem desanimar e por me mostrarem, diariamente, que estar cercado de pessoas boas apenas nos engrandece e nos impulsiona. Por fim, e não menos importante, meu agradecimento especial aos docentes do Bacharelado em IA. Obrigado por se dedicarem à transformar o futuro do país, mudando o

modelo tradicional de ensino e construindo um ambiente acadêmico como ele realmente deveria ser: uma preparação humanizada, prática e visionária para o futuro dos jovens.

APÊNDICE 1

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 4 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

HUGO RODRIGUES PESSONI

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Para a primeira Semana do processo da Residência em IA foi feito:

- Definição do tema oficial.
- Entender o tema e isso inclui história, conceitos, técnicas e problemas. Isso foi feito:
  a. Primeiro entendendo onde estou me envolvendo, fui ver uma "netflix":
     - https://www.youtube.com/watch?v=QfFRNF5AhME
     - https://www.youtube.com/watch?v=m2LokuUdeVg
     - https://www.youtube.com/watch?v=K75j8MkwgJ0&t=64s
     - https://www.youtube.com/watch?v=AIGOSz2tFP8
  b. Depois, realizei uma busca (página 15 do Google Search) por referências de onde vou me basear para a escrita de um artigo, nisso foram levantadas:
     - 3 artigos
     - 15 blogs (medium, ultralytics, ibm, hugging face, etc)
     - 10 Documentações Técnicas (OpenAI, Google AI Edge, TensorFlow)
     - 1 repositório
     - 2 survey
  c. Buscou-se então separar o que era útil ou não, pegando as técnicas e as conclusões aplicadas.
  d. Nisso, foi escrito um resumo, englobando praticamente todas as possíveis técnicas de serem exploradas - https://docs.google.com/document/d/1FuNBIFRXAHSBIBAZmakwDwavZsLVngVCnjwq-Ts8aeY/edit?usp=sharing
  e. Percebi que o buraco é bem mais embaixo.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Leitura (dinâmica) dos 2 surveys para verificar se não faltou nenhuma técnica ou conceito:

- Optimizing Edge AI: A Comprehensive Survey on Data, Model, and System Strategies (**2025**) **- 24**

páginas, porém apenas 15 são em relação a IA.

- Optimizing Deep Learning Models for Edge Computing: Techniques for Efficient Inference, Model Compression, and Real-Time Processing (Australian Journal of Machine Learning Research & Applications - **2024) - 39 páginas**

Decidir como atacar esse assunto. Em relação a se será comparar modelos de Visão ou LLM e quais técnicas dar foco.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Resumo do documento citado no Termo de Aceite de Entrega do dia 4 de setembro de 2025:

Link:
https://docs.google.com/document/d/1FuNBIFRXAHSBIBAZmakwDwavZsLVngVCnjwq-Ts8aeY/edit?tab=t.1x3dsa2z2g1

O documento apresenta um estudo aprofundado sobre a viabilidade de executar modelos complexos de Inteligência Artificial e *Machine Learning* em ambientes com recursos restritos, como dispositivos IoT, dispositivos de borda, *wearables* e sistemas automotivos.

O texto aborda desde a fundamentação teórica do assunto até as técnicas práticas mais modernas para equilibrar desempenho (acurácia) e eficiência (consumo de energia e memória).

Principais tópicos abordados durante o texto:

- Contexto e História: Uma análise da evolução da otimização, desde suas raízes na matemática aplicada e engenharia até se tornar um requisito estratégico para a expansão da *Edge AI.*
- Conceitos Fundamentais: Definição clara do que é otimização (performance vs. eficiência vs. generalização) e as motivações críticas para a aplicação, como latência em tempo real, limitações de bateria e redução de custos de infraestrutura.
- A Tríade de Trade-offs: Discussão sobre o equilíbrio delicado entre as três métricas principais: Acurácia, Velocidade (Latência) e Uso de Recursos.
- Guia de Técnicas de Otimização: detalha as principais metodologias utilizadas na indústria, incluindo:
    - Quantização: Redução da precisão numérica dos pesos dos modelos (ex: FP32 para INT8) para economia de memória, cobrindo técnicas como PTQ (*Post-Training*) e QAT (*Quantization-Aware Training*).
    - Poda (*Pruning*): Remoção estratégica de neurônios ou conexões redundantes (poda por magnitude e estruturada).
    - Destilação de Conhecimento: Transferência de aprendizado de um modelo "Professor" complexo para um modelo "Aluno" mais leve.
    - Outras Estratégias: Otimização de hiperparâmetros, *clustering* de pesos, *sparsity* e otimizações específicas para *hardware* (GPUs, TPUs).

O texto completo serve como um guia para entender como tornar a Inteligência Artificial acessível, escalável e eficiente fora dos grandes servidores, detalhando as ferramentas necessárias para superar as limitações físicas de hardware.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 11 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

HUGO RODRIGUES PESSONI

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Para a segunda Semana do processo da Residência em IA foi feito:

1. Leitura dos survey que foram citados no Gate 1 e conferência das técnicas. Foi quando entendi que otimização é um nome generalizado que pode conter:
    a. Model Compression
    b. Lightweight (Compact) Architectures
    c. Software Optimization (frameworks voltados pra isso)
    d. Hardware Optimization
    e. Fine-Tuning
2. Foi incrementado no documento de resumo do Gate 1, o que foi aprendido com a leitura dos dois survey acima. https://docs.google.com/document/d/1FuNBIFRXAHSBIBAZmakwDwavZsLVngVCnjwq-Ts8aeY/edit?usp=sharing
3. Foi encontrado um terceiro survey intitulado: Vision-Language Models for Edge Networks: A Comprehensive Survey - esse de 2 meses atrás.
4. Antes dúvida estava entre comparar modelos de Visão ou LMMs, resolvi juntar e voltar os estudos aos VLMs.
5. Por conta da escrita, percebi a necessidade de elaborar uma boa metodologia de busca.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

1. Entender a matemática por trás dos VLMs, como funcionam e quais técnicas há hoje em dia.
2. Entender como a compressão impacta esses modelos.
3. Buscar artigos relacionados de acordo com a metodologia criada.
4. Elaborar estratégias para a Residência, ou seja, será uma comparação, uma revisão, uma aplicação em específico.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

leitura do survey:
https://drive.google.com/drive/folders/1m5P7YlMve-X4pFvG0RIKFo98f-MEdoUA?usp=sharing

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Resumo do documento citado no Termo de Aceite de Entrega do dia 11 de setembro de 2025:

Link: https://docs.google.com/document/d/1FuNBlFRXAHSBIBAZmakwDwavZsLVngVCnjwq-Ts8aeY/edit?tab=t.umppnk4mp7dq

O documento oferece uma consolidação técnica sobre como viabilizar Inteligência Artificial em ambientes descentralizados, dando continuidade ao Gate 1. Ele conecta as técnicas de compressão de modelos com a arquitetura de *Edge Computing* e é apresentado um protocolo metodológico definido para a seleção de artigos científicos.

