Análise Comparativa de Modelos para Detecção de Assinatura em Imagens

José Gabriel Nardes França[1] **, Jonathan de Andrade Silva**[1]

[1]Faculdade de Computação (FACOM) – Universidade Federal de Mato Grosso do Sul (UFMS)
Campo Grande – MS – Brasil

josegabrielnf404@gmail.com, jonathan.andrade@ufms.br

Abstract. *In real-world applications, such as fraud prevention systems or process automation, the ability to compare signatures between different documents is crucial. The first and most critical step in this pipeline is to accurately locate all signatures present on a scanned page. This paper addresses this initial stage by conducting a comparative analysis of six modern object detection models: YOLOv12, DINO, RetinaNet, Faster R-CNN, Double Heads, and VFNet. Using a public dataset of annotated documents and a standardized experimental protocol, we evaluate which model is most effective for the specific task of signature detection. The results provide a solid foundation for developing robust automatic signature analysis systems, indicating the most suitable architectures for this essential detection phase.*

Resumo. *Em aplicações reais, como sistemas de prevenção a fraudes ou de automação de processos, a capacidade de comparar assinaturas entre diferentes documentos é fundamental. A primeira e mais crítica etapa nesse fluxo de trabalho é localizar com precisão todas as assinaturas presentes em uma página digitalizada. Este trabalho aborda justamente essa fase inicial, realizando uma análise comparativa de seis modelos modernos de detecção de objetos: YOLOv12, DINO, RetinaNet, Faster R-CNN, Double Heads e VFNet. Utilizando uma base de dados pública de documentos anotados e um protocolo experimental padronizado, avaliamos qual modelo é mais eficaz para a tarefa específica de detecção de assinaturas. Os resultados fornecem uma base sólida para o desenvolvimento de sistemas robustos de análise automática de assinaturas, indicando as arquiteturas mais adequadas para essa etapa essencial de detecção.*

1. Introdução

A verificação de autenticidade de documentos em ambientes corporativos, jurídicos e bancários frequentemente depende da análise de assinaturas manuscritas [Pervouchine and Leedham 2006]. Um sistema automatizado para essa finalidade precisa, em primeiro lugar, responder a uma pergunta fundamental: onde estão as assinaturas no documento? A tarefa de localizar e extrair essas regiões de interesse é o alicerce para qualquer análise subsequente, como a comparação entre duas assinaturas para verificar se pertencem à mesma pessoa. Este trabalho foca precisamente nessa etapa fundacional, avaliando sistematicamente modelos de ponta para a detecção de assinaturas em imagens de documentos.

A motivação para este estudo surge de necessidades práticas concretas. Considere um sistema projetado para validar a legitimidade de um contrato, comparando a assinatura

presente com uma outra de referência. Antes que qualquer comparação possa ocorrer, o sistema deve primeiro identificar a localização exata da assinatura em ambos os documentos. Outra aplicação relevante é a triagem automática: um sistema pode verificar se um lote de formulários foi devidamente assinado, rejeitando aqueles sem assinatura antes de encaminhá-los para a validação humana. Em ambos os cenários, a falha na etapa de detecção compromete todo o fluxo de trabalho.

Do ponto de vista técnico, a detecção de assinaturas é um problema de detecção de objetos. No entanto, as assinaturas possuem características que as tornam um desafio único. Elas podem variar drasticamente em estilo, tamanho e cor. Frequentemente, assemelham-se a texto cursivo, rabiscos, ou podem estar sobrepostas a linhas de formulário, carimbos e outros ruídos visuais presentes em documentos digitalizados [Zhai et al. 2017, Yan et al. 2022]. Essa complexidade exige modelos robustos, capazes de generalizar a partir de diferentes padrões visuais.

Os avanços recentes em *deep learning* e, mais especificamente, em algoritmos de detecção de objetos, oferecem ferramentas poderosas para resolver esse problema. Contudo, a vasta gama de arquiteturas disponíveis levanta uma questão importante: qual delas é a mais adequada para as peculiaridades da detecção de assinaturas?

Para responder a essa pergunta, este trabalho realiza uma análise comparativa do desempenho de seis modelos de detecção de objetos de última geração: YOLOv12, DINO, RetinaNet, Faster R-CNN, Double Heads e VFNet. Nosso objetivo é identificar o modelo mais eficaz para ser incorporado como o primeiro módulo em sistemas de verificação automática de documentos. Para isso, utilizamos um protocolo de avaliação rigoroso, um conjunto de dados público e métricas padronizadas, como mAP, precisão e *recall*. Ao final, discutimos as vantagens e limitações de cada abordagem, oferecendo uma orientação clara para o desenvolvimento de aplicações práticas e futuras pesquisas na área.

2. Fundamentação Teórica

Nesta seção, revisamos os conceitos de detecção de objetos e as arquiteturas empregadas neste trabalho. Adicionalmente, apresentamos uma breve revisão da literatura sobre pesquisas recentes na tarefa de detecção automática de assinaturas manuscritas, contextualizando nosso estudo no cenário científico atual.

2.1. Detecção de Objetos – Visão Geral

A detecção de objetos visa identificar e localizar instâncias de categorias de interesse em imagens, geralmente delimitando-as com caixas (bounding boxes) e atribuindo-lhes um rótulo de classe. Os métodos modernos podem ser amplamente agrupados em três categorias principais, cada uma representada pelos modelos que avaliamos.

Primeiro, temos os detectores de estágio único (*one-stage*), como YOLO e RetinaNet. Eles tratam a detecção como um problema de regressão direta, prevendo as coordenadas da caixa e a classe em uma única passagem pela rede, o que os torna extremamente rápidos.

