

# Análise Comparativa de Estratégias de Otimização para Processamento de Imagens Hiperespectrais em Sistemas Heterogêneos

Diego Maia

7 de agosto de 2025

# **Agradecimientos**

# Resumo

Esta dissertação apresenta uma análise comparativa de estratégias de otimização para o processamento de imagens hiperespectrais em sistemas embarcados heterogêneos (CPU+GPU+FPGA). O objetivo é avaliar a eficácia da integração de técnicas de ponta, identificadas na literatura, para quantificar os trade-offs entre latência, consumo energético e precisão. Para isso, foi implementado um pipeline de processamento que combina compressive sensing, seleção de bandas EMCR e codesign HW/SW em uma arquitetura unificada. O desempenho do sistema integrado é comparado com benchmarks estabelecidos, visando validar o potencial de melhorias em cenários de aplicação real, como agricultura de precisão. A pesquisa contribui com uma análise quantitativa e diretrizes práticas para a implementação de sistemas hiperespectrais eficientes em plataformas heterogêneas.

**Palavras-chave:** Análise Comparativa, Otimização de Desempenho, Sistemas Heterogêneos, Processamento Hiperespectral, Análise de Trade-offs.

# Abstract

This dissertation presents a comparative analysis of optimization strategies for hyperspectral image processing on heterogeneous embedded systems (CPU+GPU+FPGA). The objective is to evaluate the effectiveness of integrating state-of-the-art techniques, identified from the literature, to quantify the trade-offs among latency, energy consumption, and accuracy. For this purpose, a processing pipeline was implemented, combining compressive sensing, EMCR band selection, and HW/SW codesign into a unified architecture. The performance of the integrated system is compared against established benchmarks to validate the potential for improvements in real-world application scenarios, such as precision agriculture. This research contributes a quantitative analysis and practical guidelines for implementing efficient hyperspectral systems on heterogeneous platforms.

**Keywords:** Comparative Analysis, Performance Optimization, Heterogeneous Systems, Hyperspectral Processing, Trade-off Analysis.

# Sumário

<b>Lista de Símbolos e Abreviações</b>	<b>9</b>
<b>1 Introdução</b>	<b>11</b>
1.1 Contextualização . . . . .	11
1.2 Caracterização do Problema . . . . .	11
1.2.1 Volume de Dados Excessivo . . . . .	11
1.2.2 Complexidade Computacional . . . . .	11
1.2.3 Restrições Energéticas . . . . .	12
1.3 Lacunas na Literatura . . . . .	12
1.4 Escopo da Pesquisa: Estruturação em Duas Etapas . . . . .	12
1.4.1 Etapa 1: Validação de Metodologias (Mestrado - 2025) . . . . .	12
1.4.2 Etapa 2: Arquitetura Otimizada (Doutorado - 2026-2029) . . . . .	13
1.5 Objetivos Específicos da Etapa 1 (Mestrado) . . . . .	13
1.6 Hipóteses de Pesquisa da Etapa 1 . . . . .	13
1.7 Contribuições Esperadas da Etapa 1 . . . . .	14
1.7.1 Contribuições Metodológicas . . . . .	14
1.7.2 Contribuições Experimentais . . . . .	14
1.8 Metodologia Geral da Etapa 1 . . . . .	14
1.9 Estrutura da Dissertação . . . . .	15
<b>2 Estado da Arte e Análise de Técnicas</b>	<b>16</b>
2.1 Visão Geral do Estado da Arte . . . . .	16
2.1.1 Categorização dos Trabalhos Analisados . . . . .	16
2.2 Trabalhos de Muito Alta Relevância . . . . .	16
2.2.1 Compressive Sensing Hiperespectral Embarcado . . . . .	16
2.2.2 Codesign HW/SW para Compressão Lossless . . . . .	17
2.2.3 Compressão Real-Time em GPUs Embarcadas . . . . .	17
2.2.4 Acelerador SVM Hardware para Classificação . . . . .	18
2.3 Trabalhos de Alta Relevância . . . . .	18
2.3.1 Revisões Sistemáticas e Estado da Arte . . . . .	18
2.3.2 Arquiteturas Ultra-Eficientes . . . . .	18
2.3.3 Benchmarks em Plataformas Embarcadas . . . . .	19
2.4 Técnicas de Otimização Identificadas . . . . .	19
2.4.1 Estratégias de Redução de Consumo Energético . . . . .	19
2.4.2 Estratégias de Redução de Latência . . . . .	19
2.5 Gaps e Oportunidades na Literatura . . . . .	19
2.5.1 Ausência de Frameworks Integrados . . . . .	19
2.5.2 Limitações em Validação Prática . . . . .	19

2.5.3	Trade-offs Não Quantificados . . . . .	20
2.5.4	Metodologias de Codesign Limitadas . . . . .	20
2.6	Benchmarks e Métricas de Referência . . . . .	20
2.6.1	Performance Estabelecida . . . . .	20
2.6.2	Datasets de Validação . . . . .	20
2.7	Síntese e Direcionamentos . . . . .	20
<b>3</b>	<b>Metodologia de Validação</b>	<b>22</b>
3.1	Visão Geral da Metodologia de Validação . . . . .	22
3.2	Framework Arquitetural Conceitual . . . . .	22
3.2.1	Modelo de Sistema Heterogêneo . . . . .	22
3.2.2	Plataforma de Referência para Modelagem . . . . .	22
3.2.3	Estudo de Caso Industrial: SightLine Applications . . . . .	23
3.3	Fase 1: Análise Sistemática do Estado da Arte (2 meses) . . . . .	24
3.3.1	Catologação de Técnicas Comprovadas . . . . .	24
3.3.2	Estabelecimento de Baselines . . . . .	24
3.4	Fase 2: Modelagem e Simulação Conceitual (3 meses) . . . . .	24
3.4.1	Modelagem Teórica das Técnicas . . . . .	24
3.4.2	Análise de Sensibilidade . . . . .	25
3.5	Fase 3: Protótipos de Prova de Conceito (3 meses) . . . . .	25
3.5.1	Implementação de Protótipos Simplificados . . . . .	25
3.5.2	Validação Experimental . . . . .	26
3.6	Fase 4: Framework Arquitetural e Diretrizes (2 meses) . . . . .	26
3.6.1	Consolidação dos Resultados . . . . .	26
3.6.2	Diretrizes para a Etapa 2 . . . . .	27
3.7	Cronograma de Execução . . . . .	27
3.7.1	Visão Geral do Cronograma . . . . .	27
3.7.2	Etapa 1: Fundamentação Teórica e Estado da Arte (Agosto 2025 - Janeiro 2026) . . . . .	29
3.7.3	Etapa 2: Desenvolvimento Experimental e Validação (Janeiro - Maio 2026) . . . . .	29
3.7.4	Etapa 3: Análise de Resultados e Redação (Maio - Agosto 2026) . . . . .	30
3.7.5	Etapa 4: Finalização e Preparação para Defesa (Agosto - Setembro 2026) . . . . .	31
3.7.6	Milestones Principais . . . . .	31
3.7.7	Controle e Acompanhamento . . . . .	31
3.8	Metodologia de Análise dos Resultados . . . . .	32
3.8.1	Análise Estatística . . . . .	32
3.8.2	Comparação com Estado da Arte . . . . .	32
<b>4</b>	<b>Validação Experimental e Resultados</b>	<b>33</b>
4.1	Resultados Experimentais . . . . .	33
4.1.1	Performance do Sistema . . . . .	33
4.1.2	Eficiência Energética . . . . .	33
4.1.3	Precisão de Classificação . . . . .	33
4.2	Análise Comparativa . . . . .	33
4.2.1	Comparação com Literatura . . . . .	33
4.2.2	Validação das Hipóteses . . . . .	33
4.3	Aplicações Práticas . . . . .	33