Principais tópicos abordados durante o texto:

- Otimização de Modelos de Deep Learning Uma análise detalhada das técnicas para tornar redes neurais "mais rápidas, menores e precisas", essenciais para hardware limitado. O texto explora:
    - Técnicas de Compressão: Explicação aprofundada de *Pruning* (Poda estruturada e não estruturada), Quantização (PTQ, QAT, 16x8) e Destilação de Conhecimento.
    - Ajuste Fino e Hiperparâmetros: Estratégias como *Fine-Tuning* supervisionado e Otimização Bayesiana.
    - Frameworks de Mercado: Comparativo entre ferramentas como NVIDIA TensorRT, OpenVINO, TensorFlow Lite e ONNX Runtime.
- Arquitetura de Edge Computing, um paradigma da computação de borda, detalhando a colaboração entre as três camadas principais: o "IoT" (dispositivo final/inicial), a Borda (processamento local) e a Nuvem.
- Além da arquitetura, são discutidos os desafios críticos como latência, segurança de dados (privacidade), e a otimização de custos energéticos e de comunicação.
- Protocolo de Pesquisa: apresenta a metodologia estruturada para a revisão sistemática de literatura focada em VLMs (Vision-Language Models) em sistemas embarcados.

APÊNDICE 2

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 17 de set. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

HUGO RODRIGUES PESSONI

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Para a terceira Semana do processo da Residência em IA foi feito:

- Iniciei os estudos sobre VLMs e como é seu funcionamento, desde o pré-treino até os benchmarks atuais aplicados e as tasks para esse tipo de modelo.
- Para isso tive de explorar um pouco mais sobre seu funcionamento. Mais um vez cai em um cenário imenso, com diversas técnicas possíveis de serem realizadas para um bom funcionamento. **CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021)**
- Foquei em entender a situação da avaliação desses modelos e percebi que é tão generalizada quanto para NLP.
- Estou buscando querer comparar um modelo de visão (Yolo) quantizado e não quantizado e um VLM quantizado e não quantizado para uma mesma task de identificação de objetos.
- Métodos de quantização para VLMs, no que tange à compactação não muda, o que posso promover essa comparação.
- Utilizando a metodologia proposta no Gate 2 para pesquisa de artigos, foram selecionados 10 artigos, em conjunto com os 3 surveys anteriores como base sólida daqui pra frente.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Pretendo já começar com a escrita do documento final, começando ali pela introdução, motivação.
- Pretendo ler 2 artigos por semana desses que levantei, no caso serão 4 semanas no total.
- Pretendo iniciar tutoriais de código em relação ao meu tema, ir pelo básico primeiro.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Tudo que está criado e estudado tenho colocado nessa pasta:
https://drive.google.com/drive/folders/1m5P7YlMve-X4pFvG0RIKFo98f-MEdoUA?usp=sharing

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Resumo do documento citado no Termo de Aceite de Entrega do dia 17 de setembro de 2025:

Link:

Este documento detalha o funcionamento dos VLMs (Vision-Language Models) como sistemas multimodais que unem a visão computacional e o processamento de linguagem natural. Ao contrário dos modelos tradicionais de Visão Computacional, limitados a um número fixo de classes, os VLMs interpretam imagens, gráficos e documentos PDF e conseguem contextualizá-los com a flexibilidade da linguagem humana.

Principais tópicos técnicos abordados nesse documento:

- Aplicações Práticas (*Tasks*): O texto define as capacidades centrais dos VLMs, como VQA (*Visual Question Answering*), *Image Captioning* e *Document Understanding.*
- A Ponte entre Visão e Linguagem:
    - Explicação de como os *pixels* das imagens são traduzidas para algo que um LLM entenda em forma de *tokens*.
    - O papel crucial do *Vision Encoder* para criar vetores de características e do *Projector* para alinhar esses vetores ao espaço de embeddings do modelo de linguagem.
- O Paradigma CLIP (*Contrastive Language-Image Pre-training*):
    - Como imagens e textos são mapeados para o mesmo espaço numérico, ou seja, espaço latente.
    - A vantagem do *Zero-shot*: A capacidade de classificar objetos nunca vistos durante o treinamento, superando a limitação de *labels* fixas da visão computacional clássica.
- Metodologia de Treinamento:
    - Descrição do processo de aprendizado contrastivo utilizando pares de imagem-texto.

- A lógica da "Matriz de Distâncias": Maximizar a similaridade na diagonal (pares corretos) e minimizar fora dela, treinando os encoders para alinharem suas representações.

- Componentes Arquiteturais Persistentes:
  - Uma análise dos blocos de construção que se mantêm relevantes, incluindo *Vision Encoders* (ViT), *Text Decoders* e *Cross-Attention Mechanisms* (que permitem que a visão influencie a geração do texto).

O documento completo é essencial para compreender a base teórica de como as máquinas modernas "enxergam" e "falam" simultaneamente, detalhando a matemática e a arquitetura por trás dessa integração. As estruturas foram compiladas em um mapa mental.

Figura 1: Mapa mental sobre o funcionamento de modelos VLM.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: **25 de set. de 2025**

Participantes da Entrega **[matriculados em Residência em IA]**:

HUGO RODRIGUES PESSONI

Entrega: **[descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]**

Para a quarta Semana do processo da Residência em IA foi feito:

- Iniciei a leitura de artigos mais diretamente voltados para a área de interesse, com destaque para três principais:
  - Ternarization of Vision Language Models for use on edge devices
  - On-Device Language Models: A Comprehensive Review
  - VaVLM: Toward Efficient Edge-Cloud Video Analytics With Vision-Language Models.
- Além disso, comecei a acompanhar uma pessoa chamada Julia Turc, engenheira de software no Google. Apesar de não falar diretamente sobre VLMs, os materiais em NLP têm sido de grande ajuda para entender com mais clareza a matemática por trás dos modelos e ampliar meu entendimento sobre o assunto.
- Com a definição mais clara do tema e de um objetivo como a comparação entre modelos de Visão Tradicionais e VLMs em tasks específicas comecei a procurar quem mais teria feito isso e descobri uma nova área de aplicação e testes, com até mesmo benchmarks sendo criados.
- Para melhor a busca, adaptei a metodologia de busca de artigos já utilizada anteriormente no Gate 2, ajustando alguns tópicos, e utilizei a ferramenta manus.ai como apoio para encontrar as referências de que precisava (foram encontrados 10 artigos no total).
- Mudei de direção no aspecto prático. A ideia era partir para um pouco mais de prática e busquei tutoriais de ferramentas e frameworks que implementam quantização a modelos. Com uma variedade grande existente, percebi a necessidade de estruturar melhor esse conhecimento e aprender mais um pouco. Para isso, montei uma tabela comparativa entre os principais frameworks (descrição, vantagens e desvantagens).
- Por fim, elaborei também um fluxo de informação derivado dessa tabela, buscando facilitar a visualização e a análise comparativa (hardware e técnicas aplicáveis). Esse material deve servir de apoio nas próximas etapas para me ajudar a escolher em relação à prática.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

- Pretendo escolher um desses frameworks para teste.
- Iniciar a parte prática
- Continuar a leitura dos artigos e concluir mais 3 (2 para quantização e 1 de comparação)