Em segundo lugar, estão os detectores de dois estágios (*two-stage*), como a família R-CNN. Esses métodos primeiro propõem regiões de interesse (ROIs) onde um objeto pode estar e, em seguida, uma segunda rede classifica e refina a localização dessas propostas. Historicamente, essa abordagem oferece maior precisão ao custo de menor velocidade.

Por fim, uma vertente mais recente são os detectores baseados em *Transformers*, como o DETR e suas variantes. Inspirados pelo sucesso em processamento de linguagem natural, esses modelos utilizam mecanismos de atenção para tratar a detecção como um problema de predição de conjuntos, eliminando a necessidade de componentes manuais como as propostas de região.

A seguir, detalhamos os modelos específicos selecionados para este estudo.

2.2. Detectores de Estágio Único (One-Stage)

RetinaNet: Proposto por Lin et al. [Lin et al. 2017], o RetinaNet foi um marco por resolver a lacuna de desempenho que existia entre detectores de um e dois estágios. Sua principal contribuição foi a *Focal Loss*, uma função de perda que foca o treinamento em exemplos difíceis, mitigando o severo desequilíbrio entre o fundo da imagem e os objetos de interesse, um problema comum em detectores *one-stage*.

VFNet (VarifocalNet): O VFNet [Zhang et al. 2021] evolui a partir de detectores como o RetinaNet com foco em melhorar o ranqueamento das predições. Ele introduz uma pontuação de classificação que considera a qualidade da localização da caixa (IoU-aware Classification Score) e uma função de perda correspondente, a *Varifocal Loss*. Isso permite que o modelo priorize detecções que não são apenas da classe correta, mas que também estão bem localizadas.

YOLOv12: A família YOLO é sinônimo de detecção em tempo real. O YOLOv12 [Ultralytics 2023] representa a geração mais recente, incorporando mecanismos de atenção e otimizações de arquitetura para atingir precisão de ponta sem sacrificar a velocidade. É um dos detectores *one-stage* mais avançados e populares atualmente disponíveis.

2.3. Detectores de Dois Estágios (Two-Stage)

Faster R-CNN: Um modelo canônico nos detectores de dois estágios, o Faster R-CNN [Ren et al. 2015] introduziu a Rede de Proposta de Região (RPN). A RPN é uma sub-rede que aprende a gerar propostas de objetos de alta qualidade, unificando o processo de detecção em um único pipeline de ponta a ponta e tornando-o muito mais rápido que seus predecessores.

Double Heads: Inspirado nos trabalhos de Wu et al. [Wu et al. 2020], o método Double Heads propõe uma melhoria na arquitetura de dois estágios. Ele utiliza cabeças de predição distintas e especializadas para as tarefas de classificação e regressão de caixas. A hipótese é que uma cabeça totalmente conectada é melhor para classificação (sensibilidade global), enquanto uma cabeça convolucional é mais adequada para a regressão (sensibilidade espacial), resultando em maior precisão geral.

2.4. Detector Baseado em Transformer

DINO (DETR with Improved DeNoising Anchor boxes): O DETR [Carion et al. 2020] reformulou a detecção de objetos como um problema de predição de conjuntos usando *Transformers*. O DINO [Zhang et al. 2022a] é uma evolução que melhora significativamente a eficiência de treinamento e o desempenho do DETR. Ele introduz técnicas como um treinamento com ruído (*denoising*) e uma melhor seleção de queries, resultando em um dos detectores mais precisos da atualidade.

2.5. Revisão de Literatura

A detecção de assinaturas é um campo ativo de pesquisa. A Tabela 1 resume estudos relevantes, destacando os métodos empregados e as conclusões que informam nosso trabalho.

Tabela 1. Resumo de estudos relevantes em detecção de objetos e assinaturas.

Estudo	Características / Dataset	Método / Implementação	Resultados e Conclusões
Sharma et al. [Sharma et al. 2018]	Tobacco-800, documentos reais.	YOLOv2 vs. Faster R-CNN.	Faster R-CNN supera YOLOv2 em precisão; YOLOv2 permanece mais rápido.
Yan et al. [Yan et al. 2022]	ChiSig (documentos chineses).	Faster R-CNN, YOLOv3, DETR.	Faster R-CNN apresenta o desempenho mais alto no benchmark, à frente de YOLOv3 e DETR.
Hauri [Hauri 2021]	Documentos técnicos chineses.	YOLOv5 com Copy-Paste.	YOLOv5 + Copy-Paste obteve desempenho superior aos demais métodos testados.
Zhang et al. [Zhang et al. 2022b]	Dataset industrial chinês.	YOLOv5 + Copy-Paste.	A estratégia Copy-Paste elevou a precisão do YOLOv5 sem alterar sua arquitetura.
Edozie et al. [Edozie et al. 2025]	Vários datasets públicos (COCO, PASCAL VOC).	Síntese de *one-stage*, *two-stage* e Transformers.	Modelos Transformer (DETR/DINO) entregam a melhor precisão, enquanto YOLO lidera em velocidade.

2.6. Desafios na Detecção de Assinaturas

A tarefa de detectar assinaturas em documentos digitalizados apresenta desafios específicos. As assinaturas podem variar enormemente em aparência, desde nomes legíveis até rubricas estilizadas, muitas vezes com inclinações variadas. Elas podem ser confundidas com outros elementos manuscritos, como datas ou anotações, e frequentemente aparecem sobrepostas a linhas de formulário, selos ou carimbos.

