Modelos de Linguagem-Visão como Pré-Anotadores em Imagens Aéreas: Avaliação de Localização Semântica em Cenários Zero-Shot e Few-Shot

Daniel Martins Monteiro da Silva[1,2], Vinicius de Lemos Cardoso Rondon[1], Renan Vieira de Cerqueira[2], Lamartine Nogueira Frutuoso Guimaraes[2], Dimas Irion Alves[1,2] e Sarah Negreiros de Carvalho[1,2]

[1]Centro de Competência em Guerra Eletrônica – Instituto Tecnológico de Aeronáutica (ITA) – São José dos Campos/SP – Brasil

[2] Instituto Tecnológico de Aeronáutica (ITA) – São José dos Campos/SP – Brasil

***Resumo*— A anotação manual de imagens aéreas é custosa e limita o desenvolvimento de sistemas supervisionados em domínios de dados escassos. Este trabalho avalia dois Modelos de Linguagem-Visão (VLMs), GPT-5.5 e Gemini 3.1 Pro, como pré-anotadores automáticos de barracas, construções e barcos em 28 imagens aéreas da Amazônia, mantidas independentes de duas imagens adicionais usadas exclusivamente como suporte *few-shot*. Sob estratégias de *prompting zero-shot* e *few-shot*, os modelos retornam caixas delimitadoras. A análise empregou três métricas complementares: F1 com IoU $\geq 0,5$ (padrão), IoU $\geq 0,25$ (relaxado) e *Center Hit* (verifica se o centro da predição pertence à caixa de referência), permitindo distinguir erros de reconhecimento semântico de erros de delimitação espacial. O *few-shot* mostrou ganhos de até 0,31 no F1. O Gemini 3.1 Pro alcançou 78,1% de F1 por *Center Hit* contra 54,9% por IoU $\geq$ 0,5, indicando consciência espacial sem precisão de contorno. Os resultados apoiam o uso de VLMs como ferramentas de pré-anotação com posterior revisão humana.**

Palavras-Chave*— Modelos de Linguagem-Visão, detecção de objetos, imagens aéreas, anotação assistida, *few-shot*, *Center Hit

I. Introdução

A vigilância aérea de regiões da Amazônia tem crescido em importância operacional com a expansão de missões de reconhecimento que produzem grandes volumes de imagens [1], [2]. Transformar esse material em dados de treinamento para sistemas automáticos de detecção de objetos exige anotação manual especializada, processo lento, caro e difícil de escalar. Detectores modernos da família YOLO [3], [4] atingem desempenho de ponta quando há dados suficientes, mas sua eficácia depende diretamente da quantidade e qualidade das anotações disponíveis. Esse gargalo é particularmente crítico em domínios aeroespaciais e de defesa, onde a produção de dados rotulados é limitada.

Em trabalho anterior, os autores abordaram o problema de classificação binária de imagens, distinguindo regiões de garimpo de regiões sem atividade, e obtiveram aproximadamente 98% de acurácia em imagens aéreas com redes neurais convolucionais [5]. A etapa seguinte natural é a detecção de objetos individuais de interesse, como barracas, construções e embarcações. Para isso, porém, é necessário produzir anotações espaciais detalhadas, o que representa custo elevado e limita a construção de conjuntos de treinamento em larga escala.

D.M.M. Silva, monteirodmms@ita.br; V.L.C. Rondon; R.V. Cerqueira; L.N.F. Guimaraes; D. I. Alves; S.N.C. Leite — Centro de Competência em Guerra Eletrônica – Divisão de Engenharia Eletrônica, ITA, São José dos Campos/SP, Brasil.

Os Modelos de Linguagem-Visão (VLMs) surgem como alternativa promissora para contornar esse problema [6]. Diferentemente dos detectores supervisionados, os VLMs são treinados sobre grandes volumes de pares imagem-texto e desenvolvem representações visuais que se generalizam entre domínios sem necessidade de retreinamento. Sua capacidade de operar sem exemplos anotados já foi documentada em diversas tarefas de visão computacional [7], [8], mas a aplicação a imagens aéreas em perspectivas oblíquas ou zenitais ainda é pouco explorada na literatura [9].

As contribuições deste trabalho são:

1) Avaliação dos modelos GPT-5.5 (OpenAI) e Gemini 3.1 Pro (Google) para detecção de três classes de interesse operacional em imagens aéreas amazônicas, preenchendo lacuna em domínio crítico para aplicações de defesa.
2) Estudo da influência dos modelos VLMs sob condições *zero-shot* e *few-shot* em imagens aéreas amazônicas.
3) Adaptação da métrica *Center Hit* para dissociar localização semântica de precisão geométrica. Embora métricas semelhantes sejam utilizadas em visual grounding [10], [11], seu emprego para analisar o comportamento de VLMs como pré-anotadores de imagens aéreas permanece inexplorado na literatura corrente.
4) Análise quantitativa da capacidade de VLMs localizarem corretamente muito mais objetos do que sugerem métricas baseadas apenas em IoU.

