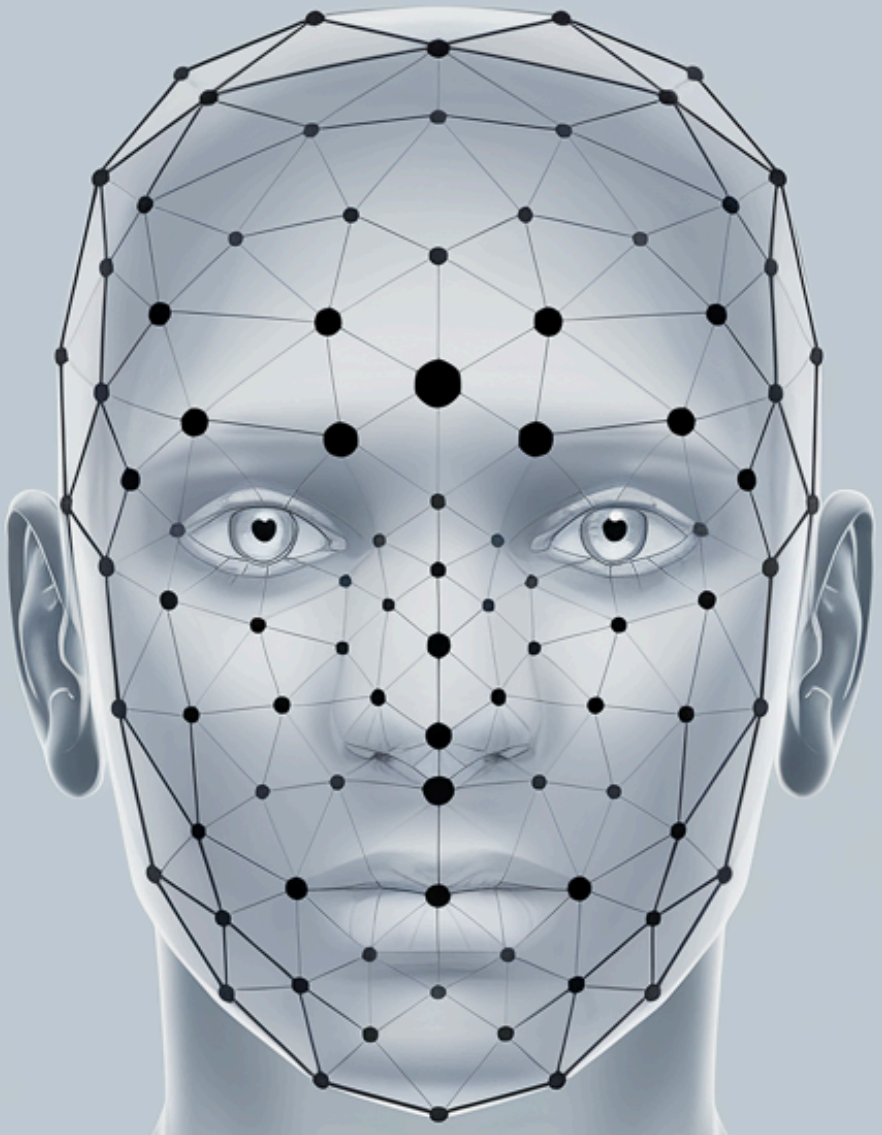
UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

Visão Computacional no Diagnóstico Dermatológico

Aplicação de Técnicas de Detecção de Objetos para Identificação de Lesões de Câncer de Pele em Imagens Médicas

Guilherme Henrique dos Reis



UFG
UNIVERSIDADE FEDERAL DE GOIÁS

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

GUILHERME HENRIQUE DOS REIS

Visão Computacional no Diagnóstico Dermatológico

Aplicação de Técnicas de Detecção de Objetos para Identificação de Lesões de Câncer de Pele em Imagens Médicas

Goiânia
2025

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

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

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

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

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

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): GUILHERME HENRIQUE DOS REIS

Título do trabalho: Visão Computacional no Diagnóstico Dermatológico

Aplicação de Técnicas de Detecção de Objetos para Identificação de Lesões de Câncer de Pele em Imagens Médicas

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

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

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

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

Documento assinado eletronicamente por **Guilherme Henrique Dos Reis**, **Discente**, em 15/01/2025, às 17:20, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Fernando Marques Federson**, **Professor do Magistério Superior**, em 16/01/2025, às 18:27, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5089575** e o código CRC **67266087**.

Referência: Processo nº 23070.001560/2025-10 SEI nº 5089575

GUILHERME HENRIQUE DOS REIS

Visão Computacional no Diagnóstico Dermatológico

Aplicação de Técnicas de Detecção de Objetos para Identificação de Lesões de Câncer de Pele em Imagens Médicas

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.
Orientador: Prof. Dr. Fernando Marques Federson

Goiânia
2025

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

REIS, GUILHERME HENRIQUE DOS
Visão Computacional no Diagnóstico Dermatológico [manuscrito] : Aplicação de Técnicas de Detecção de Objetos para Identificação de Lesões de Câncer de Pele em Imagens Médicas / GUILHERME HENRIQUE DOS REIS. - 2025.
72 f.

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

1. inteligência artificial. 2. visão computacional. 3. detecção de objetos. I. Federson, Fernando Marques , orient. II. Título.

CDU 004

GUILHERME HENRIQUE DOS REIS

Visão Computacional no Diagnóstico Dermatológico

Aplicação de Técnicas de Detecção de Objetos para Identificação de Lesões de Câncer de Pele em Imagens Médicas

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

Data da Aprovação: 17 de dezembro de 2024.

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

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

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

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

GUILHERME HENRIQUE DOS REIS

Visão Computacional no Diagnóstico Dermatológico

Aplicação de Técnicas de Detecção de Objetos para Identificação de Lesões de Câncer de Pele em Imagens Médicas

RESUMO

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

Palavras-chave: inteligência artificial, modelos grandes de linguagem, geração automática de datasets.

ABSTRACT

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

Keywords: artificial intelligence, large language models, automatic dataset generation.