A qualidade da digitalização também é um fator crítico; resoluções baixas podem tornar os traços da assinatura pouco nítidos. Adicionalmente, existe um desequilíbrio de classes extremo: a maior parte de um documento é "fundo", com apenas uma ou poucas pequenas regiões contendo a assinatura. Técnicas como a *focal loss* (RetinaNet) e a *varifocal loss* (VFNet) são particularmente úteis para mitigar o impacto desse desequilíbrio.

Finalmente, a aplicação prática exige alta precisão e *recall*. Uma assinatura não detectada (falso negativo) pode invalidar um processo automatizado, enquanto a detecção de um elemento que não é uma assinatura (falso positivo) pode introduzir erros na etapa seguinte de análise. Portanto, um modelo ideal deve equilibrar bem essas duas métricas para garantir robustez e confiabilidade.

3. Metodologia

3.1. Base de Dados de Assinaturas

Para os experimentos, utilizamos um conjunto de dados público focado na detecção de assinaturas [Tech4Humans 2024]. Ele é composto por 2.819 imagens de documentos com dimensões de 640x640 pixels. Este dataset combina imagens do Tobacco-800 [tob 2006] e de um subconjunto do Roboflow 100, oferecendo grande diversidade de tipos de documentos, qualidades de escaneamento (150 a 300 dpi), estilos de assinatura e orientações.

As imagens foram anotadas com uma única classe, "assinatura". O conjunto de dados foi dividido em: 70% para treinamento (1.973 imagens), 10% para validação (281 imagens) e 20% para teste (565 imagens). Essa partição garante que os modelos sejam treinados e avaliados em uma ampla gama de cenários.

3.2. Treinamento dos Modelos

Para assegurar uma comparação justa entre as diferentes arquiteturas, adotamos um protocolo de treinamento padronizado. A maior parte dos experimentos foi implementada com o *framework* MMDetection [Contributors 2018], uma plataforma aberta que facilita a padronização de experimentos com detectores de objetos, enquanto para o modelo YOLOv12, utilizamos a implementação oficial da Ultralytics [Ultralytics 2023]. Todos os modelos foram inicializados com pesos pré-treinados em grandes datasets (ImageNet ou COCO), uma técnica conhecida como *transfer learning*. O treinamento foi executado por no máximo 50 épocas (exceto para o YOLOv12, que foi estendido para analisar a convergência), com um mecanismo de *early stopping* monitorando o desempenho na validação. A seleção do melhor *checkpoint* de cada arquitetura foi baseada na época que apresentou o melhor resultado de mAP no conjunto de validação.

Os principais hiperparâmetros utilizados foram os seguintes:

- **Otimizador:** SGD com *momentum* de 0,9.
- **Taxa de aprendizado (*learning rate*):**
  - Double Heads e Faster R-CNN: 0.02
  - DINO: 0.0002
  - RetinaNet e VFNet: 0.002
- **Decaimento de peso (*weight decay*):**
  - YOLOv12: 0.001
  - Demais modelos: 0.0001

Os experimentos foram executados na plataforma Google Colaboratory Pro, equipada com uma GPU NVIDIA L4 (24 GB de VRAM), em um ambiente com Python 3.11 e CUDA 12.4. Para fins de reprodutibilidade, o Jupyter Notebook contendo todo o código de treinamento dos modelos está disponível publicamente em um repositório no GitHub [França 2025].

3.3. Avaliação e Métricas

Neste trabalho, serão apresentadas as avaliações de desempenho nos conjuntos de validação e de teste. Enquanto a avaliação no conjunto de validação foi utilizada para a seleção do melhor checkpoint de cada modelo, a avaliação final, que serve como base para a comparação definitiva entre as arquiteturas, foi realizada no conjunto de teste. Em ambas as etapas, uma detecção é considerada correta se o *Intersection over Union* (IoU) entre a caixa prevista e a anotação real (*ground-truth*) for de pelo menos 0,5. Para assegurar uma base de comparação equitativa, a avaliação foi padronizada com a imposição de um limiar de confiança (*score threshold*) de 0,5, de modo que somente predições com essa confiança mínima foram contabilizadas nas métricas.

As métricas utilizadas para comparar o desempenho dos modelos foram:

- **mAP@0.5:** A principal métrica para comparar o desempenho geral. Representa a precisão média calculada com um limiar de IoU de 0,5. Um mAP alto indica que o modelo é bom tanto em encontrar as assinaturas (*recall*) quanto em garantir que suas detecções estão corretas (precisão).
- **Precisão (Precision):** $P = \frac{TP}{TP+FP}$. Mede a proporção de detecções corretas entre todas as detecções feitas. Alta precisão é crucial para evitar que o sistema envie alarmes falsos (ex: confundir um carimbo com uma assinatura) para a próxima etapa de análise.
- **Recall:** $R = \frac{TP}{TP+FN}$. Mede a proporção de assinaturas reais que o modelo conseguiu encontrar. Um *recall* elevado é vital para sistemas de triagem, garantindo que nenhum documento assinado seja incorretamente descartado por falha na detecção.

No contexto específico da detecção de assinaturas, os termos Verdadeiro Positivo (TP), Falso Positivo (FP) e Falso Negativo (FN) são definidos da seguinte forma:

- **Verdadeiro Positivo (TP):** O modelo detecta corretamente uma assinatura que existe no documento. A caixa de detecção prevista possui um IoU $\geq$ 0,5 com a anotação real da assinatura.
- **Falso Positivo (FP):** O modelo aponta uma região como sendo uma assinatura, mas não há uma assinatura real naquele local (ex: confunde um carimbo, uma data ou um rabisco com uma assinatura). Também ocorre quando uma assinatura real é detectada com uma sobreposição insuficiente (IoU < 0,5).
- **Falso Negativo (FN):** Uma assinatura real está presente no documento, mas o modelo não a detecta.