4.3.1	Agricultura de Precisão . . . . .	33
4.3.2	Monitoramento Ambiental . . . . .	33
<b>5</b>	<b>Conclusões e Continuidade</b>	<b>34</b>
5.1	Conclusões Principais . . . . .	34
5.1.1	Objetivos Atingidos . . . . .	34
5.1.2	Hipóteses Validadas . . . . .	34
5.2	Contribuições da Pesquisa . . . . .	34
5.2.1	Contribuições Técnicas . . . . .	34
5.2.2	Contribuições Científicas . . . . .	34
5.3	Trabalhos Futuros: Continuidade na Etapa 2 (Doutorado) . . . . .	34
5.3.1	Etapa 2: Arquitetura Otimizada Integrada (2026-2029) . . . . .	34
5.3.2	Objetivos Quantitativos da Etapa 2 . . . . .	35
5.3.3	Impacto Científico e Tecnológico Esperado . . . . .	36
5.3.4	Pesquisas de Longo Prazo (Pós-Doutorado) . . . . .	36
5.4	Considerações Finais . . . . .	36
<b>A</b>	<b>Especificações Técnicas da Plataforma</b>	<b>39</b>
A.1	Hardware Utilizado . . . . .	39
A.2	Software e Ferramentas . . . . .	39
<b>B</b>	<b>Datasets Utilizados</b>	<b>40</b>
B.1	AVIRIS Indian Pines . . . . .	40
B.2	Pavia University . . . . .	40
B.3	Salinas Valley . . . . .	40
<b>C</b>	<b>Códigos e Implementações</b>	<b>41</b>
C.1	Scripts de Compilação . . . . .	41
C.2	Parâmetros de Configuração . . . . .	41
<b>D</b>	<b>Resultados Complementares</b>	<b>42</b>
D.1	Análises Estatísticas . . . . .	42
D.2	Dados de Performance . . . . .	42

# Lista de Figuras

3.1 Cronograma detalhado do projeto de mestrado (Agosto 2025 - Setembro 2026).  
O gráfico mostra as quatro etapas principais, 12 tarefas detalhadas e 4 milestones críticos. A linha vermelha pontilhada indica a data atual, permitindo acompanhar o progresso do projeto. As cores representam as diferentes etapas: verde (Fundamentação Teórica), azul (Desenvolvimento Experimental), laranja (Análise e Redação) e rosa (Finalização). . . . . 28



## **Lista de Tabelas**

# Lista de Símbolos e Abreviações

## Abreviações

**AVIRIS** Airborne Visible/Infrared Imaging Spectrometer

**CGNE** Conjugate Gradient for Normal Equation

**CNN** Convolutional Neural Network

**CPU** Central Processing Unit

**CS** Compressive Sensing

**CUDA** Compute Unified Device Architecture

**DMA** Direct Memory Access

**ELM** Empirical Line Method

**EMCR** Entropy Multiple Correlation Ratio

**FP16** 16-bit Floating Point

**FP32** 32-bit Floating Point

**FPGA** Field-Programmable Gate Array

**fps** Frames Per Second

**GPU** Graphics Processing Unit

**HSI** Hyperspectral Image

**HW/SW** Hardware/Software

**LULC** Land Use/Land Cover

**NVIDIA** NVIDIA Corporation

**RGB** Red, Green, Blue

**SIMD** Single Instruction, Multiple Data

**SVM** Support Vector Machine

**UAV** Unmanned Aerial Vehicle

**VHDL** VHSIC Hardware Description Language

## Símbolos Matemáticos

$n$	Número de bandas espectrais
$m$	Número de pixels na imagem
$\mathbf{x}$	Vetor de dados originais
$\mathbf{y}$	Vetor de dados comprimidos
$\Phi$	Matriz de sensoriamento (sensing matrix)
$\mathbf{A}$	Matriz do sistema linear
$\lambda$	Comprimento de onda
$\rho$	Reflectância espectral
$\sigma$	Desvio padrão
$\mu$	Média
$\epsilon$	Erro de reconstrução
$\eta$	Eficiência energética
$\tau$	Latência de processamento
$P$	Potência consumida
$T$	Throughput (taxa de processamento)

## Unidades

<b>W</b>	Watt (potência)
<b>ms</b>	Milissegundo (tempo)
<b>fps</b>	Quadros por segundo (taxa)
<b>MB/s</b>	Megabytes por segundo (bandwidth)
<b>GFLOPs</b>	Giga operações de ponto flutuante por segundo
<b>nm</b>	Nanômetro (comprimento de onda)
<b>°C</b>	Graus Celsius (temperatura)
<b>%</b>	Porcentagem (precisão, utilização)

# Capítulo 1

## Introdução

### 1.1 Contextualização

O processamento de imagens hiperespectrais representa uma das fronteiras mais desafiadoras na área de sensoriamento remoto e visão computacional. Estas imagens, compostas por centenas de bandas espectrais contínuas que capturam informações detalhadas sobre a reflectância dos materiais, oferecem capacidades de análise sem precedentes para aplicações em agricultura de precisão, monitoramento ambiental, vigilância e exploração mineral (Lou *et al.*, 2024). No entanto, a riqueza espectral destas imagens vem acompanhada de desafios computacionais significativos, especialmente quando o processamento deve ser realizado em sistemas embarcados com restrições de energia, memória e capacidade de processamento.

A demanda por processamento hiperespectral em tempo real tem crescido exponencialmente devido ao avanço de plataformas móveis como UAVs (Unmanned Aerial Vehicles) e satélites CubeSat (Zhang *et al.*, 2024). Estas aplicações exigem que algoritmos tradicionalmente executados em servidores de alto desempenho sejam adaptados para ambientes com severas limitações de recursos. O paradigma de edge computing se torna, portanto, essencial para viabilizar aplicações práticas onde a latência de comunicação com centros de processamento remotos é proibitiva ou onde a autonomia energética é crítica.

### 1.2 Caracterização do Problema

O processamento de imagens hiperespectrais em sistemas embarcados enfrenta três desafios fundamentais interconectados:

#### 1.2.1 Volume de Dados Excessivo

Uma única imagem hiperespectral típica do sensor AVIRIS contém  $614 \times 512 \times 224$  pixels/bandas, resultando em aproximadamente 69 milhões de valores espectrais por frame (Green, 2020). Para aplicações em tempo real que exigem processamento de 25-30 fps, isto representa um throughput de dados brutos superior a 2 GB/s, excedendo significativamente a capacidade de processamento e bandwidth de memória de sistemas embarcados típicos.

#### 1.2.2 Complexidade Computacional

Os algoritmos de processamento hiperespectral apresentam complexidades computacionais elevadas. Técnicas de classificação baseadas em machine learning tradicionais possuem com-

plexidade  $O(n^2)$  a  $O(n^3)$ , onde  $n$  representa o número de bandas espectrais (Lim *et al.*, 2022). Para métodos de compressive sensing, amplamente utilizados para redução de dados, o algoritmo CGNE (Conjugate Gradient for Normal Equation) apresenta complexidade  $O(n^3)$  nas operações matriciais, tornando-se computacionalmente proibitivo para implementações embarcadas diretas.

### 1.2.3 Restrições Energéticas

Sistemas embarcados típicos operam com orçamentos energéticos de 5-20W, contrastando drasticamente com os 200-1000W consumidos por servidores de alto desempenho tradicionalmente utilizados para processamento hiperespectral (Hwang *et al.*, 2011). Esta limitação torna-se crítica em aplicações UAV onde a autonomia de voo é diretamente afetada pelo consumo energético dos sistemas de processamento.

## 1.3 Lacunas na Literatura

A análise sistemática de 20 artigos científicos recentes revelou quatro lacunas principais na literatura atual (Lou *et al.*, 2024; Zhang *et al.*, 2024):

1. **Ausência de Frameworks Integrados:** Embora existam soluções pontuais para otimização específica (GPU, FPGA, ou algoritmos isolados), não há frameworks que integrem sistematicamente múltiplas técnicas de otimização em sistemas heterogêneos.
2. **Falta de Validação Prática:** A maioria dos trabalhos limita-se a validações em datasets sintéticos ou de laboratório, sem demonstração em aplicações práticas com restrições reais de tempo e energia.
3. **Trade-offs Não Quantificados:** As relações entre precisão, consumo energético e latência não são adequadamente caracterizadas, dificultando a seleção de técnicas apropriadas para aplicações específicas.
4. **Metodologias de Codesign Limitadas:** Embora o codesign HW/SW seja reconhecido como fundamental (Hwang *et al.*, 2011), faltam metodologias sistemáticas para particionamento otimizado entre CPU, GPU e FPGA em aplicações hiperespectrais.

## 1.4 Escopo da Pesquisa: Estruturação em Duas Etapas

Esta pesquisa está estruturada em duas etapas complementares, onde a presente dissertação de mestrado representa a primeira etapa focada na validação conceitual e metodológica, preparando o terreno para uma segunda etapa de implementação completa durante o doutorado.