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Tudo que está sendo criado e estudado tenho colocado nessa pasta:
https://drive.google.com/drive/folders/1m5P7YlMve-X4pFvG0RIKFo98f-MEdoUA?usp=sharing

Documento relativo ao o Gate 4:
https://docs.google.com/document/d/1FuNBlFRXAHSBIBAZmakwDwavZsLVngVCnjwq-Ts8aeY/edit?usp=sharing

Fluxos/Mapas mentais/Diagramas:
https://www.canva.com/design/DAGzK1-PyvQ/_SlvyNwZJtyut8oD1IITuw/edit?utm_content=DAGzK1-PyvQ&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Resumo do documento citado no Termo de Aceite de Entrega do dia 25 de setembro de 2025:

Este documento compila o estado da arte na execução de *Vision-Language Models* (VLMs) e *Large Language Models* (LLMs) em dispositivos de borda (*Edge Devices*). O conteúdo abrange desde técnicas de compressão matemática até arquiteturas de sistemas distribuídos, e finaliza com um melhoramento do protocolo metodológico apresentado anteriormente para a pesquisa de artigos.

Destaques para os artigos analisados:

1. Ternarização de VLMs (Pesos em -1, 0, +1) Um estudo prático sobre como comprimir o modelo Moondream2 (~1.6B parâmetros) para rodar em celulares.

- A Técnica: Conversão dos pesos para valores ternários usando inicialização K-means e operadores customizados no TensorFlow Lite.
- O Resultado: A variante `q2-matmul` provou ser o melhor *trade-off*, sendo 2x mais rápida e usando 50% menos memória que a quantização INT8 padrão, mantendo uma perplexidade aceitável para uso real.

2. *Review* Abrangente: LLMs "On-Device" Um panorama das estratégias para viabilizar IA generativa fora da nuvem.

- Arquiteturas Eficientes: Discussão sobre modelos "Deep & Thin" e *Mixture of Experts* (MoE).
- Co-design**:** A necessidade de alinhar software (quantização, *pruning*) com aceleradores de hardware (NPUs/DSPs).

3. VaVLM: Análise de Vídeo Híbrida (Edge-Cloud) Apresentação de um sistema onde o VLM atua como um "filtro inteligente".

- Funcionamento: O dispositivo de borda usa um VLM para recortar apenas as Regiões de Interesse (RoI) e as envia para a nuvem.

- Impacto: Redução de 80% na largura de banda e 89% no custo computacional, viabilizando análise de vídeo em tempo real com hardware modesto (ex: Raspberry Pi).

Sobre o protocolo de pesquisa a metodologia é alterada para um estudo comparativo entre VLMs e Modelos de Visão Computacional Tradicionais (CNNs clássicas).

- Objetivo: Mapear em quais cenários (classificação, detecção, etc.) os VLMs superam os modelos tradicionais e onde o custo computacional dos VLMs não se justifica.
- Critérios de Busca: Foco em publicações recentes (2018-2025) que apresentem benchmarks quantitativos de acurácia vs. eficiência energética/latência.

Este material consolida evidências de que a execução de modelos multimodais na borda não é apenas teórica, mas prática, desde que aplicadas as técnicas corretas de compressão e arquitetura de sistemas.

APÊNDICE 3

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 2 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

HUGO RODRIGUES PESSONI

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante as quatro primeiras Semanas, escolhi **Otimização de Modelo VLMs** como tema da minha Residência. Revisei as técnicas de otimização e descobri uma área antiga e gigantesca. Foquei em uma subárea chamada de **Otimização por Compactação e** produzi um documento (residecia_hugo) que compila todas elas e como funcionam, através da análise de artigos, reviews, documentações e surveys. Além de compilar também tudo que foi feito e aprendido durante os Gates anteriores. Esse arquivo é separado por Gates.

Para a quinta Semana, coloquei prioridade para iniciar a parte prática e ver como funciona esse tema em termos de código, mas sem deixar de lado a leitura.

As atividades desenvolvidas durante a Semana:

- Na semana passada, foi feito um **diagrama dos principais frameworks** que realizam otimização de modelo. Como complemento adicionei novos frameworks e a popularidade de cada um deles, melhorando também a tabela de comparação (comparacao_frameworks_otm)
- Sempre busco alternativas de vídeo para me conceituar melhor a respeito de tópicos, melhorar o glossário, relembrar conceitos ou aprender algo novo. Então montei uma lista (vídeos_residência) de acompanhamento de vídeos semanais que vou assistindo e anotando tudo que entendo sobre eles no documento (resumo_videos_residencia_ia)
- Dado a trajetória até o momento, sobre a pesquisa e entendimento do que é otimização, depois otimização por compactação, escolha de modelos VLM, funcionamento de modelos VLM e possível comparação entre VLM e CV (tradicional) defini uma rota para estudo prático:
  - Implementação e Treinamento de VLM
  - Implementação do ViT (encoder)
  - Fine Tuning de Modelo VLM e Otimização de modelo VLM
  - Fine Tuning de Modelo CV e Otimização de Modelos CV
  - Comparativo entre modelos VLM e CV (quantizado)
  - Comparativo entre modelos VLM e CV (fine tuning)

- Realizei a primeira etapa da jornada pratica que é a Implementação e Treinamento de VLM com base no **tutorial feito por Uygar Kurt** (Engenheiro de Machine Learning) em que foi feito um acompanhamento dos ensinamentos dele, linha a linha e fazendo comentários relevantes durante o todo o código para melhor compreensão da lógica. train_VLM_residencia_ia.ipynb

- Foi feita a leitura de 2 artigos de comparação de modelos:
  - Benchmarking Vision-Language Models on Optical Character Recognition in Dynamic Video Environments - benchmark e comparar o desempenho de Modelos de Visão-Linguagem (VLMs) com sistemas tradicionais de Reconhecimento Óptico de Caracteres (OCR) baseados em Visão Computacional (CV).

  - Performance Comparison of Vision-Language Models In Image Classification - análise comparativa abrangente e baseada em métricas do desempenho de oito Modelos de Visão-Linguagem (VLMs) de última geração em tarefas de classificação de imagens puramente visuais.

  - Anotações sobre o artigo (resumos_artigos_gate_5)

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Analisando essa necessidade de saber diferentes estratégias de como realizar a otimização e a busca por conhecimento prático, como planejamento para essa próxima Semana pretendo:

- Realizar a busca de novos frameworks de modo a complementar a tabela e o diagrama.
- Realizar a busca de mais vídeos sobre os assuntos (VLM e Otimização) e fazer o resumo deles
- Fazer a leitura de dois artigos:
  - A Comparative Study of CNNs and Vision-Language Models for Chart Image Classification
  - Hidden in plain sight: VLMs overlook their visual representations
- Seguir com a jornada prática e implementar Vision Transformers (encoder) de modelo VLM “From Scratch”

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 9 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

HUGO RODRIGUES PESSONI

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Durante as cinco primeiras Semanas, optei por estudar a **Otimização de Modelos VLMs** como tema da minha Residência. Nessa jornada, foi feita uma revisão das técnicas de otimização e me deparei com uma área extensa e bastante consolidada. Minha abordagem foi focada em uma subárea chamada Otimização por Compactação, resultando na produção de um documento (residecia_hugo) que reúne todas as técnicas relevantes e explica seu funcionamento, com base na análise de artigos, revisões, documentações e surveys. Esse documento também compila tudo o que está sendo desenvolvido e aprendido durante os Gates anteriores, sendo organizado de acordo com cada um deles, inclusive as partes práticas que se iniciaram na semana passada.