4. Resultados e Discussão

Nesta seção, apresentamos os resultados quantitativos e qualitativos. Analisamos as métricas de desempenho e, em seguida, inspecionamos exemplos visuais para entender os pontos fortes e as limitações de cada modelo.

4.1. Análise Quantitativa

Para entender como cada modelo aprendeu ao longo do tempo, a análise inicia-se pela observação da evolução do seu desempenho e da função de perda durante a etapa de treinamento. As figuras a seguir ilustram visualmente este processo, mostrando a convergência e a estabilidade de cada arquitetura.

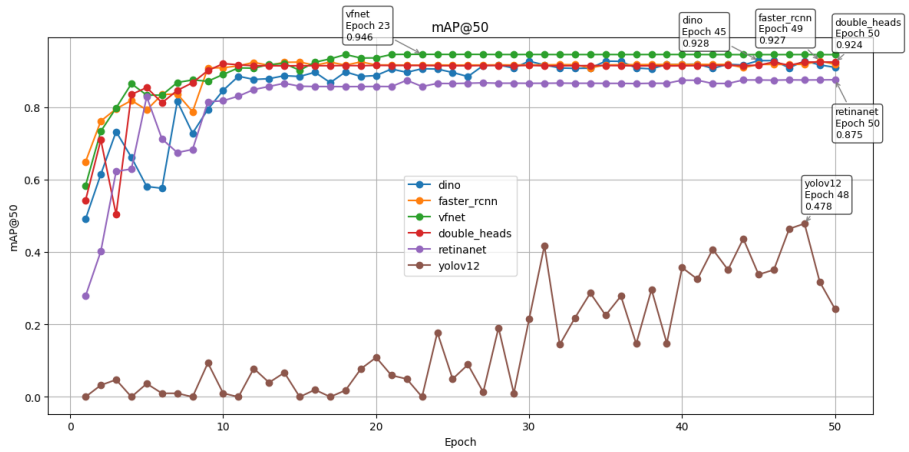

Figura 1. Comparação da evolução do mAP_{50} durante o treinamento no conjunto de validação.

A análise da evolução do mAP_{50} no conjunto de validação, ilustrada na Figura 1, revela padrões de aprendizado distintos entre os modelos. Observa-se que a maioria das arquiteturas apresenta uma convergência rápida, atingindo um **nível de estabilidade** em alto desempenho já a partir da décima época de treinamento.

Neste cenário, o **VFNet** se destacou, alcançando e mantendo consistentemente o maior valor de mAP_{50}. Logo em seguida, os modelos **DINO**, **Faster R-CNN** e **Double Heads** formaram um grupo coeso com desempenho competitivo e valores de mAP_{50} muito próximos entre si. Por outro lado, o **RetinaNet**, embora também tenha convergido rapidamente, estabilizou em um patamar inferior aos demais.

Em um comportamento notavelmente diferente, o **YOLOv12** não atingiu a convergência nas 50 épocas iniciais. Em vez disso, demonstrou uma tendência de aprendizado contínua e gradual, sugerindo que sua arquitetura se beneficia de um período de treinamento mais extenso para alcançar seu desempenho máximo.

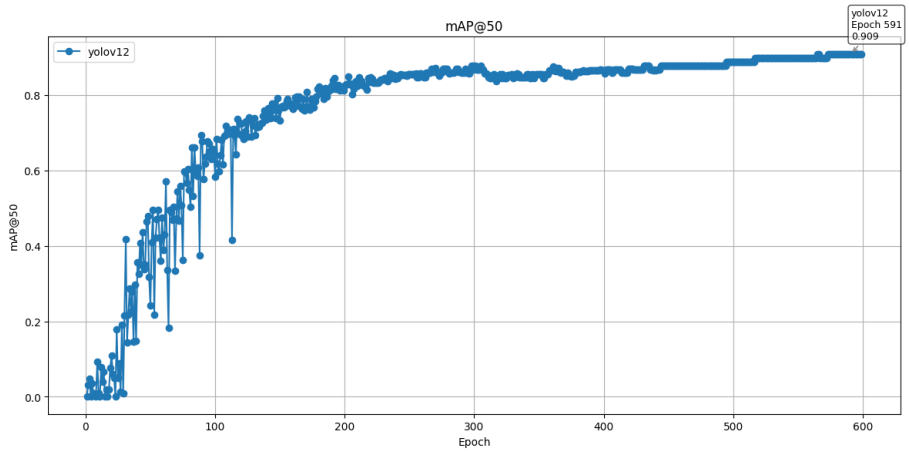

Figura 2. Evolução do mAP_{50} para o YOLOv12, estendido por 600 épocas, no conjunto de validação.

Ao estender seu treinamento por 600 épocas (Figura 2), o modelo alcançou um

mAP_{50} de 90,9%. Este resultado posicionou seu desempenho como superior ao do RetinaNet, embora ainda inferior ao dos demais modelos de ponta (VFNet, DINO, Faster R-CNN e Double Heads).

As curvas de perda (Figura 3) confirmam que todos os modelos tiveram um processo de treinamento estável e bem-sucedido, com a função de perda decrescendo consistentemente, indicando um aprendizado efetivo.