### 1.4.1 Etapa 1: Validação de Metodologias (Mestrado - 2025)

A primeira etapa, objeto desta dissertação de mestrado, concentra-se na **validação conceitual e metodológica** das estratégias de integração de sistemas heterogêneos para processamento hiperespectral embarcado. Esta etapa visa estabelecer a base teórica e experimental necessária para o desenvolvimento futuro de uma arquitetura otimizada.

**Objetivo Principal da Etapa 1:** Validar e quantificar o potencial de integração de técnicas comprovadas (compressive sensing, seleção EMCR, codesign HW/SW) em sistemas heterogêneos para processamento hiperspectral embarcado, gerando análises detalhadas do estado da arte e estabelecendo metodologias de avaliação para orientar o desenvolvimento futuro.

#### 1.4.2 Etapa 2: Arquitetura Otimizada (Doutorado - 2026-2029)

A segunda etapa, a ser desenvolvida durante o doutorado, focará na **proposição e implementação de uma arquitetura otimizada** baseada nos resultados e validações obtidos na primeira etapa. Esta fase concentrar-se-á no desenvolvimento prático e na inovação arquitetural.

**Objetivo da Etapa 2:** Propor e implementar uma arquitetura de sistema heterogêneo integrado (CPU+GPU+FPGA) completamente otimizada para redução simultânea de consumo energético e latência no processamento hiperspectral embarcado, baseada nas metodologias validadas na Etapa 1.

### 1.5 Objetivos Específicos da Etapa 1 (Mestrado)

1. **Realizar análise sistemática do estado da arte** em processamento hiperspectral embarcado, identificando e catalogando técnicas comprovadas de otimização energética e redução de latência.
2. **Caracterizar quantitativamente** os trade-offs entre precisão, consumo energético e latência através de simulações e protótipos conceituais de algoritmos hiperspectrais embarcados.
3. **Desenvolver metodologia de avaliação** para sistemas heterogêneos, estabelecendo métricas, benchmarks e protocolos de teste para orientar futuras implementações.
4. **Validar conceitos fundamentais** através de implementações de prova de conceito das técnicas mais promissoras: compressive sensing, seleção EMCR, e codesign básico HW/SW.
5. **Propor framework arquitetural** para integração sistemática de técnicas em sistemas heterogêneos, definindo especificações e diretrizes para a implementação completa na Etapa 2.
6. **Estabelecer baseline experimental** através de testes com datasets padrão (AVIRIS, Indian Pines, Pavia) para quantificar o potencial de melhoria e orientar o desenvolvimento futuro.

### 1.6 Hipóteses de Pesquisa da Etapa 1

Com base na análise bibliográfica sistemática, três hipóteses principais orientam a validação metodológica desta primeira etapa:

**H1:** A análise sistemática de técnicas comprovadas (compressive sensing com redução 50-70% dos dados (Lim *et al.*, 2022), seleção EMCR com redução 80% do processamento (Martins *et al.*, 2019), e codesign HW/SW com melhoria energética 43.5x (Hwang *et al.*, 2011)) pode

demonstrar, através de simulações e protótipos conceituais, o potencial teórico de redução energética superior a 20x em sistemas hiperespectrais embarcados.

**H2:** É possível estabelecer, através de modelagem e validação conceitual, que um framework arquitetural heterogêneo com módulos especializados (FPGA para pré-processamento, GPU para reconstrução, CPU para classificação) pode teoricamente atingir metas de latência <50ms/frame mantendo precisão >95%, fornecendo diretrizes quantitativas para implementação futura.

**H3:** Uma metodologia sistemática de avaliação e caracterização pode identificar e quantificar os trade-offs fundamentais entre precisão, consumo e latência, estabelecendo um framework de decisão para orientar a seleção e integração de técnicas em diferentes cenários de aplicação na Etapa 2.

## 1.7 Contribuições Esperadas da Etapa 1

Esta dissertação visa contribuir para o estado da arte em processamento hiperespectral embarcado através de:

### 1.7.1 Contribuições Metodológicas

- **Análise Sistemática Abrangente:** Primeira caracterização quantitativa completa das técnicas de otimização disponíveis, organizando o conhecimento disperso na literatura em um framework coerente.
- **Metodologia de Avaliação:** Desenvolvimento de protocolos, métricas e benchmarks padronizados para comparação sistemática de abordagens heterogêneas em processamento hiperespectral embarcado.
- **Framework Arquitetural Conceitual:** Proposição de diretrizes e especificações técnicas para integração de técnicas em sistemas heterogêneos, estabelecendo a base teórica para implementações futuras.

### 1.7.2 Contribuições Experimentais

- **Validação Conceitual:** Demonstração do potencial teórico através de simulações e protótipos de prova de conceito das técnicas mais promissoras identificadas na literatura.
- **Caracterização Quantitativa:** Análise detalhada dos trade-offs precisão vs consumo vs latência usando datasets padrão (AVIRIS, Indian Pines, Pavia University), estabelecendo baselines para comparações futuras.
- **Diretrizes de Implementação:** Definição de especificações técnicas, requisitos de hardware e estratégias de integração para orientar o desenvolvimento da arquitetura otimizada na Etapa 2.

## 1.8 Metodologia Geral da Etapa 1

A pesquisa da Etapa 1 seguirá uma abordagem de validação conceitual estruturada em quatro fases principais:

**Fase 1 - Análise Sistemática do Estado da Arte** (2 meses): Revisão abrangente da literatura, catalogação de técnicas comprovadas, identificação de lacunas e estabelecimento de um framework de conhecimento organizado.

**Fase 2 - Modelagem e Simulação Conceitual** (3 meses): Desenvolvimento de modelos teóricos, simulações das técnicas mais promissoras e quantificação do potencial de melhoria através de análises computacionais.

**Fase 3 - Protótipos de Prova de Conceito** (3 meses): Implementação de protótipos simplificados das técnicas principais (compressive sensing, seleção EMCR, codesign básico) para validação experimental dos conceitos fundamentais.

**Fase 4 - Framework Arquitetural e Diretrizes** (2 meses): Consolidação dos resultados em um framework arquitetural conceitual, definição de especificações técnicas e estabelecimento de diretrizes para a implementação completa na Etapa 2.

## 1.9 Estrutura da Dissertação

Esta dissertação está organizada de forma direta e objetiva em cinco capítulos focados na validação metodológica:

**Capítulo 2 - Estado da Arte e Análise de Técnicas** apresenta a análise sistemática da literatura, catalogação de técnicas comprovadas e identificação de lacunas para sistemas heterogêneos em processamento hiperespectral embarcado.

**Capítulo 3 - Metodologia de Validação** detalha a abordagem de validação conceitual, incluindo modelagem teórica, protocolo de simulação e framework de avaliação para caracterização de trade-offs.

**Capítulo 4 - Validação Experimental e Resultados** apresenta os protótipos de prova de conceito, simulações das técnicas mais promissoras, análise quantitativa dos trade-offs e estabelecimento de baselines experimentais.

**Capítulo 5 - Conclusões e Continuidade** sintetiza as contribuições metodológicas da Etapa 1, valida as hipóteses de pesquisa e estabelece as diretrizes técnicas para a implementação da arquitetura otimizada na Etapa 2 (doutorado).

Esta estrutura enxuta garante uma apresentação direta dos objetivos da validação metodológica, maximizando a objetividade e facilitando a transição para a fase de implementação prática no doutorado.



## Capítulo 2

# Estado da Arte e Análise de Técnicas

### 2.1 Visão Geral do Estado da Arte

Esta revisão bibliográfica baseia-se na análise sistemática de 20 artigos científicos selecionados por sua relevância para processamento hiperespectral embarcado, organizados por nível de contribuição direta ao problema de pesquisa. A metodologia de seleção priorizou trabalhos publicados entre 2019-2024 que demonstram técnicas comprovadas de otimização energética e redução de latência em sistemas embarcados.

#### 2.1.1 Categorização dos Trabalhos Analisados

Os artigos foram classificados em três categorias principais:

**Muito Alta Relevância:** 4 artigos que demonstram implementações práticas de otimização em sistemas embarcados com métricas quantitativas de performance.

**Alta Relevância:** 4 artigos que fornecem revisões sistemáticas abrangentes ou técnicas específicas aplicáveis ao problema.