Goiânia

2025

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Minha Jornada

Guilherme Henrique dos Reis
Especialista em: Visão Computacional (Detecção de Objetos)

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

MINHA JORNADA

Nome: **Guilherme Henrique dos Reis**

Especialidade: **Visão Computacional (Detecção de Objetos)**

Objetivo deste documento

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

Minha Jornada

Minha Jornada começou na **Semana 1** com o objetivo de explorar e compreender a área de Visão Computacional de forma ampla, para que eu pudesse construir uma base sólida de conhecimento. Iniciei buscando artigos científicos em plataformas como Google Acadêmico, ResearchGate e IEEE Xplore, utilizando descritores relacionados a "Computer Vision" e "Object Detection". Adotei inicialmente uma abordagem "Bottom Up", mas percebi que era insuficiente para alcançar os fundamentos e o contexto histórico desejados. Então, mudei para uma abordagem "Top Down", priorizando conceitos gerais e históricos, como apresentados no livro “Computer Vision” de George Stockman e nos artigos “A Review Paper on Computer Vision” e “Computer Vision and Image Processing: The Challenges and Opportunities for New Technologies Approach”. Essa estratégia me ajudou a perceber a inseparabilidade entre Visão Computacional e Deep Learning, o que direcionou minha pesquisa para um estudo mais específico em detecção de objetos, uma subárea que despertou meu maior interesse. Assim, organizei os materiais encontrados e planejei os

[1] Dez semanas, entre setembro de 2024 e dezembro de 2024.

próximos passos para aprofundar minha compreensão desta fascinante área. Os materiais relacionados a esta Semana podem ser encontrados no **Apêndice 1**.

Nas **Semanas 02 e 03**, foquei no aprofundamento de conceitos sobre detecção de objetos usando deep learning. Estudei o artigo "A Survey of Deep Learning-based Object Detection", que aborda métodos clássicos como R-CNN, YOLO e SSD, além de conceitos fundamentais como redes backbone e pipelines de detecção. A seguir, busquei materiais mais recentes, destacando o artigo “A review of object detection: Datasets, performance evaluation, architecture, applications and current trends”, que introduz técnicas modernas, como modelos híbridos CNN-Transformers e abordagens baseadas em transformers como DETR, além de discutir novos datasets, métricas e aplicações emergentes. A comparação entre os artigos revelou a evolução da área, marcada por avanços como eficiência computacional, detecção em tempo real e aplicações em cenários complexos. Meu próximo objetivo foi explorar arquiteturas como YOLOv8 e DETR em maior profundidade para consolidar esses conhecimentos. Os materiais relacionados a estas duas Semanas podem ser encontrados no **Apêndice 2**.

As **Semanas 04, 05 e 06** foram dedicadas ao estudo e análise das principais arquiteturas de detecção de objetos: YOLO e DETR. Na Semana 4, o foco foi entender os principais pontos dessas arquiteturas, comparando suas abordagens, treinamento, inferência e desempenho. A Semana 5 aprofundou-se nas implementações dessas arquiteturas, destacando as diferenças no desenvolvimento de cada uma e realizando testes para avaliar sua performance em diferentes cenários. Já na Semana 6, o objetivo foi explorar a evolução das arquiteturas DETR, analisando as melhorias trazidas por versões subsequentes, como a atenção deformável e a supervisão hierárquica densa, que resultaram em maior eficiência, precisão e otimização para inferência em tempo real. Os materiais relacionados a estas três Semanas podem ser encontrados no **Apêndice 3**.

Nas **Semanas 07 e 08**, estive focado em melhorar o desempenho de modelos de detecção de feridas e investigar as causas do baixo desempenho observado. Inicialmente, treinei o modelo DETR, mas obtive resultados insatisfatórios, o que levou à exploração do YOLOv8. Embora ambos os modelos apresentassem métricas similares e fracas, identifiquei

que o dataset de feridas não estava adequadamente anotado para tarefas de detecção, o que comprometeu os resultados. Na Semana 08, busquei um novo dataset com anotações apropriadas para detecção de objetos, focado em melanomas, o que resultou em melhorias significativas nas métricas de desempenho. Assim, as semanas foram cruciais para perceber a importância da qualidade e da adequação do dataset para o sucesso de modelos de IA. Os materiais relacionados a estas duas Semanas podem ser encontrados no **Apêndice 4**.

Durante as Semanas 09 e 10, estive focado em melhorar os modelos de detecção de lesões cutâneas, inicialmente testando diferentes arquiteturas como DETR, YOLOv8, YOLO11x e Florence 2, a fim de superar o baseline obtido com o YOLOv8 Large. Embora o YOLO11x tenha alcançado o melhor desempenho, percebi que a qualidade e o tamanho do dataset eram limitações significativas. Decidi, então, adaptar o dataset HAM10000, originalmente destinado à classificação, para detecção de objetos, anotando manualmente 10.000 imagens. Com o novo dataset anotado, treinei novamente o modelo YOLO11x, o que resultou em melhorias consideráveis nas métricas de mAP, precisão e recall, apesar de desafios relacionados ao desequilíbrio de classes e limitações computacionais. A Semana 10 foi crucial, pois a transição para um dataset mais robusto e diversificado permitiu avanços significativos no desempenho do modelo, sinalizando um grande potencial para a detecção de doenças de pele. Os materiais relacionados a estas duas Semanas podem ser encontrados no **Apêndice 5**.

Em função de tudo que vivi nesta Jornada, gostaria de deixar registrado que a experiência foi extremamente enriquecedora e reveladora. Ao longo do processo, pude observar de perto como cada etapa, desde a compreensão dos fundamentos até a aplicação prática, exige uma abordagem cuidadosa e adaptativa. A escolha de estratégias como a abordagem "Top Down" foi fundamental para construir uma base sólida, e a transição para o estudo mais focado em detecção de objetos permitiu que eu encontrasse um nicho de grande interesse e relevância. A constante análise e ajuste das arquiteturas, bem como a atenção à qualidade dos dados, demonstraram como fatores como o dataset e a implementação adequada podem influenciar diretamente os resultados.