Para a sexta Semana, foi feito uma continuação do que foi planejado na quinta semana: código, leitura de artigos, busca por frameworks e vídeos sobre o assunto.

As atividades desenvolvidas durante a Semana:

- Foi feita a leitura de 2 artigos propostos na semana passada:
  - Hidden in plain sight: VLMs overlook their visual representations - falha crítica em Modelos de Linguagem Visual (VLMs) *open-source.*
  - A Comparative Study of CNNs and Vision-Language Models for Chart Image Classification - avaliação do desempenho de CNNs especificamente treinadas em comparação com VLMs *pre-trained* para a tarefa de classificação de imagens de gráficos.
  - Anotações sobre os artigos (resumos_artigos_gate_6)
- Não consegui realizar a pesquisa aprofundada de novos frameworks, mas mapeie 8 novos para facilitar quando for realizar as anotações.
  (lista_frameworks_a_analisar)
- Foi feito mais uma rodada de vídeos para me conceituar melhor a respeito de tópicos, aumentar o glossário, relembrar conceitos ou aprender algo novo. Na mesma lista (vídeos_residência) há o

acompanhamento de vídeos semanais e tudo que anoto sobre eles no documento (resumo_videos_residencia_ia_gate6)

- Por fim e não menos importante, foi realizada a segunda etapa da "jornada prática" apresentada na semana passada indo agora para a **Implementação do ViT (encoder)** com base no tutorial Implement and Train ViT From Scratch for Image Recognition - PyTorch de Uygar Kurt, fazendo um acompanhamento e desenvolvimento linha a linha e realizando anotações importantes durante todo o código. implement_train_ViT_residencia_ia .ipynb

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Como planejamento para essa próxima Semana pretendo seguir com meu plano de:

1. Continuar a busca de novos frameworks para complementar a tabela e o diagrama.
2. Continuar a busca de mais vídeos sobre os assuntos (VLM e Otimização), fazer o resumo deles e adicionar na lista.
3. Fazer a leitura de 2 documentos:
   a. Vision Language Models (Better, Faster, Stronger) - um compilado sobre o que aconteceu de arquitetura e melhoria desde 2024 para VLMs.
   b. Rethinking VLMs and LLMs for image classification | Scientific Reports - investiga o desempenho de VLMs e LLMs (juntos e separados) para classificação de imagens.
4. Seguir com a jornada prática e implementar Fine Tuning de Modelo VLM e Otimização de modelo VLM

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Diagrama citado no Termo de Aceite de Entrega do dia 2 de outubro de 2025:

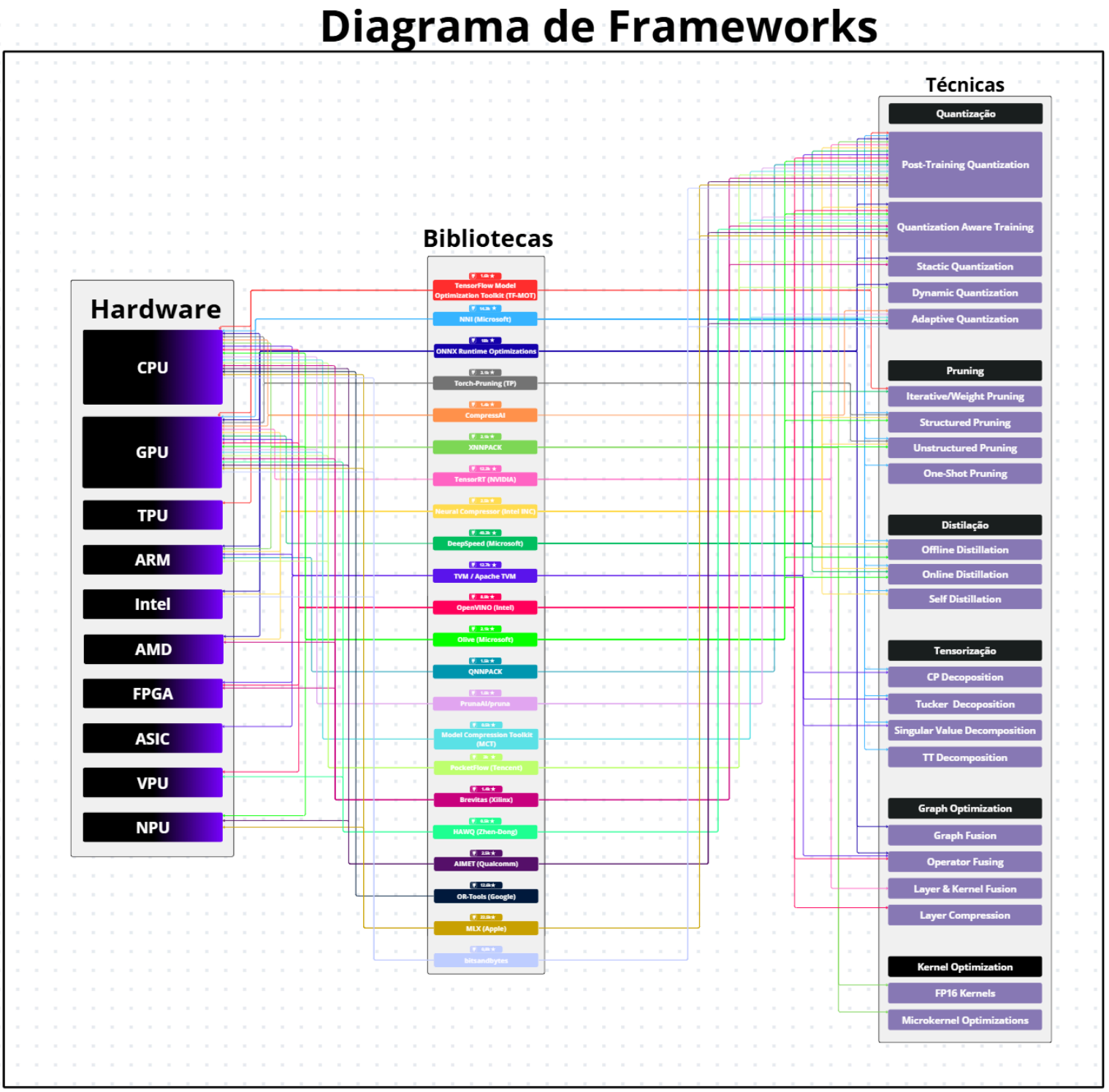


Figura 2:Diagrama de frameworks, tarefas e *hardware* adequado para cada um.