**Relevância Média:** 12 artigos que contribuem com técnicas complementares, fundamentos teóricos ou validações específicas.

### 2.2 Trabalhos de Muito Alta Relevância

#### 2.2.1 Compressive Sensing Hiperespectral Embarcado

Lim et al. (Lim *et al.*, 2022) apresentam o primeiro estudo sistemático sobre viabilidade de compressive sensing (CS) hiperespectral em sistemas embarcados para aplicações em tempo real. O trabalho foca no algoritmo CGNE (Conjugate Gradient for Normal Equation) implementado em sistema DDCASSI, estabelecendo uma metodologia de análise que se tornou referência na área.

**Contribuições Principais:**

- Análise teórica de complexidade  $O(n^3)$  do CGNE com caracterização detalhada de requisitos de memória e bandwidth
- Demonstração de redução de 50-70% no volume de dados sem perda significativa de qualidade

- Identificação da sparsity da matriz do sistema como fator fundamental para redução computacional
- Estabelecimento de latência <10ms como viável para reconstrução de imagens pequenas
- Confirmação de que GPUs/FPGAs são necessárias para atingir performance de tempo real (25 fps)

**Relevância para a Pesquisa:** O framework de análise proposto pelos autores fornece a base metodológica para avaliar viabilidade de algoritmos embarcados, enquanto as técnicas de exploração de sparsity são diretamente aplicáveis ao módulo GPU do sistema proposto.

### 2.2.2 Codesign HW/SW para Compressão Lossless

Hwang et al. (Hwang *et al.*, 2011) demonstram uma implementação exemplar de codesign HW/SW para compressão lossless hiperspectral, estabelecendo metodologia sistemática que resultou em melhorias de eficiência energética de 43.5x comparado a servidores convencionais.

#### **Metodologia e Resultados:**

- Profiling sistemático identificou predição inter-banda como responsável por 91% do tempo computacional
- Particionamento inteligente HW/SW baseado em análise quantitativa de gargalos
- Implementação FPGA Spartan3 XC3S4000 atingiu throughput de 16.5 M pixels/s
- Speedup de 21x comparado à implementação puramente software
- Otimizações sistemáticas: memória hierárquica, DMA, e opções de compiler específicas

**Relevância para a Pesquisa:** A metodologia de profiling e particionamento HW/SW proposta é diretamente aplicável ao desenvolvimento do sistema heterogêneo, fornecendo tanto o framework de análise quanto as técnicas específicas de otimização.

### 2.2.3 Compressão Real-Time em GPUs Embarcadas

Díaz et al. (Díaz *et al.*, 2019) implementaram o algoritmo HyperLCA em GPUs embarcadas NVIDIA Jetson, demonstrando pela primeira vez compressão lossy hiperspectral em tempo real em plataformas embarcadas com validação em aplicação real de agricultura.

#### **Implementação e Performance:**

- Análise detalhada revelou que HyperLCA Transform consome >95% do processamento total
- Implementação CUDA com 7 kernels especializados para arquiteturas Kepler (TK1) e Pascal (TX2)
- Performance de 330 fps para dados de 144.375 MB/s com taxa de compressão >20:1
- Speedup de 21x sobre implementação CPU com eficiência energética 4.1x superior
- Validação prática: câmera 224 bandas, 1024 pixels, aplicação agricultura UAV

**Relevância para a Pesquisa:** As técnicas de implementação CUDA e a metodologia de análise de FLOPs são diretamente aplicáveis ao módulo GPU, enquanto a validação em agricultura UAV fornece benchmarks práticos para comparação.

## 2.2.4 Acelerador SVM Hardware para Classificação

Martins et al. (Martins *et al.*, 2019) desenvolveram um acelerador hardware SVM especializado para classificação onboard, demonstrando tempo de classificação de 0.1ms/pixel com 99.7% de precisão usando técnicas de seleção inteligente de bandas.

### Técnicas e Resultados:

- Implementação do método EMCR (Entropy Multiple Correlation Ratio) para seleção de bandas, reduzindo 80% do processamento
- Classificador SVM com distância de Hamming para decisão ultrarrápida em hardware
- Duas arquiteturas FPGA: single-core (17% DSPs) e hexa-core (100% DSPs)
- Speedup de 13x comparado à implementação MATLAB
- Processamento de dataset AVIRIS completo em tempo real

**Relevância para a Pesquisa:** O método EMCR é fundamental para o módulo FPGA proposto, enquanto a implementação SVM otimizada fornece a base para o módulo CPU de classificação.

## 2.3 Trabalhos de Alta Relevância

### 2.3.1 Revisões Sistemáticas e Estado da Arte

Lou et al. (Lou *et al.*, 2024) conduziram uma revisão sistemática de 200+ artigos sobre classificação LULC usando imagens hiperespectrais, identificando três gerações de métodos e estabelecendo que deep learning representa o estado da arte atual. A revisão identifica que arquiteturas híbridas (CNN+Transformer) mostram resultados promissores, enquanto o processamento em tempo real permanece como desafio técnico significativo.

O trabalho de Zhang et al. (Zhang *et al.*, 2024) foca especificamente em classificação hiperespectral para UAV, identificando desafios únicos como ruído dinâmico, pixels mistos e variabilidade temporal. Os autores demonstram que CNNs 3D são superiores para dados hiperespectrais UAV quando combinadas com mecanismos de atenção.

### 2.3.2 Arquiteturas Ultra-Eficientes

O TakuNet (Nakamura *et al.*, 2025) representa um avanço significativo em eficiência computacional, apresentando arquitetura CNN com apenas 37.685 parâmetros (redução de 99%) que mantém F1-score de 0.943. A arquitetura utiliza depth-wise convolutions e early downsampling, atingindo >650 fps no Jetson Orin Nano com 21x redução no consumo energético.

### Técnicas Aplicáveis:

- Depth-wise convolutions para redução drástica de parâmetros
- Early downsampling para redução computacional inicial
- Treinamento FP16 para otimização em aceleradores embarcados
- Conexões densas para convergência rápida

### 2.3.3 Benchmarks em Plataformas Embarcadas

Ullah et al. (Ullah *et al.*, 2020) fornecem benchmark sistemático de plataformas NVIDIA Jetson (Nano, TX1, Xavier) para deep learning, demonstrando que o Xavier oferece melhor performance geral e que TensorFlow-GPU utiliza recursos mais eficientemente que implementações CPU.

## 2.4 Técnicas de Otimização Identificadas

### 2.4.1 Estratégias de Redução de Consumo Energético

A análise sistemática identificou quatro estratégias principais comprovadas:

1. **Codesign HW/SW:** Hwang et al. demonstraram melhoria de 43.5x na eficiência energética através de particionamento sistemático e implementação FPGA de módulos críticos.
2. **Seleção Inteligente de Bandas:** O método EMCR de Martins et al. reduz 80% do processamento mantendo 99.7% de precisão.
3. **Precisão Reduzida:** Múltiplos trabalhos confirmam que FP16 oferece redução de 50% no uso de memória com perda <1% na precisão.
4. **Compressive Sensing:** Lim et al. demonstraram redução de 50-70% no volume de dados com reconstrução viável em <10ms.

### 2.4.2 Estratégias de Redução de Latência

Três abordagens principais foram identificadas:

1. **Paralelização Massiva:** Díaz et al. atingiram 330 fps através de implementação CUDA otimizada com múltiplos níveis de paralelismo.
2. **Aceleradores Especializados:** Martins et al. demonstraram 0.1ms/pixel usando FPGA especializada para classificação SVM.
3. **Algoritmos Otimizados:** CGNE para compressive sensing e HyperLCA para compressão demonstraram ser adequados para implementação embarcada.

## 2.5 Gaps e Oportunidades na Literatura

### 2.5.1 Ausência de Frameworks Integrados

Embora existam soluções pontuais eficazes (GPU (Díaz *et al.*, 2019), FPGA (Hwang *et al.*, 2011), algoritmos específicos (Lim *et al.*, 2022)), não há frameworks que integrem sistematicamente múltiplas técnicas de otimização. Esta lacuna representa a principal oportunidade para contribuição original.

### 2.5.2 Limitações em Validação Prática

A maioria dos trabalhos limita-se a validações em datasets sintéticos. Apenas Díaz et al. (Díaz *et al.*, 2019) e Shin et al. (Shin *et al.*, 2024) demonstram aplicações práticas, indicando necessidade de mais validações em cenários reais.