Além disso, a jornada me proporcionou uma compreensão mais profunda da importância da pesquisa contínua e do aprendizado prático. Superar desafios como a adaptação de datasets e a otimização de modelos trouxe uma sensação de progresso, e me fez perceber como a Visão Computacional e o Deep Learning têm o poder de transformar áreas como a medicina, com um impacto real na detecção de doenças.

Este processo não apenas ampliou meu conhecimento técnico, mas também reforçou a importância da paciência, da experimentação e da resiliência diante das dificuldades.

APÊNDICE 1

Termo de Aceite de Entrega

Objetivo deste documento

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

Data da Reunião ("gate") de aprovação: 19 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Guilherme Henrique dos Reis

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

Descrição do Stage: Material completo neste link.

- **Objetivo:** Obter uma compreensão ampla da visão computacional, explorando desde os fundamentos até técnicas avançadas com deep learning.
- **Abordagens:**
  - **Início: Bottom Up -** Pesquisa focada em técnicas e algoritmos específicos de detecção de objetos, que se mostrou limitada.
  - **Final: Top Down -** Mudança para uma visão geral da visão computacional, estudando fundamentos, história e técnicas principais.
- **Materiais Revisados:**
  - **Livro:** "Computer Vision" de George Stockman - Fundamentos e conceitos básicos.
  - **Artigos:**
    - "A Review Paper on Computer Vision" - Principais aplicações e avanços na área.
    - "Computer Vision and Image Processing" - Desafios e técnicas recentes.
- **Transição para Visão Computacional com Deep Learning (descida de nível):**
  - **Artigo:** "Deep learning in computer vision" - Técnicas de deep learning em visão computacional.
- **Nível Final:** Detecção de objetos dentro do contexto de deep learning (contexto de visão computacional), aprofundando a pesquisa em técnicas, algoritmos, fundamentos e história específicos para essa área.

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

Estudo Detalhado do Nível Atual: Análise dos materiais coletados sobre detecção de objetos para uma compreensão mais aprofundada sobre a área. O objetivo será entender a história da área, seus fundamentos, conceitos chave e suas principais possibilidades de aplicação.

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Durante o primeiro estágio da minha jornada para me tornar um especialista, o foco foi obter uma compreensão mais ampla da área que escolhi explorar. Inicialmente, dediquei-me a identificar fontes confiáveis e de alta qualidade para encontrar artigos científicos que pudessem me auxiliar nesse processo.

Utilizei plataformas como Google Acadêmico, ResearchGate, IEEE Xplore, entre outras, para buscar artigos com base em conceitos-chave como "Computer Vision", "Object Detection", "Paper Review", "Fundamentals", entre outros descritores relevantes.

Adotei inicialmente uma abordagem "Bottom Up", buscando artigos que tratassem diretamente da detecção de objetos, suas técnicas e algoritmos relacionados. No entanto, essa abordagem se mostrou frustrante, pois a maioria dos artigos que encontrei eram voltados para aplicações práticas e específicas, deixando de lado os fundamentos e o contexto histórico que eu buscava. Percebi que, se continuasse nessa linha, acabaria limitado a estudar técnicas e algoritmos isolados, sem uma compreensão mais profunda do campo como um todo, o que não era meu objetivo naquele momento.

Diante disso, decidi ajustar minha estratégia e adotar uma abordagem "Top Down". Dessa vez, meu foco foi entender a visão computacional de forma mais ampla, a fim de explorar as técnicas, conceitos, fundamentos e a história dessa área de maneira mais completa. Retornei às mesmas plataformas de pesquisa, mas agora busquei artigos com um escopo mais macro, abrangendo o campo como um todo.

Comecei a selecionar artigos com base em descritores semelhantes, mas com foco em palavras-chave que indicassem uma visão mais geral, como "História da Visão Computacional" ou "Conceitos de Visão Computacional". Dessa forma, compilei uma lista inicial com artigos que me pareceram relevantes, os quais organizei na pasta "Stage01-Fundamentals". Posteriormente, também adicionei artigos mais específicos nessa pasta, após “descer os níveis”.

Com esse primeiro conjunto de artigos em mãos, passei a analisá-los para determinar sua utilidade. Para isso, comecei lendo apenas a introdução de cada um, o que me permitiu identificar os mais interessantes e aqueles que chamavam mais a minha atenção para continuar aprofundando os estudos.

Os materiais que decidi analisar mais profundamente foram os artigos “A Review Paper on Computer Vision” e “Computer Vision and Image Processing: the Challenges and

Opportunities for new technologies approach" e o livro "Computer Vision" de George Stockman.

Comecei seguindo uma ordem cronológica e li primeiramente o livro, datado do ano 2000. No entanto, após concluir a leitura da introdução, decidi passar para o próximo material. Embora o livro oferecesse uma sólida fundamentação e conceitos valiosos, percebi que ele não seria tão útil para aprofundar o conhecimento no nível que eu precisava.

De forma resumida, o livro apresenta a visão computacional como um campo da ciência da computação que busca desenvolver algoritmos e sistemas capazes de interpretar imagens digitais, replicando aspectos da visão humana. Os autores discutem a decomposição de imagens em representações computáveis, como a extração de características, segmentação e classificação de objetos, bem como a reconstrução 3D a partir de dados 2D. Eles destacam a relação do campo com áreas como processamento de sinais, geometria, inteligência artificial e aprendizado de máquina, além de enfatizar os desafios de traduzir informações visuais em descrições simbólicas para tomada de decisão automática. A introdução apresenta técnicas de filtragem de imagens, detecção de bordas, transformações geométricas e extração de características.

Após isso, realizei a leitura detalhada dos 2 artigos que achei mais interessantes. E resumi abaixo seus conteúdos para facilitar o entendimento.

Artigo 01: A Review Paper on Computer vision:

O artigo revisa os avanços em visão computacional, pontuando que se trata de uma área interdisciplinar que combina processamento de imagens, reconhecimento de padrões e aprendizado de máquina para interpretar e analisar dados visuais, como fotos e vídeos. As principais aplicações citadas no artigo incluem detecção de objetos, segmentação, reconhecimento facial, veículos autônomos, realidade aumentada/virtual, robótica, monitoramento agrícola, análise esportiva e diagnóstico médico. A visão computacional também desempenha um papel crucial em inteligência artificial, otimização de processos em e-commerce e serviços bancários, além de ter um futuro promissor em áreas como visão 3D, processamento em tempo real e computação de borda.