Sobre as etapas nos Termos de Aceite de Entrega do dia 2 e 9 de outubro de 2025:

- train_VLM_residencia_ia.ipynb
  - Foi usado o modelo Llama + ViT, ambos importados por meio de bibliotecas.
  - A ideia dessa parte prática não é ir do zero absoluto, implementando attention por exemplo, mas sim aproveitando o que está pronto para que possa agilizar o entendimento e poder explorar o funcionamento individual de cada parte.
  - Um dos objetivos é entender como os VLM funcionam e são treinados, criando um próprio através de um dataset fornecido.
  - O dataset utilizado é curto, apenas 16 imagens em forma de pares (texto+imagem), mas foi o suficiente para treinar rapidamente o modelo e ver alguns resultados.
  - Outro objetivo é conseguir colocar imagens no mesmo espaço vetorial de texto para um modelo LLM, para que assim ele consiga ler e treinar corretamente.
  - Ordem de processos: Image input → image encoder (ViT) → Projector (Linear) → Concatena com o texto (query) já tokenizado → envia para o modelo

- implement_train_ViT_residencia_ia .ipynb
  - Nesse tutorial foi utilizado como base o artigo: "An Image is Worth 16x16 Words: Transformers for Image Recgnition at Scale" - https://arxiv.org/abs/2010.11929
  - A ideia do artigo é basicamente pegar uma imagem e dividi-la em *patchs* (pedaços) e logo depois utilizar como alimentação para um *tranformers* do tipo *encoder*, fazendo com que as imagens (*pixels*) se tornem praticamente *tokens*.
  - Para saber o número de *patchs* de uma imagem, você pega a dimensão quadrática da sua imagem, divide pelo tamanho do *patch* e eleva tudo ao quadrado.
  - Sempre há um *CLS token* no começo do *transformer*, ele é aprendido e recebe uma classificação.

APÊNDICE 4

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 16 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

HUGO RODRIGUES PESSONI

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Resumo dos Stages até o momento:

- ➔ Gate 1: escolha do tema inicial e leitura de tópicos em conferências sugeridas. (**Otimização de modelos para sistemas embarcados**)
- ➔ Gate 2: busca por artigos relacionados à área para melhor entendimento do tema. (**Otimização por compactação para sistemas embarcados**) e início do documento residecia_hugo
- ➔ Gate 3: Leitura de artigos, blogs, documentação, fóruns, etc. Feito o levantamento da área e o quão grande ela pode ser. Estudo aprofundado sobre os VLMs. (**Otimização por compactação de modelos VLMs para sistemas embarcados)**
- ➔ Gate 4: Início do processo de busca direcionada de artigos voltados a comparação entre VLM e Visão Computacional (VC) tradicional. Além do levantamento de frameworks capazes de realizar otimização. (**Otimização de modelos VLMs**)
- ➔ Gate 5: Início do processo de prática, aprendendo como realizar a otimização dos modelos, com foco futuro em realizar uma comparação entre VLM e VC.
- ➔ Gate 6: Processo de treinamento de modelos VLMs, ainda sem a otimização com foco em entender o funcionamento na prática desse tipo de modelo.

Tema definido: **Otimização de modelos VLMs**

Para esta sétima Semana, continuei com o planejado: código, leitura de artigos, busca por frameworks.

As atividades desenvolvidas durante a Semana:

- Conclusão dos frameworks faltantes para a tabela de comparação entre eles (comparacao_frameworks_otm) e o complemento do diagrama (diagrama_frameworks).

- Leitura dos dois artigos propostos na semana passada:
  - Vision Language Models (Better, Faster, Stronger) - um compilado sobre o que aconteceu de arquitetura e melhoria desde 2024 para VLMs.

  - Rethinking VLMs and LLMs for image classification | Scientific Reports - investiga o

desempenho de VLMs e LLMs (juntos e separados) para classificação de imagens.

- Anotações e resumo sobre os artigos: resumos_artigos_gate_7

- Realização da terceira etapa da "jornada prática" definida em Gates anteriores realizando o Fine Tuning e otimização de um modelo VLM (Qwen2-VL-7B) voltado para interpretação de gráficos. fine_tune_vlm_residencia_ia.ipynb

- Em complemento à jornada prática, inicie a construção do dataset (dataset_graficos_residencia) a partir de dois datasets do Kaggle para o melhoramento do pior modelo (pior_modelo) apresentado pelo artigo A Comparative Study of CNNs and Vision-Language Models for Chart Image Classification por meio de Fine Tuning.

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Como planejamento para essa próxima Semana pretendo seguir com meu plano de:

- Realizar o melhoramento do pior modelo do artigo mencionado.
- Seguir com a jornada prática e implementar e realizar a otimização de um modelo VLM.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 23 de out. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

HUGO RODRIGUES PESSONI

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Resumo dos Stages até o momento:

- ➔ Gate 1 e 2: escolha do tema inicial, busca por artigos relacionados à área e criação do documento geral da residência residecia_hugo . (**Otimização de modelos para sistemas embarcados → Otimização por compactação para sistemas embarcados**)
- ➔ Gate 3 e 4: Estudo aprofundado sobre os VLMs e busca direcionada de artigos voltados a comparação entre VLM e Visão Computacional (VC) tradicional (**Otimização por compactação de modelos VLMs para sistemas embarcados → Otimização de modelos VLMs)**
- ➔ Gate 5 e 6: Processo com prática, aprendendo como realizar a otimização dos modelos, treinamento de um VLM e partes de um VLM também.
- ➔ Gate 7: "Conclusão" dos frameworks e diagrama de usabilidade (comparacao_frameworks_otm e diagrama_frameworks) e continuação na parte prática voltada ao Fine-tuning de um VLM.

As atividades desenvolvidas durante a Semana:

- Já que decidi tentar melhorar o pior modelo, busquei entender um pouco mais sobre ele (PaLI-GEMMA = Gemma 2B como decoder + SigLIP-So400m como encoder de imagem). (Sobre o PaliGemma)
- Troquei o dataset feito no Stage passado. (dataset_graficos_residencia)
- Adaptei esse novo dataset para o que seria utilizado no Fine-tuning, colocando no formato esperado pelo modelo, afinal o dataset no artigo é apenas imagem e target.
- Reproduzi o repositório do artigo, voltado apenas ao PaLI-GEMMA, eles testam 8 VLMs no total. Sendo todos testados em 4 prompts diferentes zero-shoting e obtive melhores resultados, sem mudar nada na pipeline, considerando que eles divulgam apenas os melhores resultados.
- Foi feito o Fine-tuning de 3 modelos:
  - mesmo modelo do artigo (**quantizado**): paligemma-3b-ft-vqav2-448
  - mesmo modelo do artigo **sem Fine-tuning** (**quantizado**): paligemma-3b-pt-448
  - modelo superior do mesmo tamanho (**não quantizado**): paligemma2-3b-pt-448

- Os resultado obtidos foram compilados em dois arquivos:
  - tabela comparativa residencia_comparacao_artigo
  - relatório e conclusões relatorio_gate_8