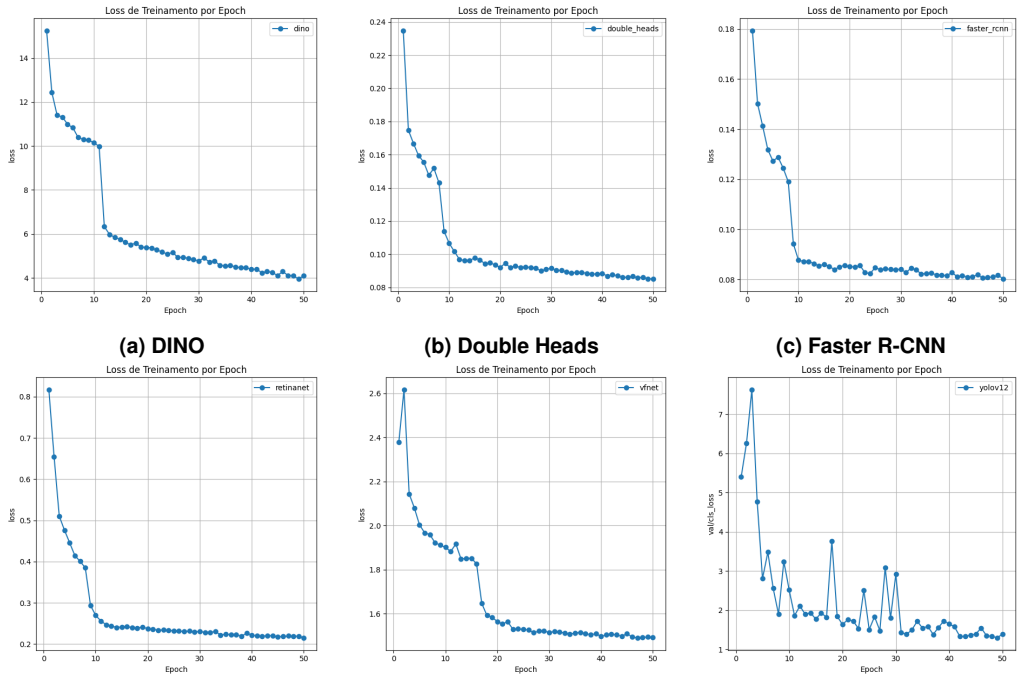

Figura 3. Evolução da perda (*loss*) durante o treinamento para os diferentes modelos.

Com base na análise do treinamento, onde selecionamos o melhor *checkpoint* para cada arquitetura, apresentamos os resultados quantitativos finais. A avaliação foi realizada tanto no conjunto de validação (Tabela 2) quanto no de teste (Tabela 3), garantindo uma análise robusta do desempenho.

Tabela 2. Resumo dos resultados obtidos pelos modelos no conjunto de validação.

Modelo	Época	mAP_{50}	$Recall_{50}$	$Precisão_{50}$
VFNet	23	**94.6%**	**95.4%**	92.5%
DINO	45	92.8%	93.2%	**97.4%**
Faster R-CNN	49	92.7%	94.1%	90.0%
Double Heads	50	92.4%	94.1%	90.5%
YOLOv12	591	90.9%	92.6%	95.8%
RetinaNet	50	87.5%	90.1%	91.3%

Tabela 3. Resumo dos resultados finais obtidos pelos modelos no conjunto de teste.

Modelo	Época	mAP_{50}	$Recall_{50}$	$Precisão_{50}$
VFNet	23	**92.8%**	**94.5%**	90.1%
Double Heads	50	90.9%	93.6%	87.0%
Faster R-CNN	49	89.3%	92.5%	86.2%
YOLOv12	591	88.0%	90.9%	91.6%
DINO	45	87.3%	89.9%	**93.5%**
RetinaNet	50	87.1%	90.1%	90.1%

Comparando os resultados de validação e teste, a tendência de desempenho se mantém consistente. O **VFNet** continua sendo a arquitetura com o melhor desempenho geral, liderando tanto em mAP_{50} (92,8%) quanto em $recall_{50}$ (94,5%) no conjunto de teste. O **DINO**, embora tenha uma queda no mAP, ainda se destaca com a maior $precisão_{50}$ (93,5%), reafirmando sua alta confiabilidade. Essa coerência entre os resultados de validação e teste valida a robustez dos modelos e a generalização do aprendizado.

Para entender o impacto prático desses números, a Tabela 4 detalha a contagem de Verdadeiros Positivos (TP), Falsos Positivos (FP) e Falsos Negativos (FN), revelando o *trade-off* entre sensibilidade e confiabilidade.

Tabela 4. Verdadeiros Positivos (TP), Falsos Positivos (FP) e Falsos Negativos (FN) no conjunto de teste.

Modelo	Época	TP	FP	FN
VFNet	23	**1241**	136	**72**
Double Heads	50	1229	184	84
Faster R-CNN	49	1215	194	98
YOLOv12	591	1193	110	120
RetinaNet	50	1183	130	130
DINO	45	1181	**82**	132

O **VFNet** alcançou o maior número de Verdadeiros Positivos (1241) e, crucialmente, o menor número de Falsos Negativos (72). Isso significa que ele é o modelo que menos deixou de encontrar assinaturas existentes, um fator vital para aplicações de triagem onde um FN levaria à rejeição indevida de um documento válido.

Em contrapartida, o **DINO** se destaca por ter o menor número de Falsos Positivos (82). Em um sistema de validação, um FP significa enviar um elemento incorreto (como um carimbo) para a etapa de comparação, gerando erros e processamento desnecessário. Portanto, o DINO é a escolha ideal para aplicações que exigem a mais alta confiabilidade em cada detecção. Os demais modelos apresentam um equilíbrio intermediário entre essas duas características.

4.2. Análise Qualitativa dos Resultados

A inspeção visual dos resultados oferece *insights* valiosos. Para esta análise, geramos predições onde as anotações de referência (*ground truth*) são verdes e as detecções do modelo são vermelhas.