Artigo 02: Computer Vision and Image Processing: the Challenges and Opportunities for new technologies approach

O artigo revisa os desafios e oportunidades nas áreas de visão computacional e processamento de imagens, destacando técnicas e avanços recentes. Ele explora a aplicação de processamento digital de imagens em várias áreas, incluindo reconhecimento de padrões e inteligência artificial, e examina como esses métodos são utilizados para

análise de imagens e extração de dados significativos. As técnicas discutidas incluem algoritmos de segmentação de imagens, detecção de bordas, e identificação de objetos usando redes neurais convolucionais (CNN). A segmentação de imagens é usada para dividir imagens em regiões distintas para facilitar a análise, enquanto a detecção de bordas é empregada para identificar contornos de objetos com base em variações de pixels. O artigo também aborda o uso de algoritmos genéticos, redes neurais artificiais e lógica fuzzy para melhorar a precisão e eficácia dos sistemas de visão computacional.

Após revisar os materiais e introduções dos artigos relacionados, constatei que, atualmente, separar visão computacional de deep learning não faz mais sentido. Por isso, decidi “descer um nível”, passando de uma abordagem geral de visão computacional para uma abordagem específica de visão computacional com deep learning.

Dito isso, perceber que a integração entre visão computacional e deep learning era essencial para um entendimento mais abrangente da área, me fez focar minha pesquisa em artigos e materiais que explorassem especificamente a visão computacional dentro do contexto do deep learning. Para essa nova etapa, adotei um processo, novamente, sistemático de busca e seleção de artigos que me permitisse aprofundar meus conhecimentos e entender as últimas tendências e avanços e, novamente, salvei esses artigos na pasta “Stage01-Fundamentals”.

A análise dos “novos” artigos selecionados revelou a vasta gama de técnicas e abordagens disponíveis na área de visão computacional com deep learning. Nessa etapa pude perceber que mesmo estando em “um nível abaixo” do contexto geral ainda havia uma infinidade de áreas de aplicação, técnicas, algoritmos, frameworks e conteúdos científicos que sustentam a nova área.

Novamente li a introdução de todos os artigos e resumi os artigos lidos (aqueles que achei mais interessantes) para facilitar o entendimento:

Artigo 03: Deep learning in computer vision: A critical review of emerging techniques and application scenarios:

Este artigo revisa criticamente os avanços recentes em deep learning (DL) aplicados à visão computacional (CV), destacando oito técnicas emergentes: AlexNet, VGGNet, GoogLeNet & Inception, ResNet, DenseNet, MobileNets, EfficientNet e RegNet. Ele investiga suas origens, atualizações e aplicações em quatro cenários principais: reconhecimento de imagens, rastreamento visual, segmentação semântica e restauração de imagens. O artigo divide o desenvolvimento da área em três estágios (2012-2016, 2016-2019, e de 2019 em diante) e identifica tendências futuras tanto no lado técnico quanto nas aplicações, oferecendo insights valiosos para pesquisadores e profissionais da indústria de CV.

Cada artigo estudado trouxe contribuições valiosas que ajudaram a aprofundar meu entendimento sobre como os modelos de deep learning podem ser aplicados para resolver problemas de visão computacional em cenários variados. Entretanto, ainda era muita coisa e eu percebi que era necessário “descer o nível” mais uma vez.

Para afunilar mais e reduzir o escopo, fiz um levantamento de alguns dos conceitos mais citados em todos artigos que estudei, sendo eles:

- **Processamento de imagens**
- **Extração de características**
- **Segmentação Semântica**
- **Detecção de Objetos**
- **Reconhecimento de padrões**
- **Classificação de imagens**

E dentre os vários conceitos que me deparei, Detecção de objetos continua sendo o que mais mexe com o meu coração. Dessa forma, decidi descer mais um nível, passando de visão computacional para visão computacional com deep learning, e daí passando para detecção de objetos.

Mais uma vez pesquisei materiais ajustando minha busca de acordo com o nível em que eu estava, e, mais uma vez, coloquei os resultados dessa busca na pasta “Stage01-Fundamentals”.

O meu próximo passo será estudar esses materiais que encontrei para compreender, de fato, a área que escolhi, pois, por mais que já tenha descido dois níveis, ainda é uma área extensa e com muitos materiais e técnicas próprias.

APÊNDICE 2

Termo de Aceite de Entrega

Objetivo deste documento

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

Data da Reunião ("gate") de aprovação: 26 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Guilherme Henrique dos Reis

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

Descrição do Stage: Material completo neste link.

- **Objetivo:** Estudar os fundamentos e história da Detecção de objetos usando Deep Learning.
- **Abordagem:**
  - Ler os artigos selecionados anteriormente sobre Detecção de objetos com Deep Learning
- **Materiais Revisados:**
  - **Artigos:**
    - "A survey of Deep Learning-based Object Detection" - fundamentos e história do campo de pesquisa.
- **Final:** Necessidade de encontrar materiais mais recentes pois grande parte do material selecionado pode estar obsoleto e não discutir técnicas e avanços recentes.

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

Estudar profundamente os artigos selecionados que foram publicados nos últimos meses para obter uma compreensão sobre o estado da arte em Detecção de Objetos com Deep Learning.

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

O objetivo desse stage era estudar mais profundamente os artigos encontrados sobre detecção de objetos, com foco no uso de técnicas de Deep Learning. Para isso, analisei os materiais de forma parecida ao que havia feito anteriormente, ou seja, comecei lendo apenas a introdução de cada artigo, o que me permitiu identificar os mais interessantes e aqueles que chamavam mais a minha atenção para continuar aprofundando os estudos.

O primeiro artigo que decidi estudar foi o artigo "A Survey of Deep Learning-based Object Detection" que, de forma resumida, faz uma revisão detalhada dos principais métodos de detecção de objetos baseados em Deep Learning, abordando suas características, arquiteturas e aplicações.