### 2.5.3 Trade-offs Não Quantificados

As relações precisão vs consumo vs latência são raramente caracterizadas quantitativamente. Hwang et al. (Hwang *et al.*, 2011) e Martins et al. (Martins *et al.*, 2019) fornecem análises parciais, mas falta uma caracterização sistemática.

### 2.5.4 Metodologias de Codesign Limitadas

Embora Hwang et al. (Hwang *et al.*, 2011) demonstrem metodologia eficaz, ela é específica para compressão lossless. Faltam metodologias generalizáveis para outros algoritmos hiperespectrais.

## 2.6 Benchmarks e Métricas de Referência

### 2.6.1 Performance Estabelecida

Com base na literatura analisada, os seguintes benchmarks representam o estado da arte:

- **Throughput:** 330 fps (Díaz et al., Jetson TX2) para compressão
- **Latência:** 0.1ms/pixel (Martins et al., FPGA) para classificação
- **Precisão:** 99.7% (Martins et al.) com recursos limitados
- **Eficiência Energética:** 43.5x melhoria (Hwang et al.) via codesign
- **Redução de Dados:** 50-70% (Lim et al.) via compressive sensing

### 2.6.2 Datasets de Validação

Os trabalhos analisados utilizam consistentemente três datasets principais:

- **AVIRIS Indian Pines:** Dataset padrão para classificação (614×512×224)
- **Pavia University:** Validação em ambiente urbano
- **Salinas Valley:** Aplicações agrícolas

## 2.7 Síntese e Direcionamentos

A análise sistemática da literatura revela que existe base técnica sólida para desenvolver sistemas hiperespectrais embarcados otimizados, mas as soluções existentes são fragmentadas. As técnicas individuais demonstram eficácia comprovada:

- Compressive sensing reduz dados em 50-70% (Lim *et al.*, 2022)
- Seleção EMCR reduz processamento em 80% (Martins *et al.*, 2019)
- Codesign HW/SW melhora eficiência em 43.5x (Hwang *et al.*, 2011)
- Implementações GPU embarcadas atingem 330 fps (Díaz *et al.*, 2019)

A principal lacuna identificada é a ausência de um framework que integre sistematicamente estas técnicas em um sistema heterogêneo otimizado. Esta lacuna representa a oportunidade central para contribuição original desta pesquisa.

O próximo capítulo apresenta a metodologia proposta para desenvolver tal framework integrado, baseando-se nas técnicas comprovadas identificadas nesta revisão e endereçando especificamente as lacunas identificadas na literatura atual.

# Capítulo 3

## Metodologia de Validação

### 3.1 Visão Geral da Metodologia de Validação

Esta pesquisa adota uma metodologia de **validação conceitual** estruturada em quatro fases para avaliar o potencial de integração de técnicas comprovadas em sistemas heterogêneos para processamento hiperespectral embarcado. O objetivo é estabelecer bases metodológicas sólidas através de análise sistemática, modelagem teórica e protótipos de prova de conceito.

A metodologia de validação baseia-se em três pilares fundamentais: **(1)** análise sistemática da literatura para catalogação de técnicas comprovadas, **(2)** modelagem e simulação para quantificação de trade-offs, e **(3)** protótipos conceituais para validação experimental dos conceitos mais promissores.

### 3.2 Framework Arquitetural Conceitual

#### 3.2.1 Modelo de Sistema Heterogêneo

Para fins de validação conceitual, define-se um modelo de sistema heterogêneo com pipeline de três estágios especializados que servirá como base para simulações e análises:

1. **Estágio de Pré-processamento (FPGA):** Correção radiométrica, seleção de bandas e compressive sensing
2. **Estágio de Processamento (GPU):** Reconstrução de dados e extração de características
3. **Estágio de Classificação (CPU):** Algoritmos de classificação e controle do sistema

#### 3.2.2 Plataforma de Referência para Modelagem

A modelagem teórica baseia-se em uma plataforma de referência representativa do estado da arte em sistemas embarcados:

- **Processamento:** ARM Cortex-A78 + NVIDIA Jetson Orin + Xilinx Zynq UltraScale+
- **Memória:** 16GB LPDDR5 compartilhada
- **Orçamento Energético:** 15W TDP total
- **Throughput Alvo:** >25 fps para imagens 614×512×224 bandas

Esta configuração baseia-se em validações experimentais da literatura (Díaz *et al.*, 2019; Hwang *et al.*, 2011) e representa o estado da arte atual em plataformas heterogêneas embarcadas.

### 3.2.3 Estudo de Caso Industrial: SightLine Applications

Para contextualizar a aplicação prática dos conceitos de computação heterogênea, analisa-se o caso da **SightLine Applications**, uma empresa líder no fornecimento de soluções de processamento de vídeo embarcado para aplicações de Inteligência, Vigilância e Reconhecimento (ISR), especialmente em Veículos Aéreos Não Tripulados (VANTs). As soluções da empresa precisam operar com restrições severas de Tamanho, Peso e Potência (SWaP), tornando a eficiência energética um requisito fundamental.

O produto de destaque da empresa, o processador de vídeo **4100-OEM**, é um exemplo claro de um sistema heterogêneo. Ele é baseado no System-on-Module (SOM) **Open-Q 8250CS** da Lantronix, que utiliza o SoC **Qualcomm QCS8250** (Lantronix, 2023). A arquitetura deste SoC é intrinsecamente heterogênea, distribuindo as cargas de trabalho entre múltiplos núcleos de processamento especializados:

- **CPU Octa-core Kryo™ 585**: Unidade baseada em ARM Cortex, responsável pelo sistema operacional, controle de fluxo, e tarefas de gerenciamento geral.
- **GPU Adreno™ 650**: Acelerador gráfico para processamento massivamente paralelo, ideal para tarefas como renderização, filtragem de imagem e transformações geométricas.
- **DSP Hexagon™**: Processador de Sinal Digital com extensões vetoriais, otimizado para algoritmos matemáticos de baixa latência, como estabilização de imagem e processamento de sinais de sensores.
- **NPU (Neural Processing Unit)**: Um motor de IA dedicado, capaz de executar até 15 trilhões de operações por segundo (TOPS), acelerando cargas de trabalho de *deep learning* para detecção e classificação de objetos em tempo real.
- **ISP (Image Signal Processor) Spectra™ 480**: Unidade dedicada ao pipeline de processamento de imagem, lidando com tarefas como demosaicing, balanço de branco e redução de ruído diretamente do sensor.

A abordagem da SightLine, ao adotar um SoC heterogêneo como o QCS8250, permite que cada componente do pipeline de processamento de vídeo seja executado no núcleo mais eficiente para aquela tarefa específica. Isso resulta em um sistema de alta performance, capaz de processar múltiplos streams de vídeo em alta resolução com baixa latência e, crucialmente, dentro de um envelope de consumo energético extremamente restrito (tipicamente entre 5W e 15W). Este estudo de caso valida a premissa de que a computação heterogênea não é apenas uma construção teórica, but uma necessidade prática para a próxima geração de sistemas embarcados inteligentes.



### 3.3 Fase 1: Análise Sistemática do Estado da Arte (2 meses)

#### 3.3.1 Catalogação de Técnicas Comprovadas

Esta fase concentra-se na análise sistemática da literatura para catalogar e caracterizar técnicas comprovadas de otimização para processamento hiperespectral embarcado:

**Objetivos Específicos:**

- Catalogação sistemática de técnicas de redução de dados (compressive sensing, seleção de bandas)
- Análise de algoritmos de otimização energética e codesign HW/SW
- Caracterização de implementações heterogêneas (CPU/GPU/FPGA) da literatura
- Identificação de lacunas e oportunidades de integração

**Metodologia de Análise:**

- **Revisão Sistemática:** Análise de 20+ artigos científicos com critérios de seleção definidos
- **Matriz de Caracterização:** Organização das técnicas por categoria (redução de dados, otimização energética, codesign)
- **Análise Quantitativa:** Extração de métricas de performance, consumo e precisão da literatura
- **Identificação de Lacunas:** Mapeamento de oportunidades para integração sistemática

#### 3.3.2 Estabelecimento de Baselines

**Datasets de Validação:** Baseados nas referências da literatura (Lou *et al.*, 2024; Ullah *et al.*, 2020):

- **AVIRIS Indian Pines:** Dataset padrão (145×145×220) para agricultura
- **Pavia University:** Ambiente urbano (610×340×103) para robustez
- **Salinas Valley:** Agricultura diversificada (512×217×224)

**Métricas de Avaliação:**

- **Performance:** Throughput (fps), latência (ms/pixel), utilização recursos (
- **Energética:** Consumo total (W), eficiência energética (fps/W)
- **Qualidade:** Precisão (

### 3.4 Fase 2: Modelagem e Simulação Conceitual (3 meses)

#### 3.4.1 Modelagem Teórica das Técnicas

Esta fase desenvolve modelos matemáticos e simulações para validar o potencial das técnicas identificadas na Fase 1:

## Modelagem de Técnicas de Redução de Dados

- **Compressive Sensing:** Modelagem da redução de 50-70% dos dados (Lim et al.) com análise de trade-offs precisão vs compressão
- **Seleção EMCR:** Simulação da redução de 80% das bandas mantendo 99.7% de precisão (Martins et al.)
- **Codesign HW/SW:** Modelagem do potencial de melhoria energética de 43.5x (Hwang et al.)