O restante do artigo está organizado como se segue: a Seção II apresenta a fundamentação teórica; a Seção III descreve a metodologia; a Seção IV apresenta e discute os resultados; a Seção V conclui com direções futuras.

II. Fundamentação Teórica

Modelos como GPT-5.5 (OpenAI) [12] e Gemini 3.1 Pro (Google) [13] são treinados sobre grandes volumes de imagens associadas a descrições textuais, o que lhes permite desenvolver representações visuais abrangentes que se transferem entre domínios sem necessidade de retreinamento. Ao contrário de detectores supervisionados, que aprendem um mapeamento fixo de características visuais para categorias predefinidas, esses modelos raciocinam sobre imagens a partir de instruções em linguagem natural, o que possibilita redirecioná-los para novas tarefas com relativa facilidade [6].

Embora VLMs ainda apresentem desempenho inferior ao de detectores supervisionados em domínios com dados abundantes [14], trabalhos recentes mostram que estratégias de

prompting mais elaboradas, como cadeias de raciocínio visual com sobreposição de ferramentas de medição na imagem, elevam substancialmente seu desempenho em detecção *zero-shot* [8]. A inclusão de exemplos anotados no contexto (*few-shot prompting*) tende a melhorar os resultados de forma consistente em relação ao modo *zero-shot*, tanto em tarefas de linguagem quanto de visão [15]. Uma limitação recorrente, porém, é a imprecisão nas coordenadas das *bounding boxes* pois esses modelos foram concebidos para raciocínio semântico global e não para a estimação precisa de posições [1].

Imagens capturadas em perspectiva zenital ou oblíqua apresentam desafios distintos dos abordados nos *benchmarks* típicos de visão computacional. Os objetos aparecem em escala reduzida, com variação de iluminação, sombras projetadas e fundos complexos que frequentemente imitam a aparência dos próprios alvos. A vegetação densa pode ser confundida com barracas de lona, e superfícies d'água com cascos de embarcações [1]. Essa divergência entre a aparência semântica esperada e a aparência visual efetiva dificulta tanto detectores tradicionais quanto VLMs. Abordagens baseadas em aprendizado profundo têm demonstrado viabilidade para detecção de objetos em imagens aéreas [1], [16], mas dependem de conjuntos de dados anotados de escala considerável, cuja produção é custosa e escassa nesse domínio [17]. Nesse cenário, abordagens que reduzam a dependência de grandes volumes de anotação tornam-se especialmente relevantes.

A avaliação de sistemas de detecção de objetos convencionalmente adota o Índice de Intersecção sobre União (IoU) como critério de acerto, com o limiar IoU $\geq$ 0,5 estabelecido como padrão pelo *benchmark* PASCAL VOC [18]. Esse critério, porém, é sensível à convenção de anotação: quando o anotador desenha caixas folgadas e o modelo as desenha apertadas, ou o contrário, detecções semanticamente corretas acabam penalizadas mesmo que o objeto tenha sido efetivamente localizado.

No campo de *visual grounding*, que avalia a capacidade de modelos de linguagem-visão de localizar regiões associadas a descrições textuais, é comum utilizar métricas baseadas na correta localização do objeto, sem exigir alinhamento preciso entre as fronteiras previstas e as de referência [10]. *Benchmarks* como RefCOCO e RefCOCO+ adotam variantes dessa estratégia para avaliar a localização semântica independentemente da precisão geométrica da região prevista [11]. Inspirados por essa abordagem, neste trabalho é empregada também a métrica *Center Hit*, que considera uma detecção correta quando o centro geométrico da caixa prevista está contido na caixa de referência correspondente.

O fluxo de trabalho de anotação com revisão humana (*Human-in-the-Loop*) propõe o uso de predições automáticas como ponto de partida para o anotador, que então corrige as caixas em vez de desenhá-las do zero [17]. Nesse contexto, a *Center Hit* adquire relevância particular: se o modelo indicou a região correta, o trabalho do anotador resume-se a ajustar bordas, tarefa estimada como substancialmente mais rápida do que a anotação completa. O IoU $\geq$ 0,5 clássico, nesse cenário, subestima o valor prático das predições ao descartar como erros caixas que já apontam para os objetos corretos com baixa precisão de contorno.

III. Metodologia

O conjunto de dados consiste em 30 imagens aéreas coloridas de 1920×1080 pixels adquiridas por sensores ópticos a bordo de aeronaves tripuladas em sobrevoos na Amazônia. As imagens foram anotadas manualmente por especialista no domínio com o *software* Roboflow, produzindo 353 *bounding boxes* no formato YOLO distribuídas em três classes: 94 barracas (estruturas improvisadas de lona típicas de acampamentos), 140 construções (edificações com telhado metálico visível) e 119 barcos (embarcações sobre ou próximas ao curso d'água).