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Como planejamento para essa próxima Semana:
- Buscar realizar uma comparação de VLM e CV tradicional para a tarefa de classificação de imagens.
- Nessa comparação, quero trazer diferentes frameworks de quantização para ver as diferenças.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Agradeço ao Victor por recomendar uma plataforma mais barata que usarei com certeza daqui pra frente.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Resultado do Fine-tuning e a otimização de um modelo VLM (Qwen2-VL-7B) descrito no Termo de Aceite de Entrega do dia 16 de outubro de 2025:

```
generated_text, actual_answer = text_generator(sample_data) #fazer o teste de inferência novamente
print(f"Generated Answer: {generated_text}")
print(f"Actual Answer: {actual_answer}")
```

```
Prompt: <|im_start|>system
You are a highly advanced Vision Language Model (VLM), specialized in analyzing, describing, and interpreting visual data.
Your task is to process and extract meaningful insights from images, videos, and visual patterns,
leveraging multimodal understanding to provide accurate and contextually relevant information.<|im_end|>
<|im_start|>user
<|vision_start|><|image_pad|><|vision_end|>How many food item is shown in the bar graph?<|im_end|>
<|im_start|>assistant

------------------------------
Generated Answer: system
You are a highly advanced Vision Language Model (VLM), specialized in analyzing, describing, and interpreting visual data.
Your task is to process and extract meaningful insights from images, videos, and visual patterns,
leveraging multimodal understanding to provide accurate and contextually relevant information.
user
How many food item is shown in the bar graph?
assistant
There are 12 food items shown in the bar graph.
Actual Answer: 14
```

Figura 3: Resultado da última célula de um treinamento de modelo VLM para teste de VQA (Visual Question Answering).

Resultado obtidos modelo através da reprodução mais o Fine-tuning do pior modelo do artigo (A Comparative Study of CNNs and Vision-Language Models for Chart Image Classification) descrito no Termo de Aceite de Entrega do dia 23 de outubro de 2025:

Modelo	prompt type	# accuracy	# precision	# recall	# f1-score	Tr Classificação	OBS	OBS 2
Modelo 0		0,9682	0,9686	0,9682	0,9682	Melhor modelo de Visão	Vencedor do artigo	
Modelo 1	second	0,5050	0,5643	0,4856	0,4783	Resultado no artigo do pior modelo VLM	Resultado presente no artigo	
Modelo 1	first	0,4919	0,6469	0,5003	0,4896	Reprodução do artigo	Feito por mim em uma H100	
Modelo 1	second	0,5246	0,5678	0,5352	0,5073	Reprodução do artigo	Feito por mim em uma H100	
Modelo 1	third	0,2998	0,5352	0,3213	0,3537	Reprodução do artigo	Feito por mim em uma H100	
Modelo 1	fourth	0,2831	0,4312	0,2985	0,2740	Reprodução do artigo	Feito por mim em uma H100	
Modelo 2	second	0,5248	0,5674	0,5355	0,5076	Reprodução do artigo com Fine-tuning	O modelo aqui foi finetunado 2 vezes	sem FT no ViT
Modelo 3	second	0,4797	0,8400	0,4644	0,5491	Reprodução do artigo com Fine-tuning	Peguei a base do modelo e fiz o FT	sem FT no ViT
Modelo 4	second	0,5103	0,8106	0,5066	0,5350	Modelo mais novo sem quantização	Modelo gemma 2 pre-treinado e finetunado	sem FT no ViT
Modelo 2 - V2	second	0,6298	0,7119	0,7442	0,6861	Reprodução do artigo com Fine-tuning	O modelo aqui foi finetunado 2 vezes	com FT no ViT
Modelo 3 - V2	second	0,6956	0,8829	0,8311	0,8641	Reprodução do artigo com Fine-tuning	Peguei a base do modelo e fiz o FT	com FT no ViT
Modelo 4 - V2	second	0,7399	0,9250	0,8630	0,8813	Modelo mais novo sem quantização	Modelo gemma 2 pre-treinado e finetunado	com FT no ViT

Figura 4: Reprodução e melhoria dos resultados do artigo.

O melhor modelo obtido via Fine-Tuning apresentou desempenho **16% superior ao melhor VLM** do artigo (LLaVA-v1.5-13B) e ficou **20% abaixo do melhor modelo de Visão** (Xception), porém com diferenças **inferiores a 10% nas métricas** de precisão, recall e F1-score.

APÊNDICE 5

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 6 de nov. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

HUGO RODRIGUES PESSONI

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Resumo dos Stages até o momento:

- **Gate 1 e 2**:
  - Escolha do tema da Residência: Otimização de modelos para sistemas embarcados;
  - Levantamento e análise da área;
  - Criação do documento base do projeto (residecia_hugo);
- **Gate 3 e 4**:
  - Estudo aprofundado dos Visual Language Models (VLMs) e de suas aplicações;
  - Pesquisa direcionada a artigos que comparam VLMs com abordagens tradicionais de Visão Computacional (VC);
  - Tema refinado para Otimização de modelos VLMs;
- **Gate 5 e 6**:
  - Execução de técnicas de otimização de modelos;
  - Treinamento parcial de um modelo VLM e de partes específicas do modelo.
- **Gate 7 e 8**:
  - Conclusão da análise comparativa dos frameworks utilizados (comparacao_frameworks_otm e diagrama_frameworks);
  - Desenvolvimento da etapa prática voltada ao Fine-Tuning de um VLM, com o objetivo de melhorar o desempenho no artigo.

As atividades desenvolvidas durante a Semana:

- Comecei **atualizando todo o código** de treinamento do Fine-Tuning do PaLI-GEMMA e vale ressaltar que não foi uma simples alteração de parâmetros (False para True). Tive que reestruturar muita coisa por conta de limitações de memória. Então fui testando quantas camadas LoRa poderia **explorar das três principais** partes do modelo: LLM, ViT e Projector. Todos os modelos se encontram no Hugging Face (https://huggingface.co/PessoniHugo)

- Após alguns dólares e algumas horas, finalmente consegui **criar resultados "bons"**, mostrando a teoria apresentada no artigo-base: comparação entre modelos de Visão Computacional tradicional (com Fine-Tuning) e VLMs sem Fine-Tuning (avaliados apenas em zero-shot com prompts).

- No artigo eles destacam que a visão tradicional ganha dos VLMs, mas assumem que um **VLM Treinado (Fine-Tuning)** para uma tarefa específica pode performar melhor que qualquer outro modelo e era isso que eu queria comprovar. Lembrando que esse treinamento foi feito no próprio dataset do artigo com mais de 20.000 imagens e 25 classes e foi preciso criar um dataset de imagem+descrição.

- Os **resultados dessa comparação** foram organizados nessa planilha (residencia_comparacao_artigo) . O melhor modelo obtido via Fine-Tuning apresentou desempenho **16% superior ao melhor VLM** do artigo (LLaVA-v1.5-13B) e ficou **20% abaixo do melhor modelo de Visão** (Xception), porém com diferenças **inferiores a 10% nas métricas** de precisão, recall e F1-score.

- Dando continuidade, resolvi que iria **adotar uma estratégia semelhante** do artigo, porém com uma abordagem diferente. Enquanto o artigo reproduziu todos os testes em um cluster de A100, resolvi fazer em um **Orange Pi 3B, uma SBC de 2023**, com um hardware limitadíssimo.