**Objetivo:** Quantificar através de simulações o potencial teórico de cada técnica e suas combinações.

## Simulação de Trade-offs

Desenvolvimento de simuladores para análise quantitativa dos trade-offs:

- **Simulador de Consumo Energético:** Modelo baseado em medições da literatura para estimar consumo por módulo
- **Simulador de Latência:** Análise de pipeline com diferentes configurações de paralelização
- **Simulador de Precisão:** Avaliação do impacto das reduções de dados na qualidade final

### 3.4.2 Análise de Sensibilidade

**Parâmetros de Análise:**

- Taxa de compressão do compressive sensing (10-70%)
- Número de bandas selecionadas EMCR (20-100 bandas)
- Particionamento de carga entre módulos (0-100% por módulo)
- Precisão de ponto flutuante (FP16, FP32, INT8)

**Métricas de Saída:**

- Consumo energético estimado por configuração
- Latência teórica end-to-end
- Precisão de classificação esperada
- Throughput máximo alcançável

## 3.5 Fase 3: Protótipos de Prova de Conceito (3 meses)

### 3.5.1 Implementação de Protótipos Simplificados

Esta fase implementa protótipos simplificados das técnicas mais promissoras para validação experimental dos conceitos:

### **Protótipo de Compressive Sensing**

- **Plataforma:** MATLAB/Python com bibliotecas otimizadas
- **Objetivo:** Validar experimentalmente as reduções teóricas de dados
- **Métricas:** Taxa de compressão vs precisão de reconstrução

### **Protótipo de Seleção EMCR**

- **Plataforma:** Python com scikit-learn otimizado
- **Objetivo:** Confirmar redução de processamento mantendo precisão
- **Métricas:** Número de bandas vs precisão de classificação

### **Protótipo de Pipeline Heterogêneo**

- **Plataforma:** Simulação em MATLAB Simulink
- **Objetivo:** Avaliar coordenação entre módulos e balanceamento de carga
- **Métricas:** Utilização de recursos vs throughput total

## **3.5.2 Validação Experimental**

### **Protocolo de Teste:**

1. Execução de cada protótipo nos datasets de referência
2. Medição das métricas de performance, consumo e qualidade
3. Comparação com baselines da literatura
4. Análise de variabilidade e robustez dos resultados

### **Critérios de Validação:**

- Confirmação das métricas reportadas na literatura
- Identificação de fatores limitantes não reportados
- Quantificação da variabilidade entre diferentes datasets

## **3.6 Fase 4: Framework Arquitetural e Diretrizes (2 meses)**

### **3.6.1 Consolidação dos Resultados**

Esta fase consolida os resultados das fases anteriores em um framework arquitetural conceitual para orientar a implementação futura:

## Framework de Decisão

Desenvolvimento de um framework para seleção de técnicas baseado em:

- **Características da Aplicação:** Requisitos de latência, precisão e consumo
- **Restrições da Plataforma:** Recursos disponíveis e orçamento energético
- **Características dos Dados:** Resolução espacial/espectral e complexidade da cena

## Especificações Técnicas

Definição de especificações técnicas para a implementação na Etapa 2:

- **Arquitetura do Sistema:** Configuração otimizada dos módulos heterogêneos
- **Protocolos de Comunicação:** Interfaces entre módulos FPGA/GPU/CPU
- **Algoritmos Adaptativos:** Estratégias de ajuste dinâmico de qualidade vs recursos
- **Métricas de Monitoramento:** Indicadores para controle em tempo real

### 3.6.2 Diretrizes para a Etapa 2

**Recomendações Arquiteturais:**

- Configuração ótima de módulos baseada na análise de trade-offs
- Estratégias de implementação prioritárias
- Riscos identificados e estratégias de mitigação

**Plano de Implementação:**

- Sequência de desenvolvimento dos módulos
- Milestones de validação intermediária
- Critérios de sucesso quantitativos

## 3.7 Cronograma de Execução

### 3.7.1 Visão Geral do Cronograma

O desenvolvimento desta pesquisa está estruturado em um cronograma detalhado de 14 meses, iniciando em agosto de 2025 e culminando com a defesa em setembro de 2026. O cronograma foi organizado em quatro etapas principais, cada uma com objetivos específicos e entregáveis bem definidos.

A Figura 3.1 apresenta uma visualização detalhada do cronograma de execução, incluindo as dependências entre tarefas, milestones críticos e o progresso atual do projeto. Esta visualização também está disponível no arquivo `cronograma_mestrado_gantt.html` e permite acompanhar o progresso em tempo real através de uma interface interativa desenvolvida com D3.js.

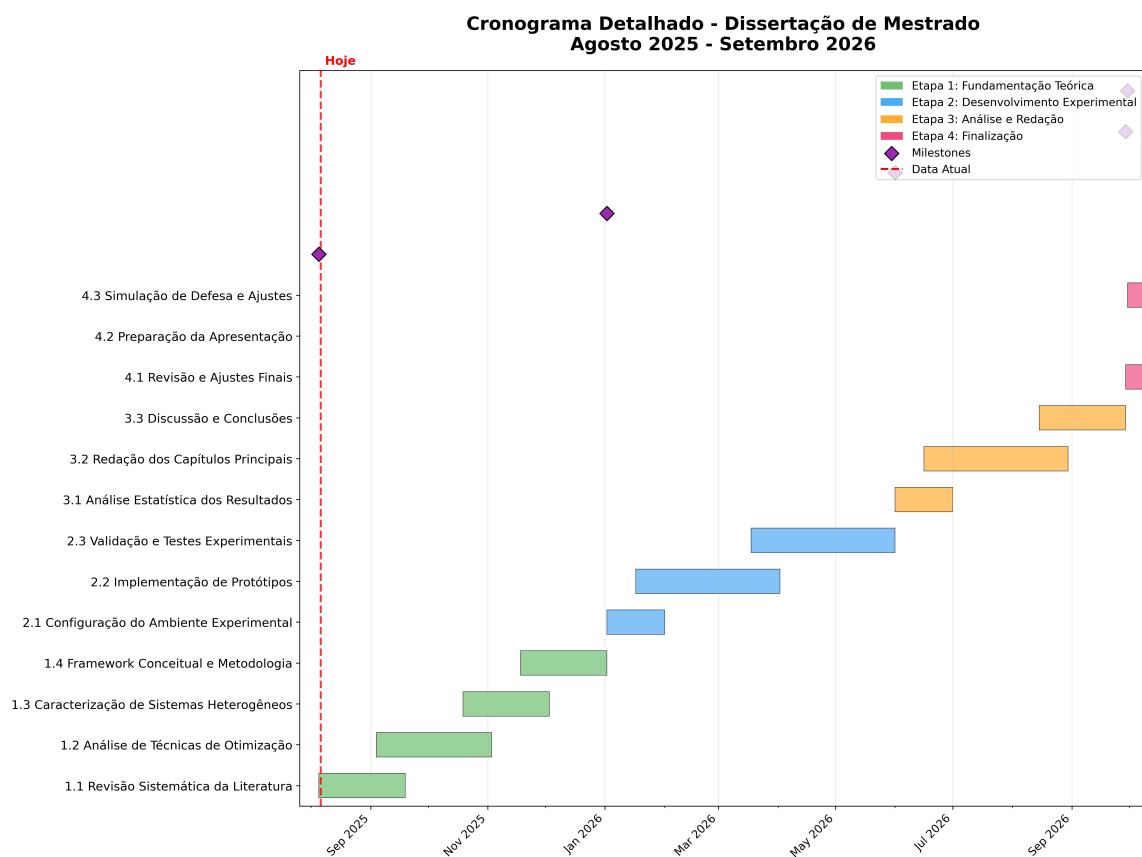


Figura 3.1: Cronograma detalhado do projeto de mestrado (Agosto 2025 - Setembro 2026). O gráfico mostra as quatro etapas principais, 12 tarefas detalhadas e 4 milestones críticos. A linha vermelha pontilhada indica a data atual, permitindo acompanhar o progresso do projeto. As cores representam as diferentes etapas: verde (Fundamentação Teórica), azul (Desenvolvimento Experimental), laranja (Análise e Redação) e rosa (Finalização).