4.2.1. Casos de Sucesso

Todos os modelos demonstraram grande robustez, detectando corretamente assinaturas em cenários desafiadores como documentos rotacionados, com múltiplas assinaturas, fundos coloridos e até mesmo em imagens de baixa resolução (Figuras 4 a 6). Isso mostra que os modelos aprenderam características visuais generalizáveis e não apenas padrões simples.

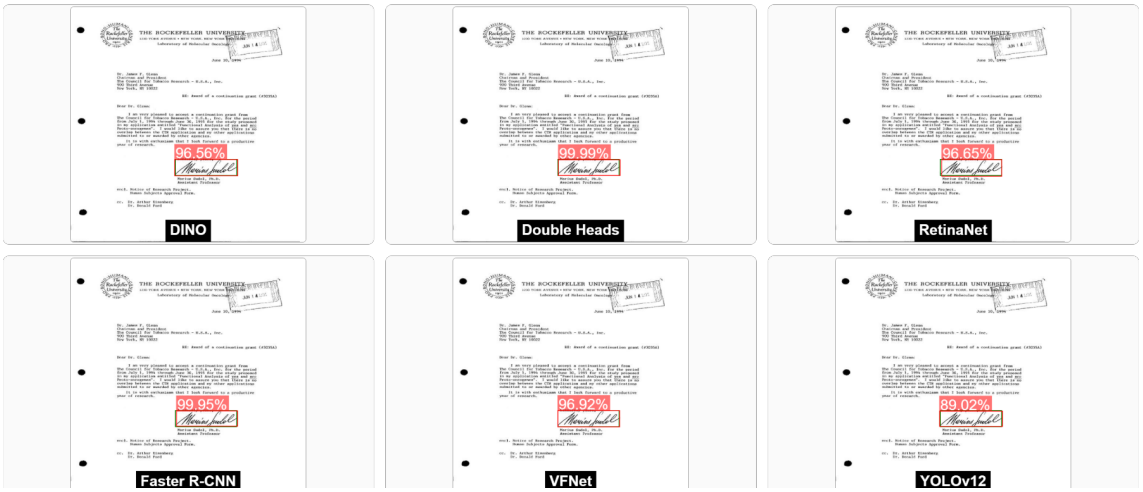

Figura 4. Caso base: todos os modelos detectam corretamente a assinatura.

Figura 5. Sucesso na detecção em documento rotacionado com múltiplas assinaturas.

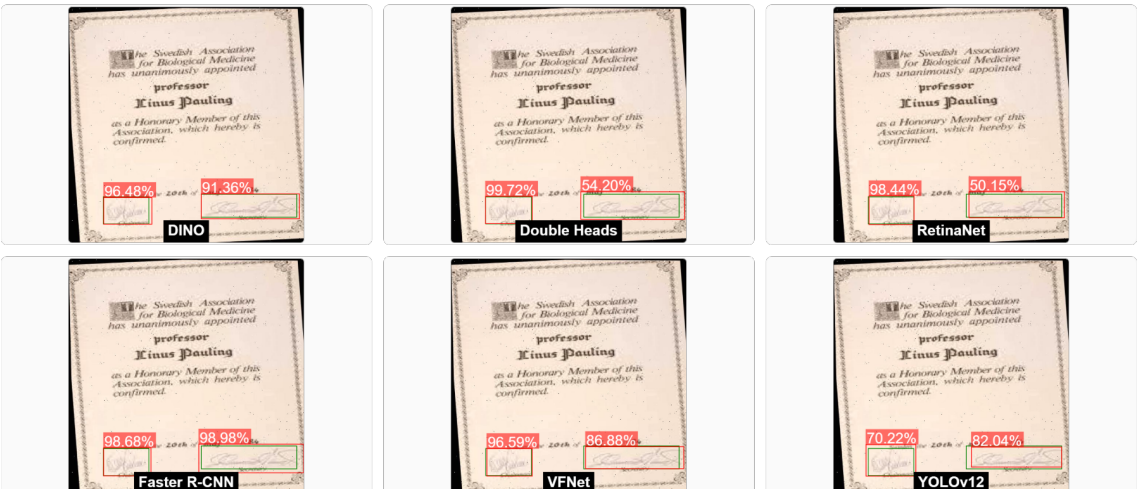

Figura 6. Detecção correta em documento de baixa resolução, mostrando robustez.

É notável a capacidade dos modelos de distinguir assinaturas de outros elementos manuscritos, como datas, o que reduz a incidência de FPs e aumenta a confiabilidade do sistema.

4.2.2. Casos de Falha

As falhas geralmente ocorrem em cenários específicos. Um desafio comum é a detecção de assinaturas muito próximas (Figura 7), onde alguns modelos podem confundir detecções ou gerar caixas sobrepostas