A introdução do artigo destaca a importância da detecção de objetos em várias áreas, como segurança, direção autônoma e análise de imagens de drones. A evolução das Redes Neurais Convolucionais (CNNs) e o aumento da capacidade de computação são fatores críticos para o avanço nesta área. A tarefa de detecção envolve a localização de instâncias de objetos semânticos de uma determinada classe (ex.: humanos, carros) em imagens e vídeos. Os métodos de detecção são divididos em detectores de uma e duas fases.

O artigo também traz o conceito de "backbone", que refere-se à rede de extração de características usada como base para detecção de objetos. Redes populares incluem ResNet, MobileNet e ShuffleNet. A escolha da rede backbone depende da necessidade de balanço entre precisão e eficiência. Redes mais profundas como ResNet garantem alta capacidade de detecção, enquanto redes leves como MobileNet são usadas para aplicações em dispositivos móveis.

A seção subsequente do material apresenta as principais arquiteturas de detecção de objetos baseadas em deep learning R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD e RetinaNet.

Também é citado alguns datasets e benchmarks para essa tarefa como PASCAL VOC, MS COCO, ImageNet e VisDrone.

O artigo também destaca quatro etapas principais de um pipeline típico de detecção: pré-processamento, extração de características, classificação/localização e pós-processamento.

Além disso, é destacado que a detecção de objetos tem inúmeras aplicações práticas, como em veículos autônomos, vigilância, e até em diagnósticos médicos. As ramificações incluem a detecção de textos em cena, detecção de pedestres e segmentação de instâncias, como no caso do Mask R-CNN, que combina detecção de objetos com segmentação.

O artigo estudado forneceu uma base sólida de fundamentos e conceitos chave da área que estavam presentes na maioria dos materiais selecionados. Entretanto, esse material foi publicado em 2019 e não está mais compatível com o estado da arte.

Dessa forma, fui atrás de encontrar algum material mais recente que pudesse conter técnicas e conceitos modernos da área. Encontrei o artigo “Object Detection in 20 years: a survey” que traz uma revisão mais recente que cita conceitos até então mal mencionados, como Transformers.

Assim, o meu próximo passo será estudar esse artigo profundamente, e outros materiais recentes, para entender o cenário atual da detecção de objetos usando deep learning.

Termo de Aceite de Entrega

Objetivo deste documento

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

Data da Reunião ("gate") de aprovação: 3 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

GUILHERME HENRIQUE DOS REIS

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

Descrição do Stage: Material completo neste link.

- **Objetivo:** Estudar materiais recentes sobre Detecção de Objetos Usando Deep Learning
- **Abordagem:**
  - Ler os artigos selecionados que foram publicados no último ano sobre Detecção de objetos com Deep Learning
- **Materiais Revisados:**
  - **Artigo:**
    - "A review of object detection: Datasets, performance evaluation, architecture, applications and current trends" - novidades na área de Detecção de Objetos

Final: Conclui o estudo de um artigo de revisão recente na área e o comparei com o conhecimento adquirido durante o Stage anterior, destacando as inovações introduzidas recentemente.

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

Estudar as arquiteturas de Detecção de Objetos YOLO (You Only Look Once) e DETR (Detection Transformers).

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

O objetivo deste Stage foi realizar uma busca por materiais atualizados sobre detecção de objetos utilizando deep learning, já que os recursos encontrados até o momento, datados de 2019, poderiam não contemplar as técnicas mais recentes. Para isso, utilizei plataformas como Google Acadêmico e IEEE Xplore, entre outras, aplicando filtros de data para selecionar publicações dos últimos 12 meses. A pesquisa foi realizada com o termo "Object Detection Review".

Após filtrar os materiais encontrados, decidi ler o artigo intitulado “A review of object detection: Datasets, performance evaluation, architecture, applications and current trends” de Wei Chen e colaboradores, publicado em janeiro de 2024.

Esse artigo revisa a evolução da detecção de objetos, destacando a transição dos métodos tradicionais para os baseados em aprendizado profundo, como as redes neurais convolucionais (CNNs), que trouxeram avanços significativos em precisão e eficiência. Enquanto os métodos tradicionais dependiam da extração manual de características (como HOG e SIFT), as CNNs permitem a extração automática de características de alto nível.

O estudo classifica os métodos modernos de detecção de objetos em três categorias: baseados em âncoras (Anchor-based), sem âncoras (Anchor-free) e baseados em transformadores (Transformer-based). Além de discutir a estrutura, vantagens e desvantagens desses métodos, o artigo também explora os principais conjuntos de dados (como PASCAL VOC, MS COCO e ImageNet) e as métricas de avaliação (como mAP e IoU).

As aplicações da detecção de objetos abrangem diversas áreas, como transporte, medicina e vigilância. O artigo também aponta tendências futuras na pesquisa, destacando o uso crescente de transformadores e os desafios de melhorar a detecção de objetos pequenos e em cenários complexos.

O que mais se destacou para mim foi a relevância das adições que o artigo trouxe em relação aos materiais estudados no Stage 02. Embora mantenha toda a base teórica dos artigos anteriores, este trabalho introduz novas técnicas e algoritmos, como os baseados em transformers, além de também apresentar datasets mais recentes.

Os artigos "*A review of object detection: Datasets, performance evaluation, architecture, applications and current trends*" de Wei Chen (estudado no Stage03) e "*A Survey of Deep Learning-based Object Detection*" de Licheng Jiao (estudado no Stage02) têm abordagens e focos diferentes em relação à detecção de objetos, então decidi comparar os dois para identificar as principais diferenças entre eles.

Diferenças entre os materiais:

Datasets e Métricas de Avaliação de Desempenho:

Wei Chen: O artigo de Wei Chen destaca-se por dar uma ênfase detalhada aos datasets utilizados na detecção de objetos e métodos de avaliação de desempenho. Ele traz uma revisão abrangente das bases de dados mais atualizadas, como *COCO*, *VOC*, *Open Images*, entre outras, além de discutir as métricas mais relevantes para a avaliação do desempenho dos algoritmos (AP, mAP, AR, F1-score etc.).