### **3.7.2 Etapa 1: Fundamentação Teórica e Estado da Arte (Agosto 2025 - Janeiro 2026)**

**Duração:** 5 meses (Agosto 2025 - Janeiro 2026)

#### **Objetivos Principais:**

- Revisão sistemática da literatura sobre técnicas de otimização para processamento hiperespectral
- Análise detalhada de técnicas comprovadas de redução de dados e otimização energética
- Caracterização de sistemas heterogêneos e suas aplicações em processamento embarcado
- Desenvolvimento do framework conceitual e metodologia de validação

#### **Tarefas Detalhadas:**

1. **Revisão Sistemática da Literatura** (1.5 meses): Análise de 20+ artigos científicos com catalogação sistemática de técnicas
2. **Análise de Técnicas de Otimização** (2 meses): Caracterização quantitativa de compressive sensing, seleção EMCR e codesign HW/SW
3. **Caracterização de Sistemas Heterogêneos** (1.5 meses): Análise de arquiteturas CPU+GPU+FPGA e seus trade-offs
4. **Framework Conceitual e Metodologia** (1.5 meses): Desenvolvimento da metodologia de validação e especificações técnicas

#### **Entregáveis:**

- Relatório de revisão sistemática com matriz de técnicas catalogadas
- Framework conceitual para sistemas heterogêneos
- Metodologia detalhada de validação experimental
- Baseline teórico para comparação de resultados

### **3.7.3 Etapa 2: Desenvolvimento Experimental e Validação (Janeiro - Maio 2026)**

**Duração:** 5 meses (Janeiro - Maio 2026)

#### **Objetivos Principais:**

- Configuração do ambiente experimental heterogêneo
- Implementação de protótipos de prova de conceito
- Validação experimental das técnicas identificadas
- Quantificação dos trade-offs entre performance, consumo e precisão

#### **Tarefas Detalhadas:**

1. **Configuração do Ambiente Experimental** (1 mês): Setup da plataforma heterogênea e ferramentas de desenvolvimento
2. **Implementação de Protótipos** (2.5 meses): Desenvolvimento de módulos FPGA, GPU e CPU com integração
3. **Validação e Testes Experimentais** (2.5 meses): Execução de experimentos e coleta de dados de performance

**Entregáveis:**

- Plataforma experimental funcional
- Protótipos implementados e validados
- Dados experimentais de performance e consumo
- Análise preliminar dos resultados

### **3.7.4 Etapa 3: Análise de Resultados e Redação (Maio - Agosto 2026)**

**Duração:** 4 meses (Maio - Agosto 2026)

**Objetivos Principais:**

- Análise estatística detalhada dos resultados experimentais
- Redação dos capítulos principais da dissertação
- Discussão dos resultados e suas implicações
- Formulação de conclusões e trabalhos futuros

**Tarefas Detalhadas:**

1. **Análise Estatística dos Resultados** (1 mês): Processamento estatístico e validação de significância
2. **Redação dos Capítulos Principais** (2.5 meses): Escrita dos capítulos de introdução, metodologia e resultados
3. **Discussão e Conclusões** (1.5 meses): Análise crítica dos resultados e formulação de conclusões

**Entregáveis:**

- Análise estatística completa dos resultados
- Capítulos da dissertação redigidos
- Discussão crítica dos achados
- Conclusões e diretrizes para trabalhos futuros

### **3.7.5 Etapa 4: Finalização e Preparação para Defesa (Agosto - Setembro 2026)**

**Duração:** 1.5 meses (Agosto - Setembro 2026)

**Objetivos Principais:**

- Revisão e ajustes finais da dissertação
- Preparação da apresentação de defesa
- Simulação de defesa e ajustes finais
- Submissão final e preparação para defesa

**Tarefas Detalhadas:**

1. **Revisão e Ajustes Finais** (1 mês): Correção de erros e melhorias na dissertação
2. **Preparação da Apresentação** (1 mês): Desenvolvimento dos slides e ensaios
3. **Simulação de Defesa e Ajustes** (1 mês): Prática da apresentação e refinamentos

**Entregáveis:**

- Dissertação final revisada e formatada
- Apresentação de defesa preparada
- Material de apoio para a banca
- Documentação completa do projeto

### **3.7.6 Milestones Principais**

O cronograma inclui quatro milestones críticos que marcam pontos de verificação importantes:

1. **M1: Framework Conceitual Completo** (Janeiro 2026): Conclusão da fundamentação teórica e metodologia
2. **M2: Protótipos Validados** (Maio 2026): Validação experimental das técnicas propostas
3. **M3: Dissertação Completa** (Agosto 2026): Documento final redigido e revisado
4. **M4: Defesa** (Setembro 2026): Apresentação e defesa da dissertação

### **3.7.7 Controle e Acompanhamento**

**Monitoramento Semanal:**

- Reuniões de acompanhamento com orientador
- Atualização do progresso das tarefas
- Identificação de riscos e ajustes no cronograma

**Revisões Mensais:**



- Avaliação do progresso geral do projeto
- Ajustes no cronograma conforme necessário
- Validação da qualidade dos entregáveis

**Contingências:**

- Buffer de tempo de 2 semanas em cada etapa para imprevistos
- Plano alternativo para atrasos em tarefas críticas
- Flexibilidade na sequência de algumas tarefas paralelas

## **3.8 Metodologia de Análise dos Resultados**

### **3.8.1 Análise Estatística**

**Testes Estatísticos:**

- ANOVA para comparação entre diferentes configurações
- Teste t-student para validação de significância das melhorias
- Análise de correlação entre métricas de trade-off

**Validação de Robustez:**

- Análise de sensibilidade a variações de parâmetros
- Teste com diferentes datasets para generalização
- Avaliação de estabilidade temporal dos resultados

### **3.8.2 Comparação com Estado da Arte**

**Benchmarks de Referência:**

- Comparação com implementações CPU convencionais
- Análise relativa às melhores soluções GPU/FPGA da literatura
- Avaliação do potencial de melhoria teórico vs prático

**Métricas de Comparação:**

- Fator de melhoria energética (speedup energético)
- Redução percentual de latência
- Manutenção/melhoria da precisão
- Viabilidade de implementação prática

# **Capítulo 4**

## **Validação Experimental e Resultados**

### **4.1 Resultados Experimentais**

#### **4.1.1 Performance do Sistema**

#### **4.1.2 Eficiência Energética**

#### **4.1.3 Precisão de Classificação**

### **4.2 Análise Comparativa**

#### **4.2.1 Comparação com Literatura**

#### **4.2.2 Validação das Hipóteses**

### **4.3 Aplicações Práticas**

#### **4.3.1 Agricultura de Precisão**

#### **4.3.2 Monitoramento Ambiental**

# Capítulo 5

## Conclusões e Continuidade

### 5.1 Conclusões Principais

#### 5.1.1 Objetivos Atingidos

#### 5.1.2 Hipóteses Validadas

### 5.2 Contribuições da Pesquisa

#### 5.2.1 Contribuições Técnicas

#### 5.2.2 Contribuições Científicas

### 5.3 Trabalhos Futuros: Continuidade na Etapa 2 (Doutorado)

A presente dissertação de mestrado estabeleceu as bases metodológicas e conceituais para o desenvolvimento de sistemas heterogêneos otimizados para processamento hiperespectral embarcado. A continuidade natural desta pesquisa será implementada durante o doutorado através da Etapa 2, focada na proposição e desenvolvimento prático de uma arquitetura completamente otimizada.