Duas dessas imagens, `image4.jpg` e `image16.jpg`, foram reservadas como suporte *few-shot* (Seção III-C) e excluídas de toda avaliação, em ambas as condições, para evitar contaminação entre suporte e teste. As 28 imagens restantes, com 307 caixas (70 barracas, 128 construções, 109 barcos), compõem o conjunto avaliado em todas as tabelas, figuras e matrizes deste trabalho.

Foram avaliados o GPT-5.5 (OpenAI), acessado via API OpenAI, e o Gemini 3.1 Pro (Google), via API Google GenAI, ambos VLMs multimodais de grande escala com suporte nativo à análise de imagens. Os experimentos utilizaram os pesos padrão de cada modelo, sem ajuste fino.

Cada uma das 28 imagens avaliadas foi submetida aos modelos sob duas condições de *prompting*.

***Zero-shot*:** O *prompt* descreveu as três classes em linguagem natural, enfatizando características visuais observáveis em imagens aéreas: formato irregular de lona azul, verde ou preta para barracas, telhado retangular metálico ou de madeira para construções e casco alongado próximo à margem para barcos. O modelo foi instruído a retornar detecções no formato YOLO, com uma linha por objeto contendo o identificador de classe e as coordenadas normalizadas em $[0, 1]$, sem qualquer texto adicional.

***Few-shot*:** As mesmas descrições de classe foram mantidas, acrescidas de uma imagem de suporte por classe, com suas anotações YOLO correspondentes filtradas para a classe de interesse. Foram usadas duas imagens de suporte, nenhuma pertencente ao conjunto avaliado: `image4.jpg`, com 24 instâncias anotadas de barraca, para a classe barraca; e `image16.jpg`, com 12 instâncias de construção e 10 de barco, para as classes construção e barco. Cada imagem de suporte foi apresentada como um par imagem-anotação no histórico de conversa (turno do usuário e turno do assistente) antes da imagem-alvo a ser anotada.

Em ambos os casos, a saída textual foi filtrada por expressão regular (`[0-2] cx cy w h`, $[0, 1]$) para reter apenas linhas YOLO válidas, descartando o restante sem correção. Cada imagem recebeu uma única chamada de API por condição (sem repetições; *retries* apenas para *rate limit*), em resolução original (1920×1080, JPEG/base64); nenhum parâmetro de amostragem foi especificado nas chamadas, de modo que ambas as APIs operaram com temperatura padrão de 1,0. Detecções duplicadas não foram removidas: são tratadas individualmente pelo pareamento *greedy* a seguir, que admite no máximo um acerto por objeto de referência.

As predições foram avaliadas por precisão (P), revocação (R) e F1-score por classe, reportando também a média macro entre classes. O emparelhamento predição–referência é realizado de forma *greedy*, priorizando o par de maior

IoU dentro da mesma classe, e cada referência só pode ser emparelhada com uma predição.

Três critérios de acerto foram empregados:

1) **IoU $\geq$ 0,5:** critério padrão PASCAL VOC [18], exige sobreposição substancial entre caixas prevista e de referência.
2) **IoU $\geq$ 0,25:** critério relaxado que captura detecções com localização aproximada, relevantes no contexto de pré-anotação.
3) ***Center Hit*:** o centro da caixa prevista (c_x, c_y) deve estar contido dentro da caixa de referência da mesma classe; cada referência é marcada como emparelhada no primeiro acerto (*greedy*). Esse critério avalia localização semântica independentemente do tamanho da caixa [10], [11].

A Fig. 1 ilustra o cálculo do IoU. Dados dois retângulos A *ground truth* (GT) e B (predição), o IoU é definido como:

$$\mathrm{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{1}$$

onde $|A \cup B| = |A| + |B| - |A \cap B|$. Um limiar τ define o critério de acerto: se $\mathrm{IoU}(A, B) \geq \tau$, a predição é classificada como verdadeiro positivo.

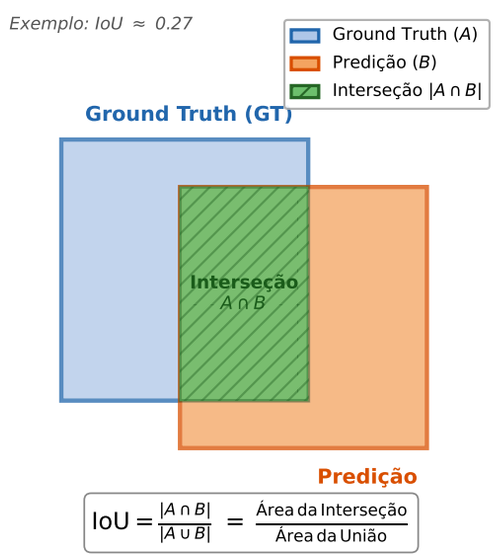


Fig. 1

ILUSTRAÇÃO DA MÉTRICA IOU. A REGIÃO AZUL É O *ground truth* (A), A LARANJA É A PREDIÇÃO (B) E O HACHURADO VERDE REPRESENTA A INTERSEÇÃO ($A \cap B$). O IOU É A RAZÃO ENTRE AS ÁREAS DE INTERSEÇÃO E DE UNIÃO.