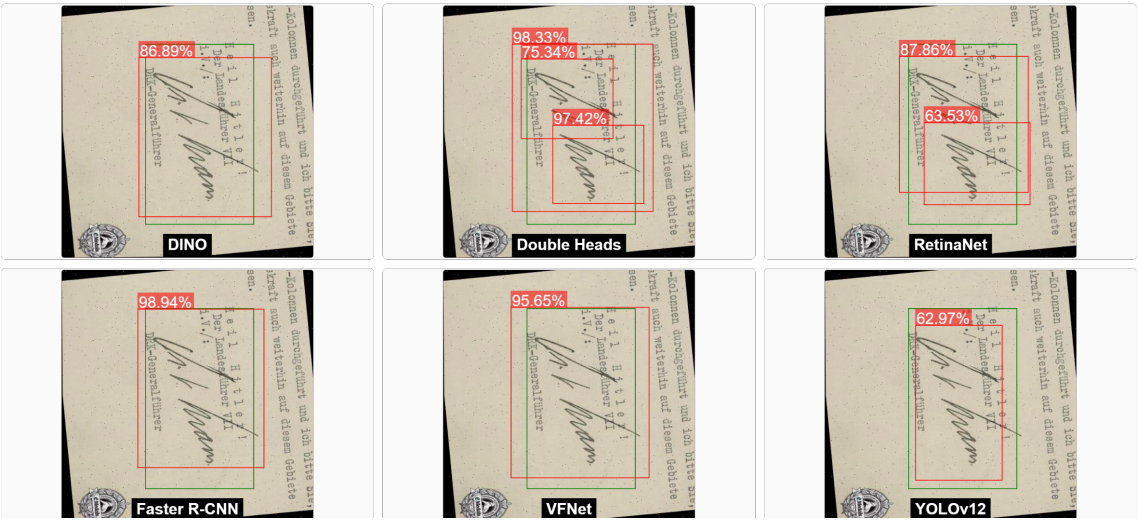

Figura 7. Falhas na detecção de assinaturas muito próximas.

A confusão com outros elementos textuais é outra fonte de erro. Na Figura 8, modelos como Double Heads e Faster R-CNN confundem uma data manuscrita com uma

assinatura, gerando um Falso Positivo. Modelos com maior precisão, como o DINO, são menos propensos a esse tipo de erro.

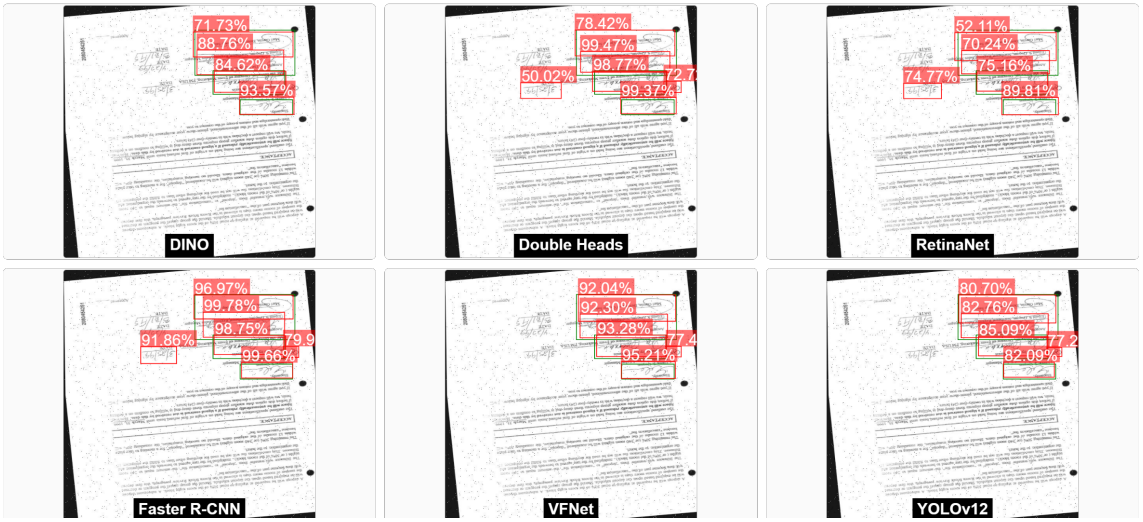

Figura 8. Exemplo de Falso Positivo: modelos confundindo a data com uma assinatura.

Apesar de se destacar pela alta precisão, o DINO demonstrou limitações em sua capacidade de detecção. O exemplo na Figura 9, onde o modelo não identifica assinaturas claras, ilustra o tipo de falha que levou o DINO a obter a maior contagem de Falsos Negativos (132 FNs) entre todos os modelos, conforme detalhado na Tabela 4. Essa tendência a omitir detecções, mesmo em casos aparentemente simples, é um ponto crítico a ser considerado em aplicações que exigem alto recall.

Figura 9. Falha isolada do DINO em detectar duas assinaturas em condições ideais.

Figura 10. Caso de falha geral dos modelos em um documento de baixa resolução, onde nenhuma assinatura foi detectada corretamente.

Figura 11. Caso complexo onde o YOLOv12 falha na detecção, e outros modelos detectam ruídos de fundo como assinaturas.

Na Figura 10, observa-se que nenhum dos modelos foi capaz de detectar corretamente as assinaturas presentes na imagem, apesar do conjunto de dados utilizado conter diversos exemplos com resoluções igualmente reduzidas. Isso demonstra que, embora os modelos tenham sido expostos a imagens de baixa qualidade durante o treinamento, a presença de ruídos extremos e degradação visual severa ainda representa um desafio significativo.

Como contraponto, na Figura 6, é possível verificar um caso em que todos os modelos obtiveram sucesso mesmo em condições de baixa resolução.

No geral, os resultados quantitativos e qualitativos se complementam, mostrando que, embora os modelos modernos sejam poderosos, ainda existem casos de borda que exigem atenção.

5. Conclusão

Este trabalho realizou uma análise comparativa abrangente de seis modelos modernos de detecção de objetos, com o objetivo de identificar a arquitetura mais eficaz para a localização de assinaturas manuscritas em documentos. Os resultados revelam um claro *trade-off* entre a capacidade de detecção (cobertura) e a confiabilidade das predições (precisão), permitindo-nos oferecer recomendações direcionadas para diferentes cenários de aplicação.

O **VFNet** destacou-se como o modelo de melhor desempenho geral, alcançando o maior **mAP**$_{50}$ (92,8%), o maior **recall**$_{50}$ (94,5%) e, de forma crucial, o menor número de falsos negativos (**FN = 72**). Isso o consagra como a escolha ideal para sistemas de triagem ou qualquer aplicação onde a prioridade máxima seja garantir que nenhuma assinatura presente no documento seja omitida.