- Novidades: Atualização dos principais datasets (adição dos datasets Open Images, Waymo Open Dataset, Berkley Deep Drive, VisDrone, ArgoVerse, entre outros) utilizados na detecção de objetos e discussão detalhada de desafios como o desequilíbrio de classes, complexidade de cenas e resolução de imagens.

Licheng Jiao: O artigo de Jiao também revisa os datasets, mas o foco é menos detalhado nesse aspecto. Ele se concentra mais em métodos baseados em aprendizado profundo e seus impactos na detecção de objetos, mencionando apenas os datasets mais amplamente usados (MS COCO, PASCAL VOC, entre outros) sem entrar em muitos detalhes sobre como eles afetam a avaliação do desempenho.

Arquiteturas:

Wei Chen: Visão atualizada das arquiteturas usadas para detecção de objetos, incluindo as mais recentes como EfficientDet, YOLOv8 (embora hoje já esteja na v11, lançada dia 30/09/24) e versões mais otimizadas da família RetinaNet. Ele também fala sobre as abordagens híbridas que combinam redes neurais convolucionais (CNNs) com Vision Transformers (ViT) e suas variações, como o DETR (Detection Transformer).

- Novidades: O uso de modelos híbridos CNN e Transformers, além de melhorias em eficiência computacional e menor consumo de memória nas novas arquiteturas, são aspectos abordados de maneira inédita.

Licheng Jiao: O foco maior é em arquiteturas baseadas em CNNs, como Faster R-CNN, SSD, YOLO (primeiras versões) e RetinaNet, visto que na época os Transformers ainda não tinham ganhado grande popularidade no campo da visão computacional.

Aplicações:

Wei Chen: Aplicações mais recentes e emergentes da detecção de objetos, como veículos autônomos, drones, cidades inteligentes, monitoramento ambiental, e novas aplicações em saúde, como na radiologia com detecção de anomalias. Ele também aborda o impacto da detecção de objetos em tempo real, discutindo tecnologias para otimização de inferência em dispositivos móveis e embarcados.

- Novidades: Aplicações emergentes como detecção em dispositivos IoT, cidades inteligentes e avanços na área médica, com ênfase em IA para saúde.

Licheng Jiao: Também discute várias aplicações como segurança (CCTV, vigilância), veículos autônomos e análise de imagens aéreas. Ele não aborda em detalhes os dispositivos móveis e embarcados.

Tendências Atuais e Desafios Futuros:

Wei Chen: O artigo traz uma seção sobre as tendências atuais e desafios futuros na detecção de objetos, como a necessidade de algoritmos mais robustos contra ataques adversários, a tendência de edge computing, o aumento da eficiência energética, e os desafios éticos e legais, como privacidade e viés algorítmico.

- Novidades: Discussão atualizada sobre a integração de edge computing, detecção de objetos em ambientes adversos e éticos no uso de IA.

Licheng Jiao: O artigo de Jiao menciona alguns desafios, mas foca mais nas limitações técnicas da época, como a necessidade de melhor generalização dos modelos e aumento da eficiência computacional, sem entrar em detalhes sobre ética ou computação em borda.

Técnicas de Pós-Processamento e Modelos de Generalização:

Wei Chen: Aborda o uso de técnicas avançadas de pós-processamento, como Non-Maximum Suppression otimizado e abordagens baseadas em aprendizado para refinamento de detecção. Também destaca como novos métodos de generalização estão surgindo para melhorar o desempenho em cenários de few-shot e zero-shot learning.

- Novidades: Uso de aprendizado com poucos dados (few-shot) e métodos de refinamento automatizados como parte essencial do pipeline de detecção.

Licheng Jiao: Menos foco em técnicas de pós-processamento avançadas e pouca menção a few-shot e zero-shot learning, pois esses conceitos ainda estavam em seus estágios iniciais.

Com base na comparação entre os artigos de Wei Chen e Licheng Jiao, ficou claro que a evolução das arquiteturas de detecção de objetos, particularmente com a introdução de técnicas baseadas em transformers e a combinação com redes convolucionais, trouxe avanços significativos em precisão, eficiência e adaptabilidade a cenários complexos. As inovações mais recentes, como as arquiteturas híbridas que combinam CNNs com Vision Transformers (ViT) e o uso de modelos como YOLOv8 (e as versoes mais atuais) e DETR, destacam-se como marcos importantes para superar desafios como a detecção de objetos pequenos e o desempenho em tempo real. Dessa forma, o próximo passo natural na jornada de estudos será explorar mais detalhadamente as arquiteturas YOLO e DETR, entendendo suas estruturas, pontos fortes e suas aplicações em cenários reais, com o objetivo de aprofundar o conhecimento sobre as tendências mais recentes da detecção de objetos.

APÊNDICE 3

Termo de Aceite de Entrega

Objetivo deste documento

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

Data da Reunião ("gate") de aprovação: 9 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

GUILHERME HENRIQUE DOS REIS

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

Descrição do Stage: Material completo neste link.

- **Objetivo:** Estudar as arquiteturas de Detecção de Objetos YOLO e DETR.
- **Abordagem:**
  - Estudar os artigos das arquiteturas.
  - Assistir videos pertinentes da playlist Modern Object Detection: from YOLO to transformers - YouTube
- **Materiais Revisados:**
  - **Artigos:**
    - "End-to-End Object Detection with Transformers" - Proposta e fundamentação do DETR
    - "You Only Look Once: Unified, Real-Time Object Detection" - Proposta e fundamentação do YOLO

Final: Levantamento de pontos chaves de cada arquitetura.

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

Estudar as implementações do YOLO e DETR

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

O objetivo deste Stage foi estudar as arquiteturas de detecção de objetos YOLO e DETR, para compreender os algoritmos que mais são usados atualmente para essa tarefa. Para isso foi estudado os artigos “End-to-End Object Detection with Transformers” e “You Only Look Once: Unified, Real-Time Object Detection”. Para deixar o resultado do estudo mais palpavel, elenquei os principais pontos dessas arquiteturas nesse material.

YOLO (You Only Look Once):

Sistema de detecção de objetos em tempo real que enquadra a tarefa de detecção como um único problema de regressão, dos pixels da imagem às coordenadas das bounding boxes e probabilidades das classes. Detecta múltiplos objetos em uma imagem, prevendo bboxes e probabilidades de classe em uma única passagem direta pela rede.