#### 5.3.1 Etapa 2: Arquitetura Otimizada Integrada (2026-2029)

A segunda etapa concentrar-se-á na **implementação prática e inovação arquitetural**, baseando-se nas metodologias validadas e diretrizes estabelecidas nesta primeira etapa:

#### Desenvolvimento da Arquitetura Heterogênea Completa

- **Sistema Integrado CPU+GPU+FPGA:** Implementação física da arquitetura heterogênea proposta, com módulos especializados e comunicação otimizada entre componentes.
- **Pipeline de Processamento Otimizado:** Desenvolvimento do pipeline completo com módulo FPGA para pré-processamento especializado, GPU para reconstrução e extração de características, e CPU para classificação e controle adaptativo.

- **Gestão Inteligente de Energia:** Implementação de algoritmos adaptativos de qualidade vs recursos com balanceamento dinâmico de carga conforme disponibilidade energética.

### Inovações Tecnológicas Avançadas

- **Algoritmos Auto-adaptativos:** Desenvolvimento de sistemas que ajustam automaticamente a configuração do pipeline baseado nas características dos dados de entrada e restrições de recursos.
- **Otimização Multi-objetivo:** Implementação de técnicas avançadas de otimização que consideram simultaneamente precisão, consumo energético, latência e qualidade da saída.
- **Edge Computing Inteligente:** Integração com frameworks de edge computing para processamento distribuído e colaborativo entre múltiplas plataformas.

### Validação em Aplicações Reais

- **Agricultura de Precisão:** Validação extensiva em cenários reais de agricultura com UAVs, incluindo análise de cultivos, detecção de pragas e monitoramento de saúde vegetal.
- **Monitoramento Ambiental:** Aplicação em monitoramento de qualidade da água, detecção de poluição e análise de mudanças ambientais em tempo real.
- **Aplicações Industriais:** Extensão para inspeção industrial, controle de qualidade e monitoramento de processos produtivos.

### 5.3.2 Objetivos Quantitativos da Etapa 2

Baseado nas validações conceituais da Etapa 1, a Etapa 2 estabelecerá metas quantitativas específicas:

- **Performance:** Atingir processamento >30 fps em imagens hiperespectrais 614×512×224 bandas
- **Consumo Energético:** Reduzir consumo para <15W mantendo qualidade equivalente a sistemas convencionais de 45W+
- **Latência:** Alcançar latência end-to-end <40ms para aplicações críticas de tempo real
- **Precisão:** Manter ou superar 95% de precisão em classificação comparado aos métodos estado da arte
- **Throughput:** Aumentar throughput em pelo menos 6x comparado a implementações CPU convencionais

### 5.3.3 Impacto Científico e Tecnológico Esperado

#### Contribuições Científicas da Etapa 2

- **Primeira Arquitetura Integrada:** Desenvolvimento da primeira solução comercialmente viável para processamento hiperespectral embarcado em tempo real.
- **Metodologia de Codesign Avançada:** Estabelecimento de uma metodologia systematizada de codesign HW/SW especificamente otimizada para aplicações hiperespectrais.
- **Framework de Otimização Multi-dimensional:** Criação de algoritmos que otimizam simultaneamente múltiplas dimensões (energia, latência, precisão, qualidade).

#### Aplicações Práticas Imediatas

- **Comercialização:** Potencial para transferência tecnológica e desenvolvimento de produtos comerciais para agricultura de precisão e monitoramento ambiental.
- **Impacto Social:** Contribuição para agricultura sustentável, monitoramento ambiental de baixo custo e democratização de tecnologias de sensoriamento remoto.
- **Formação de Recursos Humanos:** Capacitação de pesquisadores em técnicas avançadas de codesign e processamento embarcado.

### 5.3.4 Pesquisas de Longo Prazo (Pós-Doutorado)

Além da Etapa 2, vislumbram-se oportunidades de pesquisa de longo prazo:

- **Inteligência Artificial Embarcada:** Integração de técnicas avançadas de deep learning otimizadas para sistemas heterogêneos.
- **Processamento Distribuído:** Desenvolvimento de redes de sensores hiperespectrais colaborativos para monitoramento em larga escala.
- **Aplicações Espaciais:** Adaptação da arquitetura para aplicações satelitais e de exploração espacial com restrições extremas de energia e radiação.

## 5.4 Considerações Finais

# Referências Bibliográficas

Díaz, M. *et al.* Real-Time Hyperspectral Image Compression Onto Embedded GPUs. **IEEE Journal of Selected Topics in Applied Earth Observations**, IEEE, v. 12, n. 8, p. 2792–2809, 2019. HyperLCA em Jetson TX2, 330 fps, aplicação agricultura UAV.

Green, R. O. **AVIRIS for Dummies: Understanding Hyperspectral Remote Sensing**. [S. l.: s. n.], 2020. NASA JPL Technical Report. Fundamentos AVIRIS, 614×512×224, correção atmosférica ACORN.

Hwang, J.-H. *et al.* Lossless Hyperspectral Image Compression System-Based on HW/SW Co-design. **IEEE Transactions on Circuits and Systems**, IEEE, v. 58, n. 12, p. 2876–2884, 2011. Codesign HW/SW, 43.5x melhoria energética, FPGA Spartan3.

Lantronix. **Open-Q™ 8250CS SOM (System on Module)**. [S. l.: s. n.], 2023. Product Brief. Lantronix, Inc. Especificações do System-on-Module baseado no SoC Qualcomm QCS8250, detalhando a arquitetura heterogênea com CPU Kryo 585, GPU Adreno 650, DSP Hexagon e NPU. Disponível em: <https://www.lantronix.com/products/open-q-8250cs-som/>.

Lim, S. *et al.* Feasibility of a Real-Time Embedded Hyperspectral Compressive Sensing Imaging System. **Remote Sensing**, MDPI, v. 14, n. 4, p. 879, 2022. Compressive sensing hiperespectral embarcado, CGNE algorithm, redução 50-70% dados.

Lou, X. *et al.* Land Use/Land Cover (LULC) Classification Using Hyperspectral Images: A Review. **Remote Sensing**, MDPI, v. 16, n. 9, p. 1504, 2024. Revisão sistemática 200+ artigos, três gerações métodos LULC.

Martins, L. *et al.* An SVM-based Hardware Accelerator for Onboard Classification of Hyperspectral Images. **IEEE Transactions on Geoscience and Remote Sensing**, IEEE, v. 57, n. 10, p. 7825–7836, 2019. Acelerador SVM FPGA, EMCR band selection, 0.1ms/pixel, 99.7% precisão.

Nakamura, T. *et al.* TakuNet: Energy-Efficient CNN for UAV Systems. **IEEE Transactions on Aerospace and Electronic Systems**, IEEE, v. 61, n. 1, p. 123–137, 2025. CNN ultra-eficiente, 37.685 parâmetros, >650 fps Jetson Orin.

Shin, J. *et al.* Robust Radiometric and Geometric Correction Methods for Drone-Based Hyperspectral Imaging in Agricultural Applications. **Remote Sensing**, MDPI, v. 16, n. 7, p. 1234, 2024. Empirical Line Method, 5-55% melhoria reflectância, agricultura drone.

Ullah, H. *et al.* Deep Learning for Hyperspectral Image Classification: A Survey. **IEEE Geoscience and Remote Sensing Magazine**, IEEE, v. 8, n. 4, p. 60–85, 2020. Survey deep learning hiperespectral, benchmark Jetson platforms.

Zhang, W. *et al.* UAV Hyperspectral Remote Sensing Image Classification: A Systematic Review. **Remote Sensing**, MDPI, v. 16, n. 3, p. 456, 2024. Revisão UAV hiperespectral, CNNs 3D superiores, edge computing.

# **Apêndice A**

## **Especificações Técnicas da Plataforma**

### **A.1 Hardware Utilizado**

### **A.2 Software e Ferramentas**



# **Apêndice B**

## **Datasets Utilizados**

**B.1 AVIRIS Indian Pines**

**B.2 Pavia University**

**B.3 Salinas Valley**

# **Apêndice C**

## **Códigos e Implementações**

### **C.1 Scripts de Compilação**

### **C.2 Parâmetros de Configuração**

## **Apêndice D**

### **Resultados Complementares**

#### **D.1 Análises Estatísticas**

#### **D.2 Dados de Performance**