IV. RESULTADOS E DISCUSSÃO

A Tabela I apresenta o F1-score macro nas três métricas de avaliação para cada configuração modelo-condição. A Tabela II detalha os resultados por classe para o critério IoU $\geq$ 0,5.

O modo *few-shot* elevou o desempenho dos dois modelos em todas as métricas. O GPT-5.5 passou de F1 0,388 para 0,560 em IoU $\geq$ 0,5 (+0,172), enquanto o Gemini 3.1 Pro saltou de 0,240 para 0,549 (+0,309), o maior ganho absoluto observado. Esse resultado indica que exemplos visuais calibram o limiar de decisão dos modelos para o domínio aéreo, contexto significativamente distante das imagens que predominam nos dados de treinamento desses sistemas.

TABELA I

F1-SCORE MACRO POR MÉTRICA E CONFIGURAÇÃO (MÉDIA $\pm$ DP; BOOTSTRAP n=1000, REAMOSTRAGEM DAS 28 IMAGENS AVALIADAS COM REPOSIÇÃO; SUPORTE *few-shot* EXCLUÍDO).

Modelo	Cond.	IoU$\geq$0,5	IoU$\geq$0,25	Center Hit
GPT-5.5	ZS	$0{,}388 \pm 0{,}029$	$0{,}506 \pm 0{,}028$	$0{,}514 \pm 0{,}028$
GPT-5.5	FS	$0{,}560 \pm 0{,}029$	$0{,}692 \pm 0{,}033$	$0{,}708 \pm 0{,}030$
Gemini 3.1 Pro	ZS	$0{,}240 \pm 0{,}045$	$0{,}377 \pm 0{,}056$	$0{,}420 \pm 0{,}059$
Gemini 3.1 Pro	FS	$0{,}549 \pm 0{,}040$	$0{,}728 \pm 0{,}041$	$\mathbf{0{,}781 \pm 0{,}036}$

TABELA II

F1-SCORE POR CLASSE (IOU $\geq$ 0,5; 28 IMAGENS AVALIADAS).

		Barraca			Construção			Barco		
Modelo	Cond.	P	R	F1	P	R	F1	P	R	F1
GPT-5.5	ZS	0,26	0,14	0,19	0,40	0,55	0,46	0,53	0,52	0,53
GPT-5.5	FS	0,65	0,60	0,62	0,43	0,59	0,50	0,57	0,57	0,57
Gemini 3.1 Pro	ZS	0,35	0,24	0,29	0,23	0,22	0,22	0,27	0,17	0,21
Gemini 3.1 Pro	FS	**0,80**	**0,83**	**0,81**	0,49	0,49	0,49	0,41	0,30	0,35

Em modo *zero-shot*, o GPT-5.5 superou o Gemini 3.1 Pro com margem expressiva (0,388 vs. 0,240 em F1 macro), sugerindo maior robustez a instruções em linguagem natural sem exemplos. Com *few-shot*, os dois modelos convergiram para desempenho similar ($\approx$0,55), indicando que o benefício dos exemplos é maior para o Gemini, que parte de uma base *zero-shot* mais baixa.

O comportamento varia consideravelmente entre classes, e o contraste mais expressivo é o da barraca: em *zero-shot*, os dois modelos produziram F1 abaixo de 0,30 para essa classe (0,19 e 0,29), mas em *few-shot* o Gemini 3.1 Pro atingiu F1 0,81 com precisão 0,80 e revocação 0,83, o melhor resultado individual de todo o experimento, como evidenciado na Fig. 2. Trata-se de um objeto visualmente ambíguo quando visto do ar, o que ajuda a explicar o ganho desproporcional ao fornecer referências explícitas. Para barcos e construções o cenário é distinto: o GPT-5.5 manteve vantagem em barcos (F1 0,57 vs. 0,35 do Gemini), possivelmente por maior familiaridade com formas alongadas em perspectiva aérea, ao passo que ambos os modelos acumularam falsos positivos elevados em construções, provavelmente pela variação de telhados e pela semelhança com superfícies naturais de textura similar. Em aplicações práticas, essas assimetrias sugerem que a escolha do modelo pode ser orientada pela classe de interesse prioritária.

A comparação entre as três métricas revela uma assimetria sistemática: em todas as configurações, o F1 por *Center Hit* supera o F1 por IoU $\geq$ 0,5, com diferenças que variam de 12,6 a 23,2 pontos percentuais. Essa diferença não decorre de desempenho melhor, mas de que o critério IoU $\geq$ 0,5 penaliza detecções cujo centro foi corretamente posicionado mas cuja caixa não coincide com precisão suficiente com a anotação de referência.