DETR (Detection Transformers):

Propõe um framework de detecção de objetos de ponta a ponta usando Transformers, eliminando a necessidade de componentes tradicionais como propostas de região e pós-processamento (non maximum supression). Realiza a detecção de objetos e a previsão de bboxes usando uma arquitetura encoder-decoder baseada em Transformers.

Comparação entre YOLO e DETR

Arquitetura

YOLO:

Usa uma rede neural convolucional (CNN) customizada baseada na arquitetura GoogLeNet. O artigo original da YOLO utiliza 24 camadas convolucionais seguidas de 2 camadas fully conected. A YOLO divide a imagem em uma grade (S*S), e cada célula da grade prevê bounding boxes e confiança para objetos cujo centro cai dentro dessa célula. A YOLO realiza a classificação de objetos e a localização simultaneamente por meio de uma única CNN, unificando a detecção em um único framework. Cada célula da grade prevê bboxes, confiança de objeto e probabilidades de classe. Combina a loss de localização (regressão da bbox) com a loss de classificação.

DETR:

Usa uma CNN padrão (ResNet no artigo original) para extrair características visuais da imagem de entrada. Usa um modelo transformer encoder-decoder onde o encoder processa as características da imagem, e o decoder usa consultas de

objetos aprendidas para prever bounding boxes e classes. O DETR é projetado como uma abordagem fim-a-fim, sem necessidade de âncoras pré-definidas, propostas de região ou pós-processamento. Os transformers lidam tanto com a detecção de objetos quanto com a regressão das caixas delimitadoras. Leva vantagem do mecanismo de atenção dos Transformers para relacionar o contexto global entre características da imagem e consultas de objetos.

Treinamento

YOLO:

Inicialmente pré-treinado no conjunto de dados de classificação ImageNet para extração geral de características, e depois é ajustado em conjuntos de dados de detecção de objetos. Treinado em conjuntos de dados de detecção de objetos como PASCAL VOC e MS COCO. O YOLO pode lidar com várias classes de objetos e caixas delimitadoras. O YOLO não usa caixas de âncoras em sua versão original. Em vez disso, cada célula da grade prevê diretamente um número fixo de caixas delimitadoras. A função de loss usada combina várias tarefas: loss de localização (para caixas delimitadoras), loss de confiança (para existência de objeto) e loss de classificação.

DETR:

Usa um backbone CNN (ResNet pré-treinada no ImageNet) para extração de características, mas a porção transformer é treinada em conjuntos de dados de detecção de objetos. O artigo original avalia no MS COCO, mas o DETR requer conjuntos de dados maiores devido à sua dependência pesada do modelo transformer. Usa a loss Húngara para casar previsões com caixas delimitadoras de verdadeiros positivos, combinando loss de classificação (para rótulos de objetos), loss L1 (para caixas delimitadoras) e loss de IoU (Intersection over Union) generalizada.

Tempo de Inferência e Complexidade

YOLO:

O YOLO é conhecido por seu tempo de inferência rápido. Ele alcança detecção em tempo real a cerca de 45 frames por segundo (FPS) na implementação original. A única passagem direta para a detecção de objetos torna o YOLO computacionalmente eficiente e ele escala bem para aplicações em tempo real em

GPUs. O foco é na velocidade, trocando um pouco de precisão por um FPS elevado, especialmente em versões menores como o YOLO-nano.

DETR:

O DETR tem um tempo de inferência significativamente mais lento comparado ao YOLO. Devido ao mecanismo de transformer, o DETR é muito mais intensivo computacionalmente, especialmente na fase do decoder, que processa as consultas de objetos iterativamente. A complexidade computacional do DETR cresce de forma quadrática com o tamanho da imagem devido ao mecanismo de atenção nos Transformers, resultando em tempos de inferência mais lentos comparado a arquiteturas baseadas em convolução, como o YOLO. Sacrifica velocidade de inferência por uma detecção de maior qualidade, especialmente em cenários de detecção desafiadores.

Complexidade Matemática

YOLO:

A matemática do YOLO é relativamente simples. Ele define a tarefa de detecção de objetos como um problema de regressão ao dividir a imagem em uma grade. Para cada célula da grade, prevê caixas delimitadoras e pontuações de confiança associadas. O YOLO prevê diretamente as coordenadas da caixa delimitadora como \((x, y, w, h) \), juntamente com a confiança do objeto. O YOLO combina uma função de perda multi-parte: perda de localização (diferença quadrada para coordenadas de caixas delimitadoras), perda de confiança (entropia cruzada binária para a presença de objeto) e perda de classificação (entropia cruzada softmax para probabilidades de classe). A arquitetura de única CNN reduz a complexidade por não depender de redes de propostas de região, âncoras ou processamento em múltiplos estágios.

DETR:

O DETR é matematicamente complexo, dependendo fortemente do mecanismo de atenção dos Transformers. O mecanismo de auto-atenção permite que cada elemento se relacione com todos os outros globalmente, exigindo múltiplas multiplicações de matrizes por camada. O DETR usa o algoritmo de correspondência Húngaro para casar as caixas delimitadoras previstas com as verdadeiras. Isso requer a solução de um problema de otimização combinatória para minimizar a perda de correspondência entre previsões e verdadeiros positivos. As camadas transformer envolvem atenção multi-cabeça, que calcula pontuações de atenção entre cada par de tokens de entrada, aumentando o custo e a complexidade computacional. O

DETR usa uma combinação de perda L1 para regressão de caixas delimitadoras, perda de IoU generalizada e perda de entropia cruzada para classificação, o que é mais complexo do que a função de perda do YOLO.

Pós-processamento

YOLO:

O YOLO usa supressão não-máxima (NMS) para remover previsões redundantes de caixas delimitadoras, uma técnica comum em detectores de objetos. Isso é necessário para filtrar caixas sobrepostas. O YOLO original não usa âncoras. Versões posteriores (YOLOv2 e além) introduziram caixas de âncora para melhorar a precisão, mas a primeira versão prevê caixas delimitadoras diretamente.