- O primeiro passo foi relembrar o funcionamento dessas SBCs e **configurar o ambiente** de trabalho via SSH, pois o vscode gera muitos gargalos. Tive de considerar **diversas restrições**, como armazenamento interno, memória RAM, clock da CPU, arquitetura ARM e uso da NPU. A escolha de utilizar uma SBC leva a questionamentos como: **“será que esse modelinho consegue rodar?”**

- A tarefa escolhida para o teste foi a **contagem de objetos**. Essa tarefa **envolve tanto detecção e classificação quanto a contagem dos mesmos**, a partir da qual extrai-se as métricas MAE e MSE.

- O **dataset utilizado foi o COCO**, que contém mais de 300 mil imagens e 80 classes, mas foram usadas um total de cerca de 5 mil em 10 classes diferentes, criando subdatasets que foram analisados isoladamente. **(5000x6x3 = 90k**)

- Escolhi dois modelos de Visão muito utilizados: **YOLO e MobileNetV3**. Ambos já treinados no COCO e também optei por **quantizá-los em FP16 e INT8 utilizando a biblioteca ONNX** que possui uma compatibilidade ampla entre frameworks.

- Realizei um **breve estudo sobre esses dois modelos** e fiz algumas anotações (modelos_cv_residencia)

- Ao todo, **foram testadas 6 variações de modelos** (base em FP32, FP16 e INT8). Os resultados foram organizados nessa planilha (resultados_comparacao_residencia) e descritos nesse documento (relatorio_gate_9) que servirá de base para avaliar o desempenho dos modelos VLMs.

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Como planejamento para essa próxima Semana:

- O objetivo principal será **trazer os modelos VLMs para o mesmo dataset** utilizado e realizar uma comparação direta com os modelos de visão tradicional.
- Já foi feita uma lista de possíveis candidatos (lista_modelo_vlm_residencia)e será feita uma **seleção dos VLMs** considerando principalmente as restrições da Orange Pi.
- Será realizada a **quantização dos modelos VLMs** (em formatos como FP16 e INT8), seguindo a mesma metodologia aplicada aos modelos de visão.
- Quero buscar formas de **coleta de métricas de eficiência energética**, seja utilizando ferramentas de monitoramento de hardware da própria SBC ou com medições externas.
- Por fim, pretendo fazer uma **organização geral dos repositórios desenvolvidos até o momento** para facilitar a continuidade de experimentos futuros.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Sobre a Orange Pi 3B descrita no Termo de Aceite de Entrega do dia 6 de novembro de 2025:

Figura 5: Figura ilustrativa da SBC utilizada no processo.

SoC	Rockchip RK3566
CPU	Processador quad-core Cortex-A55 de 64 bits, processo avançado de 22 nm, até 1,8 GHz
GPU	• Processador gráfico ARM Mali G52 2EE • Compatível com OpenGL ES 1.1/2.0/3.2, OpenCL 2.0 e Vulkan 1.1 • Hardware integrado de aceleração 2D de alto desempenho
NPU	• Acelerador de IA RKNN NPU integrado com desempenho de 0,8 Tops@INT8 • Suporte para conversão com um clique de modelos de arquitetura Caffe/TensorFlow/TFLite/ONNX/PyTorch/Keras/Darknet
VPU	• Decodificação de vídeo H.265/H.264/VP9 em 4K a 60 fps • Codificação de vídeo H.265 em 1080p a 100 fps • Codificação de vídeo H.264 em 1080p a 60 fps
PMU	RockChip RK809-5
BATER	2GB/4GB/8GB (LPDDR4/4x)
Memória	• Suporte para módulo eMMC: 16 GB/32 GB/64 GB/128 GB/256 GB • Flash SPI: 16 MB/32 MB • Slot M.2 M-KEY (opcional): SSDs SATA3 ou PCIe 2.0 NVMe • Slot para cartão Micro SD

Wi-Fi + BT	Wi-Fi 5+BT 5.0,BLE(AP6256)
Ethernet	Ethernet 10/100/1000Mbps (Chip PHY integrado: YT8531C-CA)
Saída de vídeo	• 1* HDMI TX 2.0, até 4K a 60fps • 1* MIPI DSI de 2 pistas • eDP1.3
Câmera	1*Interface de câmera MIPI CSI de 2 pistas
USB	• 1 porta USB 2.0 com suporte para modo Dispositivo ou Host • 1 porta USB 3.0 Host • 2 portas USB 2.0 Host
Áudio	Entrada/saída de áudio para fone de ouvido de 3,5 mm
Botão	1 * tecla MaskROM, 1 * tecla RESET, 1 * tecla POWER
FÃ	Conector de ventoinha de 2 pinos, tamanho 1,25 mm, 5 V
RTC	Conector de backup de bateria de 2 pinos e 1,25 mm
40 pinos	Interface de expansão funcional de 40 pinos, compatível com os seguintes tipos de interface: GPIO, UART, I2C, SPI, PWM.
Fonte de alimentação	Tipo C 5V3A
Sistemas operacionais suportados	Android 11, Ubuntu 22.04, Ubuntu 20.04, Debian 11, Debian 12, OpenHarmony 4.0 Beta1, Orange Pi OS (Arch), Orange Pi OS (OH) baseado em OpenHarmony e outros sistemas operacionais.
Dimensões da placa de circuito impresso	56*89mm
Peso	52g

Sobre os testes realizados com os modelos de Visão Computacional Yolo e MobileNetV3 descrito no Termo de Aceite de Entrega do dia 6 de novembro de 2025:

model_name	model_type	threshold	total_gt_count	total_pred_count	mae_mean	mse_mean	total_time_mean/por subdataset	total_time_runtime (min)
mobilenetv3	fp16	0,3	15754	5874	1,628	8,600	3,572	35,715
mobilenetv3	fp32	0,3	15754	5741	1,625	8,623	3,696	28,387
mobilenetv3	int8	0,3	15754	5302	1,689	9,017	3,783	37,538
yolov8n	fp16	0,3	15754	20529	2,199	17,456	3,628	36,282
yolov8n	fp32	0,3	15754	11174	1,025	3,857	2,070	20,698
yolov8n	int8	0,3	15754	10866	1,059	3,969	2,340	23,399
mobilenetv3	fp16	0,5	15754	4514	1,851	10,143	3,559	36,963
mobilenetv3	fp32	0,5	15754	4465	1,862	10,198	2,894	26,751
mobilenetv3	int8	0,5	15754	4097	2,004	11,386	3,673	38,736
yolov8n	fp16	0,5	15754	16204	2,212	13,673	2,795	32,624
yolov8n	fp32	0,5	15754	8088	1,665	8,214	1,946	20,358
yolov8n	int8	0,5	15754	7773	1,758	9,127	2,571	24,569
mobilenetv3	fp16	0,7	15754	3705	1,851	10,143	3,559	37,831
mobilenetv3	fp32	0,7	15754	3693	1,862	10,198	2,894	27,205
mobilenetv3	int8	0,7	15754	3380	2,004	11,386	3,673	38,073
yolov8n	fp16	0,7	15754	11990	2,212	13,673	2,795	30,526
yolov8n	fp32	0,7	15754	5317	1,665	8,214	1,946	18,415
yolov8n	int8	0,7	15754	5070	1,758	9,127	2,571	25,427

Figura 6: Resultado da aplicação dos modelos de Visão Computacional no dataset COCO.