O caso mais expressivo é o Gemini 3.1 Pro em modo *few-shot*: 78,1% de F1 por *Center Hit* contra 54,9% por IoU $\geq$ 0,5. Em termos concretos, quase um a cada três objetos que o Gemini localizou corretamente não foi computado como acerto pelo critério padrão, simplesmente por insuficiência de sobreposição geométrica. O GPT-5.5 FS exibe

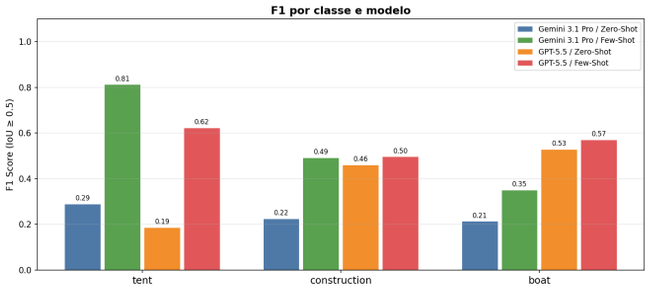


Fig. 2

F1-SCORE POR CLASSE E CONFIGURAÇÃO (IOU $\geq$ 0,5). CLASSES EM INGLÊS CONFORME RÓTULOS DO CONJUNTO DE DADOS: *tent* = BARRACA, *construction* = CONSTRUÇÃO, *boat* = BARCO.

diferença menor (14,8%), indicando que esse modelo produz caixas dimensionalmente mais próximas das anotações humanas. Mesmo ao relaxar o critério para IoU $\geq$ 0,25, o Gemini 3.1 Pro FS registra F1 0,728 contra 0,781 do *Center Hit*, diferença remanescente de 5,3%, causada por detecções cujo centro está dentro do objeto mas cuja caixa é consideravelmente mais larga ou mais estreita do que a anotada.

Os dados apontam, portanto, para uma dissociação entre duas capacidades distintas: os VLMs avaliados identificam onde o objeto está com razoável consistência, mas encontram dificuldade em estimar as dimensões precisas da caixa. Essa separação só se torna visível quando a avaliação inclui a métrica *Center Hit*.

A Fig. 5 ilustra o volume de detecções por imagem de cada configuração em comparação ao *ground truth*. Nota-se que o modo *few-shot* aproxima substancialmente a curva de detecções dos modelos à curva do GT, enquanto o *zero-shot* tende a subestimar ou superestimar o número de objetos dependendo da complexidade da cena.

A Fig. 6 apresenta as matrizes de confusão das quatro configurações. As linhas representam a classe predita e as colunas a classe real. A coluna "*Background*" agrupa predições sem correspondência no GT (falsos positivos de fundo), e a linha "*Background*" agrupa objetos do GT não detectados (falsos negativos). Os valores em cada célula indicam contagens absolutas e proporção por linha.

A análise revela que a principal fonte de erro não é a confusão entre classes, uma vez que os valores fora da diagonal principal entre classes reais são residuais, mas sim os falsos positivos de fundo e os falsos negativos. Em modo *zero-shot*, o GPT-5.5 gerou 141 predições sem correspondência no GT ("*Background*"), predominantemente na classe *const* (66 FPs), enquanto o Gemini 3.1 Pro acumulou 221 GT sem detecção (FN), reflexo de sua menor sensibilidade sem exemplos. O modo *few-shot* reduziu consideravelmente a confusão semântica: o Gemini 3.1 Pro FS registrou apenas 1 caso de confusão entre classes, contra 22 em modo *zero-shot*.

Os resultados posicionam os VLMs avaliados como *detectores de atenção*: sistemas capazes de indicar onde, na imagem, os objetos de interesse provavelmente estão, mas não de delimitar suas fronteiras com a precisão exigida por detectores supervisionados. Essa caracterização tem implicações diretas para o *design* de *pipelines* de anotação semi-automática.

Em um fluxo de trabalho com revisão humana, o anotador não parte do zero para cada imagem. O VLM gera propostas de regiões que o especialista então confirma, descarta ou ajusta. Dado que os modelos demonstram *Center Hit* em 71%–78% dos casos no modo *few-shot*, a maior parte dos objetos presentes já estará indicada no conjunto de propostas. O custo remanescente para o anotador é, predominantemente, o ajuste de bordas, tarefa que, em estudos de anotação assistida, é estimada como três a cinco vezes mais rápida do que a delimitação completa [17].