Em contrapartida, o **DINO** se sobressaiu pela maior **precisão**$_{50}$ (93,5%) e pelo menor índice de falsos positivos (**FP = 82**). Contudo, essa alta confiabilidade tem um custo: o modelo registrou a maior contagem de falsos negativos (**FN = 132**), indicando que, embora suas detecções sejam precisas, ele também apresenta uma maior propensão a não identificar assinaturas importantes que outros modelos capturam.

Os modelos **Double Heads** e **Faster R-CNN** apresentaram um desempenho muito semelhante, sem diferenças significativas em suas métricas para o escopo deste trabalho, posicionando-se como alternativas robustas e equilibradas. Já o **YOLOv12**, mesmo com resultados de mAP inferiores aos dos modelos de ponta, demonstrou uma evolução notável. Seus resultados se aproximam dos de arquiteturas mais complexas e são um avanço claro em relação a versões anteriores como YOLOv2 e YOLOv5, citadas na revisão de literatura, o que evidencia a maturação e a competitividade crescente da família YOLO.

Este estudo, portanto, cumpre seu objetivo de estabelecer uma base sólida para a primeira e crucial etapa de um sistema completo de análise de assinaturas. A escolha do detector mais adequado dependerá dos requisitos específicos da aplicação: maximizar a detecção com o VFNet ou garantir a mais alta confiabilidade — ciente do risco de omissões — com o DINO.

Como trabalhos futuros, o passo natural subsequente é a **comparação de assinaturas** (*Offline Signature Comparison*). Essa tarefa consiste em analisar duas ou mais regiões de assinatura, previamente detectadas pelos modelos aqui avaliados, para determinar seu grau de similaridade. O objetivo final é verificar se foram feitas pela mesma pessoa, detectando possíveis falsificações.

A comparação de assinaturas é um problema complexo devido à alta variabilidade na assinatura de uma mesma pessoa e à baixa variabilidade no caso de falsificações habilidosas (*skilled forgeries*). Abordagens de *deep learning*, como **Redes Siamesas (*Siamese Networks*)** [Dey et al. 2017], são promissoras, pois aprendem a mapear assinaturas em um espaço de características onde a similaridade pode ser medida de forma robusta. A combinação de um detector de alto desempenho, como os identificados neste trabalho, com uma rede siamesa representa um caminho promissor para construir um sistema de validação de assinaturas de ponta a ponta.

Referências

(2006). Tobacco-800 dataset. http://lamp.cfar.umd.edu/projects/tobacco_data. University of Maryland, LAMP.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, pages 213–229.

Contributors, M. (2018). Mmdetection: Open mmlab detection toolbox and benchmark. https://github.com/open-mmlab/mmdetection. Acesso em: 2024-06-10.

Dey, S., Dutta, A., Lladós, J., and Pal, U. (2017). Signet: Convolutional siamese network for writer independent offline signature verification. *arXiv preprint arXiv:1707.02131*.

Edozie, E., Shuaibu, A. N., John, U. K., and Sadiq, B. O. (2025). Comprehensive review of recent developments in visual object detection based on deep learning. https://doi.org/10.1007/s10462-025-11284-w. Artificial Intelligence Review, vol.58, Art.277, open access.

França, J. G. N. (2025). Repositório de notebooks e scripts de treinamento de modelos para detecção de assinatura. Disponível em: https://github.com/JoseGabrielNF/deteccao-de-assinaturas/.

Hauri, M. R. (2021). Detecting signatures in scanned document images. Mestrado, Technische Universität Wien. Disponível em: https://repositum.tuwien.at/obvutwhs/download/pdf/6682266?originalFilename=true.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

Pervouchine, V. and Leedham, G. (2006). Extraction and analysis of signatures from bank cheques. International Journal on Document Analysis and Recognition, 8, 52–61.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 91–99.

Sharma, N., Raghuwanshi, M., and Chowdhury, A. S. (2018). Signature & logo detection using deep cnn for document image retrieval. 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI).

Tech4Humans (2024). Signature detection dataset. https://huggingface.co/datasets/tech4humans/signature-detection. Acessado em: 2024-07-03.

Ultralytics (2023). Ultralytics yolov8. https://github.com/ultralytics/ultralytics. Acesso em: 2024-06-10.

Wu, Y., Cao, Y., Cheng, K., Liu, W., Chen, Y., and Cai, D. (2020). Rethinking classification and localization for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10186–10195.

Yan, K., Geng, J., Su, B., Xu, J., Zhang, S., Tang, X., He, X., and Li, X. (2022). Signature detection, restoration and verification: A novel chinese document signature benchmark. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 1356–1364. Disponível em: https://openaccess.thecvf.com/content/CVPR2022W/SketchDL/papers/Yan_Signature_Detection_Restoration_and_Verification_A_Novel_Chinese_Document_Signature_CVPRW_2022_paper.pdf.

Zhai, X., Sun, S., Yu, S., Yao, J., and Wu, J. (2017). Signature detection in document images using deep convolutional neural networks. Multimedia Tools and Applications, 76(21), 22497–22515.

Zhang, H., Wang, Y., Dayoub, F., and Sünderhauf, N. (2021). Varifocalnet: An iou-aware dense object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8514–8523.

Zhang, Y., Jiang, W., Wang, Z., Yuan, Z., Luo, P., and Zhu, L. (2022a). Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*.

Zhang, Y., Li, X., and outros (2022b). Handwritten chinese signature detection on scanned technical documents via copy-paste augmentation and yolov5. International Conference on Document Analysis and Recognition (ICDAR). Disponível em: https://arxiv.org/abs/2203.12864.