APÊNDICE 6

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("Gate") de aprovação: 12 de nov. de 2025

Participantes da Entrega [matriculados em Residência em IA]:

HUGO RODRIGUES PESSONI

Entrega: [descrever a ENTREGA - requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Resumo dos Stages até o momento:

- **Gate 1 e 2**:
  - Escolha do tema da Residência: Otimização de modelos para sistemas embarcados;
  - Levantamento e análise da área;
  - Criação do documento base do projeto (residecia_hugo);
- **Gate 3 e 4**:
  - Estudo aprofundado dos Vision-Language Models (VLMs) e de suas aplicações;
  - Pesquisa direcionada a artigos que comparam VLMs com abordagens tradicionais de Visão Computacional (VC);
  - Tema refinado para Otimização de modelos VLMs;
- **Gate 5 e 6**:
  - Execução de técnicas de otimização de modelos;
  - Treinamento parcial de um modelo VLM e de partes específicas do modelo.
- **Gate 7 e 8**:
  - Conclusão da análise comparativa dos frameworks utilizados (comparacao_frameworks_otm e diagrama_frameworks);
  - Desenvolvimento da etapa prática voltada ao Fine-Tuning de um VLM, com o objetivo de melhorar o desempenho do artigo base.
- **Gate 9 e 10:**
  - Execução de teste de inferência para contagem de objetos para comparação entre algoritmos de visão computacional e VLM.
  - Elaboração do relatório técnico sobre o experimento.

As atividades desenvolvidas durante a Semana:

- A primeira coisa que fiz foi **configurar a Orange Pi** para aceitar os novos modelos, limpando tudo o que pudesse ocupar espaço interno. Desde modelos de visão já usados, testes antigos e outros arquivos que poderiam ser descartados. O objetivo aqui era **otimizar os recursos da SBC (Single Board Computer)**.

- Utilizei a lista previamente montada no Gate anterior para me ajudar na escolha dos modelos VLMs

que seriam utilizados para a nova rodada de comparações. Aqui foi literalmente um jogo de “cabe ou não cabe” dentro das restrições da Orange Pi. Ao final, decidi testar modelos de tamanhos e complexidades diferentes:
    - Qwen2-VL-2B - **2 bilhões** de parâmetros
    - MobileVLM-v2 - **1,7 bilhões** de parâmetros
    - SmolVLM-500M - **500 milhões** de parâmetros
    - SmolVLM-256M - **256 milhões** de parâmetros

- A quantização dos modelos foi feita utilizando a técnica **post-trained quantization (PTQ)**, uma vez que é um método viável de ser executado em uma SBC do porte da Orange Pi.

- Infelizmente, não consegui coletar métricas de **eficiência energética**. Mesmo o software que monitora temperatura e consumo em Watts funcionando, não consegui automatizar a coleta desses dados durante as inferências. Até utilizei uma tomada inteligente para mostrar os consumo, mas não foi como o esperado, afinal eu estava medindo apenas o que entra na placa e por ser um baixo consumo de 12V, quase não variava esse valor.

- Sobre os resultados resultados_comparacao_residencia :
    - Foram avaliadas **30 configurações distintas de modelos**, variando arquitetura, tipo de quantização (FP32 (float32), FP16 (float16) e INT8) e thresholds de confiança. **No total, foram 18 “modelos” para visão e 12 “modelos” para VLM.**

    - Há grande trade-off entre velocidade e qualidade nos experimentos:
        - Se **buscar ambos** ao mesmo tempo acredito que seja inviável.
        - Se **priorizar velocidade**, modelos de visão com thresholds altos se mostram ideais
        - Se **priorizar qualidade e flexibilidade**, os VLMs se tornam a melhor opção.

    - O melhor **modelo de visão foi cerca de 900% mais rápido** que o melhor modelo VLM. Em contrapartida, o melhor modelo VLM apresentou **contagens corretas aproximadamente 300% superiores** às do modelo de visão.

- Essas diferenças se devem a diversos fatores: número de parâmetros, complexidade algorítmica, processos na inferência, gestão de memória e diferença de largura de banda.

- Conclusão:
    - Se o objetivo é fazer contagem ou detecção em **tempo real**, ou **processar um grande volume de imagens**, recomendo optar por modelos de visão tradicionais, pela leveza e velocidade.

    - Se o objetivo é obter **maior controle de qualidade, interpretação/resposta em linguagem natural e diversidade** de resultados **sem necessidade** de Fine-Tuning, ou retreinamento, então recomendo os modelos VLM. São capazes não apenas de detectar, mas também de descrever e contextualizar o conteúdo visual.

- Todo a conclusão do experimento, estudos sobre os modelos, análise de dados e comentários durante os testes **foram compilados no arquivo**: resumo_comparacao_residencia_ia

- Organizei o repositório conforme os Gates foram executados para facilitar implementações futuras ou mesmo continuidade no assunto. (https://github.com/HugoPessoni/residencia_em_ia)

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Em uma conversa de corredor, acabei adquirindo **uma nova SBC que possui um modelo de IA** capaz de acelerar inferências, tanto para modelos de visão, quanto modelos VLMs. Tenho a esperança de conseguir testá-la e ver se valeu a pena.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Sobre os testes realizados da comparação entre os modelos de Visão Computacional (Yolo e MobileNetV3) e VLM (Qwen2-VL-2B, MobileVLM-v2, SmolVLM-500M e SmolVLM-256M) descrito no Termo de Aceite de Entrega do dia 12 de novembro de 2025:

model_name	model_type	threshold	total_gt_count	total_pred_count	mae_mean	mse_mean	total_time_mean/por subdataset	total_time_runtime (min)
qwen2_vl_2b	fp16	-	15754	14098	0,345	0,699	19,425	194,245
qwen2_vl_2b	fp32	-	15754	14118	0,364	0,853	22,676	226,758
qwen2_vl_2b	int8	-	15754	13999	0,339	0,745	18,020	180,202
smolvlm_500m	fp16	-	15754	13913	1,330	2,770	10,032	100,322
smolvlm_500m	fp32	-	15754	13803	1,275	2,606	10,853	108,527
smolvlm_500m	int8	-	15754	13740	1,366	2,727	9,283	92,827
smolvlm_256m	fp16	-	15754	13550	2,086	4,380	7,693	76,926
smolvlm_256m	fp32	-	15754	13701	1,771	3,546	8,253	82,529
smolvlm_256m	int8	-	15754	13647	2,012	3,980	6,776	67,758
mobilevlm_v2	fp16	-	15754	13605	0,521	1,062	17,582	175,821
mobilevlm_v2	fp32	-	15754	13493	0,537	1,105	19,811	198,113
mobilevlm_v2	int8	-	15754	13575	0,533	1,022	9,842	98,419

Figura 7: Resultado da aplicação dos modelos VLM no dataset COCO.