V. CONCLUSÃO

Este trabalho avaliou os modelos GPT-5.5 e Gemini 3.1 Pro para detecção de objetos em imagens aéreas da Amazônia sob condições *zero-shot* e *few-shot*, considerando três classes de interesse em um conjunto de 28 imagens anotadas manualmente, mantidas independentes de duas imagens usadas exclusivamente como suporte *few-shot*. Além das métricas convencionais baseadas em IoU, foi empregada a métrica *Center Hit*, inspirada em abordagens de *visual grounding* [10], [11], com o objetivo de dissociar a capacidade de localização semântica da precisão geométrica das caixas delimitadoras.

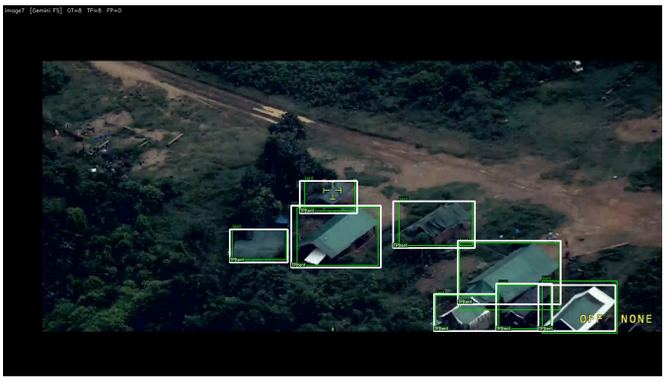
Os resultados evidenciam uma diferença substancial entre essas duas capacidades nos VLMs avaliados. O Gemini 3.1 Pro em modo *few-shot*, por exemplo, alcançou aproximadamente 78% de F1 por *Center Hit*, mas apenas 55% por IoU $\geq$ 0,5. Essa discrepância indica que os modelos frequentemente identificam corretamente a posição dos objetos, mas apresentam limitações na estimativa precisa das dimensões e fronteiras das caixas delimitadoras. Dessa forma, a métrica *Center Hit* fornece uma interpretação mais informativa do desempenho dos VLMs do que métricas baseadas exclusivamente em sobreposição.

O *prompting few-shot* mostrou-se fundamental para o desempenho dos modelos, produzindo ganhos de até 0,31 em F1 macro. Os resultados sugerem que exemplos visuais representativos são particularmente importantes em imagens aéreas, cujas características diferem substancialmente das imagens presentes nos conjuntos de treinamento de modelos multimodais de propósito geral.

Os achados indicam que VLMs podem atuar como ferramentas de pré-anotação capazes de identificar regiões de interesse com elevada cobertura semântica, reduzindo o esforço humano necessário para a construção de bases anotadas. Nesse cenário, o papel do anotador passa a concentrar-se principalmente no refinamento geométrico das caixas geradas automaticamente. Como trabalhos futuros, pretende-se avaliar esse fluxo em conjuntos de dados mais extensos, quantificar a redução efetiva do tempo de anotação e investigar estratégias de adaptação dos modelos, incluindo ajuste fino com dados produzidos pelo próprio *pipeline*, visando melhorar a precisão geométrica das detecções.

Observa-se ainda que a análise deste trabalho foi limitada a 28 imagens aéreas de região amazônica, com apoio de apenas duas imagens de suporte *few-shot*, e a capacidade de generalização para outros ambientes aéreos (urbano, costeiro) e para conjuntos de suporte mais numerosos ou diversos requer validação.

(a) Gemini 3.1 Pro *zero-shot*: 3 TP / 5 FP



(b) Gemini 3.1 Pro *few-shot*: 8 TP / 0 FP

Fig. 3

EFEITO DO *prompting few-shot* PARA A CLASSE BARRACA (IMAGE7, 8 *ground truth* (GT); IMAGEM FORA DO CONJUNTO DE SUPORTE). CAIXAS COLORIDAS = GT; BORDAS BRANCAS = *true positive* (TP); TRACEJADO VERMELHO = *false positive* (FP). EM *zero-shot*, O GEMINI 3.1 PRO DETECTA 3 DAS 8 BARRACAS, COM 5 FALSOS POSITIVOS; EM *few-shot*, DETECTA TODAS AS 8 SEM NENHUM FALSO POSITIVO.

(a) Gemini 3.1 Pro *zero-shot*: 0 TP / 12 FP

(b) Gemini 3.1 Pro *few-shot*: 8 TP / 4 FP

Fig. 4

EFEITO DO *prompting few-shot* PARA CONSTRUÇÕES E BARCOS (IMAGE30, 12 GT, IMAGEM ANONIMIZADA, FORA DO CONJUNTO DE SUPORTE). EM *zero-shot*, O GEMINI 3.1 PRO NÃO ACERTA NENHUM DOS 12 OBJETOS E EMITE 12 FALSOS POSITIVOS; EM *few-shot*, DETECTA 8 DOS 12 OBJETOS, COM 4 FALSOS POSITIVOS REMANESCENTES.

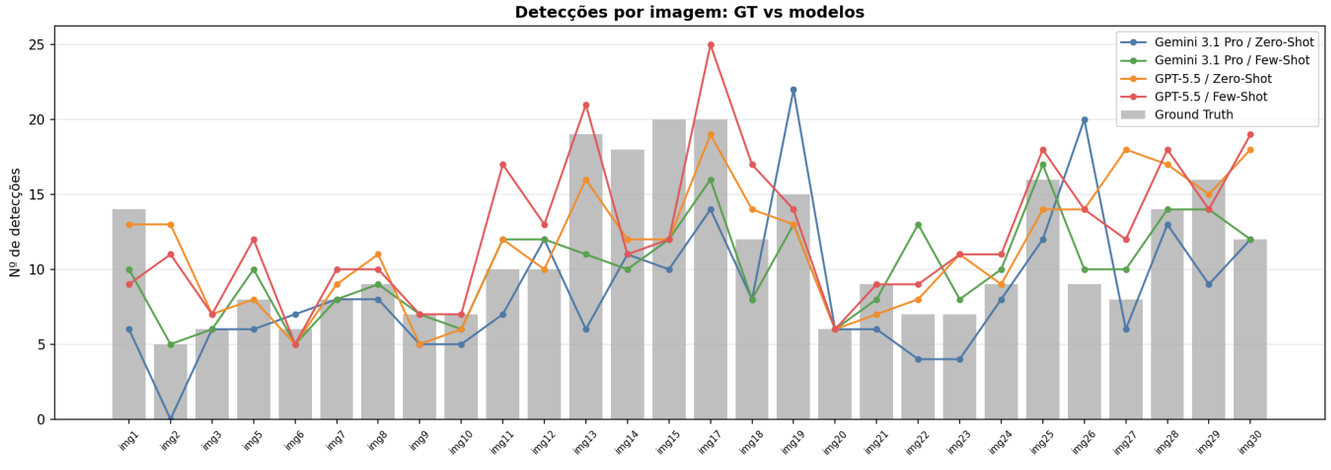


Fig. 5

NÚMERO DE DETECÇÕES POR IMAGEM PARA CADA CONFIGURAÇÃO EM COMPARAÇÃO AO GT, NAS 28 IMAGENS AVALIADAS (`IMAGE4` E `IMAGE16`, SUPORTE *few-shot*, OMITIDAS DO EIXO). O MODO *few-shot* PRODUZ PERFIS DE CONTAGEM MAIS PRÓXIMOS DO GT EM AMBOS OS MODELOS.

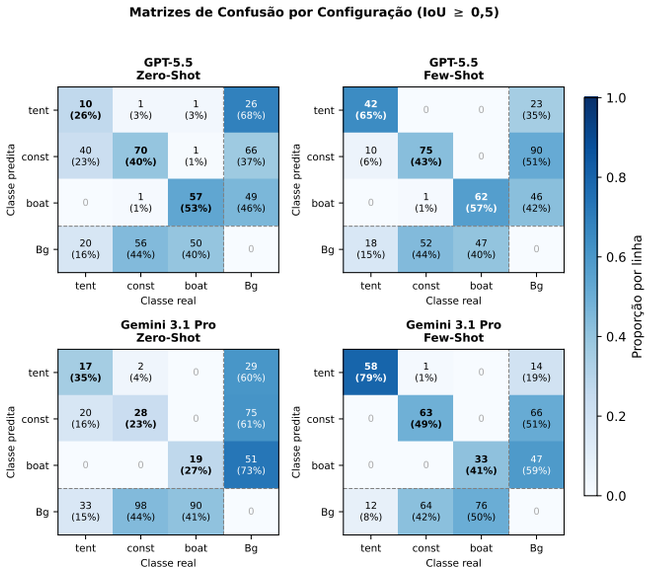


Fig. 6

MATRIZES DE CONFUSÃO PARA AS QUATRO CONFIGURAÇÕES (IOU $\geq$ 0,5). LINHAS = CLASSE PREDITA; COLUNAS = CLASSE REAL. "*Background*" (BG): COLUNA = FP SEM GT CORRESPONDENTE; LINHA = GT NÃO DETECTADO (FN). VALORES: CONTAGEM ABSOLUTA E PROPORÇÃO POR COLUNA.

REFERÊNCIAS

[1] Z. Zang, C. Lin, C. Tang, T. Wang, J. Lv, "Zero-shot aerial object detection with visual description regularization," *arXiv preprint arXiv:2402.18233*, 2024.

[2] M. Quoos, S. Kay, J. Wajoras, R. Field, E. V. Ntagiou, F. Mohammad, Y. Gao, "Survey on AI-enabled computer vision technologies and applications for space robotic missions," *Journal of Field Robotics*, 2026.

[3] G. Jocher, A. Chaurasia, J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[4] M. L. Ali Z. Zhang, "The YOLO framework: a comprehensive review of evolution, applications, and benchmarks in object detection," *Computers*, vol. 13, no. 12, p. 336, 2024.

[5] D. M. M. da Silva, D. I. Alves, S. N. C. Leite, "Classificação de áreas de garimpo ilegal na Amazônia com algoritmos de visão computacional," in *Proc. Simpósio de Aplicações Operacionais em Áreas de Defesa (SIGE)*, São José dos Campos, SP, Brasil, 2025.

[6] R. Sapkota M. Karkee, "Object detection with multimodal large vision-language models: an in-depth review," *Information Fusion*, vol. 126, p. 103575, 2026.

[7] M. Liu, C. Chen, D. Gurari, "Evaluation of GPT-4V and Gemini in online VQA," *arXiv preprint arXiv:2312.10637*, 2024.

[8] Y. Wu, Y. Wang, S. Tang, W. Wu, T. He, W. Ouyang, P. Torr, J. Wu, "DetToolChain: A new prompting paradigm to unleash detection ability of MLLM," in *European Conference on Computer Vision (ECCV)*, 2024.

[9] A. Barzilai, Y. Gigi, A. Helmy, V. Silverman, Y. Refael, B. Jaber, T. Shekel, G. Leifman, G. Beryozkin, "A recipe for improving remote sensing VLM zero shot generalization," *arXiv preprint arXiv:2503.08722*, 2025.

[10] J. Zhang, Z. Lin, J. Brandt, X. Shen, S. Sclaroff, "Top-down neural attention by excitation backprop," *International Journal of Computer Vision*, vol. 126, no. 10, pp. 1084–1102, 2018.

[11] L. Yu, P. Poirson, S. Yang, A. C. Berg, T. L. Berg, "Modeling context in referring expressions," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 69–85.

[12] OpenAI, "GPT-5.5 system card," 2026. [Online]. Available: https://openai.com/index/gpt-5-5-system-card

[13] Google DeepMind, "Gemini: a family of highly capable multimodal models," 2024. [Online]. Available: https://deepmind.google/technologies/gemini/

[14] S. Al-Hamadani, "When does supervised training pay off? The hidden economics of object detection in the era of vision-language models," *arXiv preprint arXiv:2510.11302*, 2025.

[15] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 1877–1901.

[16] G. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, L. Zhang, "DOTA: A large-scale dataset for object detection in aerial images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3974–3983.

[17] J. Deng, X. Li, Y. Fang, "Few-shot object detection on remote sensing images," *arXiv preprint arXiv:2006.07826*, 2020.

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, "The Pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

APÊNDICE

Os prompts abaixo foram utilizados de forma idêntica para GPT-5.5 e Gemini 3.1 Pro. A saída de cada modelo foi filtrada por expressão regular para extrair apenas linhas no formato YOLO válido (`<class_id> <cx> <cy> <w> <h>`).

A.1 Zero-Shot

```
Você é um assistente de análise de imagens
aéreas. Detecte objetos na imagem e retorne
as bounding boxes no formato YOLO.

Saída: uma linha por detecção no formato:
<class_id> <cx> <cy> <largura> <altura>
Valores normalizados entre 0.0 e 1.0.
Retorne APENAS as linhas de detecção.
Sem explicações, sem texto extra.

Classes:
0 - tent (barraca): estrutura de lona azul
    ou preta, formato irregular, pequena.
1 - construction (construção): telhado
    retangular metálico, claro ou escuro.
2 - boat (barco): embarcação alongada
    sobre a água ou margem do rio.
```

A.2 Few-Shot

O modo *few-shot* utiliza um histórico de conversa multi-turno. O prompt de sistema e o prompt do usuário na consulta final são apresentados abaixo. Entre eles, três pares de referência (imagem + anotação YOLO) são intercalados como turnos `user`/`assistant`.

Prompt de sistema:

```
Você é um assistente especializado em análise
de imagens aéreas. Detecte objetos e retorne
bounding boxes no formato YOLO.
Retorne APENAS linhas de detecção, sem texto.
Formato: <class_id> <cx> <cy> <largura> <altura>
(valores normalizados entre 0.0 e 1.0)
```

Estrutura dos turnos de referência (uma imagem de suporte por classe):

```
[user]   "Exemplo de <classe> -- detecte os objetos:"
         + <imagem de referência>
[assist] <anotações YOLO (classe filtrada)>
```

As imagens de suporte foram: `image4.jpg` (24 instâncias de barraca) e `image16.jpg` (12 de construção, 10 de barco), ambas excluídas das 28 imagens avaliadas.

Prompt do usuário na consulta final:

```
Detecte os objetos nesta imagem nas classes:
0 - tent (barraca): lona azul ou preta,
    formato irregular, pequena.
1 - construction (construção): telhado
    retangular metálico.
2 - boat (barco): embarcação alongada
    sobre a água ou margem.
Formato: <class_id> <cx> <cy> <largura> <altura>
```