DETR:

O DETR não requer supressão não-máxima (NMS) porque sua arquitetura baseada em transformer gera um número fixo de previsões, e o mecanismo de atenção garante menos caixas duplicadas. Isso reduz a complexidade do pós-processamento em comparação com o YOLO. O DETR prevê um conjunto fixo de consultas de objetos que são correspondidos com os verdadeiros positivos via o algoritmo Húngaro, eliminando a necessidade de propostas de região ou caixas de âncora.

Desempenho e Precisão

YOLO:

O YOLO troca parte da precisão pela velocidade. Embora seja rápido, muitas vezes tem dificuldades com objetos menores ou objetos em cenas densas devido à estrutura de grade grosseira. A estrutura de grade pode limitar seu desempenho, já que cada célula da grade é responsável por detectar apenas um número limitado de objetos, o que pode levar a erros em cenas densas.

DETR:

O DETR atinge maior precisão, especialmente em cenas complexas, e consegue lidar melhor com a detecção de objetos de tamanhos variados, devido ao contexto global fornecido pelo Transformer. Ele tem maior precisão e melhor localização de objetos, particularmente em objetos complexos ou sobrepostos. No entanto, o DETR requer tempos de treinamento significativamente mais longos e

conjuntos de dados maiores para convergir, além de ser mais lento na inferência comparado ao YOLO.

Resumo

- **YOLO**:
  - Otimizado para velocidade e eficiência, tornando-o adequado para aplicações em tempo real com precisão moderada. É mais eficiente computacionalmente, o que o torna viável para implantação em dispositivos com recursos computacionais limitados.

- **DETR**:
  - Focado na precisão, especialmente para cenas complexas e densas. Seu design fim-a-fim e dependência da arquitetura Transformer eliminam muitos componentes tradicionais, mas ao custo de inferência mais lenta e maior demanda computacional.

Termo de Aceite de Entrega

Objetivo deste documento

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

Data da Reunião ("gate") de aprovação: 16 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

GUILHERME HENRIQUE DOS REIS

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

Objetivo: material neste link
Estudar as implementações das arquiteturas de detecção de objetos YOLO e DETR.

Abordagem:

- Pesquisar as organizações responsáveis pelas arquiteturas originais de ambos os algoritmos.
- Analisar os artigos e fundamentos teóricos de cada um.
- Assistir vídeos da playlist *Modern Object Detection: from YOLO to transformers* no YouTube para complementar o entendimento.

Materiais Revisados:

- **Artigos:**
  - "End-to-End Object Detection with Transformers" - Proposta e fundamentação da arquitetura DETR.
  - "You Only Look Once: Unified, Real-Time Object Detection" - Proposta e fundamentação da arquitetura YOLO.
- **Testes com os algoritmos:**
  - Utilização de modelos pré-treinados do DETR e YOLO11 para detecção em uma imagem arbitrária.
  - O DETR identificou um objeto como *sheep* com 1.00 de confiança, enquanto o YOLO identificou como *sheep* com 0.95 de confiança, mas ambos falharam na classificação correta.
  - Após ajuste fino com um dataset de animais (1.000 imagens), o DETR identificou *goat* com 0.76 de confiança, enquanto o YOLO11 classificou como *racoon* com 0.78 de confiança.

Resultados:

- **DETR:** Conseguiu identificar a classe correta (*goat*) com 0.76 de confiança após o ajuste fino.
- **YOLO11:** Identificou incorretamente a classe (*racoon*) com 0.78 de confiança, mesmo após o ajuste.

Conclusão:
O DETR demonstrou um desempenho superior na detecção correta da classe, capturando melhor o contexto global da imagem, enquanto o YOLO11 falhou na classificação.

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

Estudar variantes do DETR (CoDETR, RTDETR, DINO)

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

O objetivo deste Stage foi estudar as implementações das arquiteturas YOLO e DETR. O primeiro passo foi identificar as organizações responsáveis pelo desenvolvimento das arquiteturas originais do DETR (DEtection TRansformers) e do YOLO (You Only Look Once). O DETR foi proposto pela equipe de pesquisa da Facebook AI Research em 2020. Esta arquitetura inovadora utiliza transformers, uma estrutura amplamente utilizada em processamento de linguagem natural, para realizar a tarefa de detecção de objetos. Diferente de modelos convencionais, como o YOLO, que utilizam uma abordagem baseada em redes convolucionais, o DETR aproveita a atenção dos transformers para identificar e localizar objetos em imagens com mais precisão, mesmo em cenários mais complexos e com sobreposição de objetos.

Já o YOLO, uma das arquiteturas mais populares em detecção de objetos, foi inicialmente desenvolvido por Joseph Redmon e colaboradores na Universidade de Washington em 2016. O modelo foi revolucionário ao propor uma abordagem de detecção em tempo real, onde a imagem é dividida em regiões e os objetos são detectados simultaneamente, em vez de utilizar uma abordagem em várias etapas como outros algoritmos tradicionais. Ao longo dos anos, o YOLO passou por várias iterações e melhorias, com a quinta versão (YOLOv5) sendo uma das mais conhecidas e amplamente adotadas. A organização Ultralytics foi responsável pelo desenvolvimento e popularização de implementações da YOLO, que foram altamente otimizadas para desempenho e facilidade de uso em projetos de visão computacional.

Ao estudar as implementações desses algoritmos, foi importante compreender as principais diferenças entre suas abordagens. O DETR, ao utilizar transformers, destaca-se por seu potencial em capturar relações globais entre objetos e seu contexto na imagem. Em contrapartida, o YOLO mantém uma abordagem baseada em redes convolucionais que permite processar imagens de forma rápida e eficiente, sendo muito utilizado em aplicações de tempo real, como drones e veículos autônomos. Cada um desses algoritmos tem suas

vantagens e desvantagens dependendo do tipo de tarefa e dos requisitos de desempenho, o que torna essencial entender o contexto no qual serão implementados.

Testes com os algoritmos:

Primeiramente decidi validar como os algoritmos iriam performar em imagens arbitrárias. Para isso, utilizei os modelos pré-treinados do DETR e YOLO11, realizando a detecção em uma imagem selecionada.

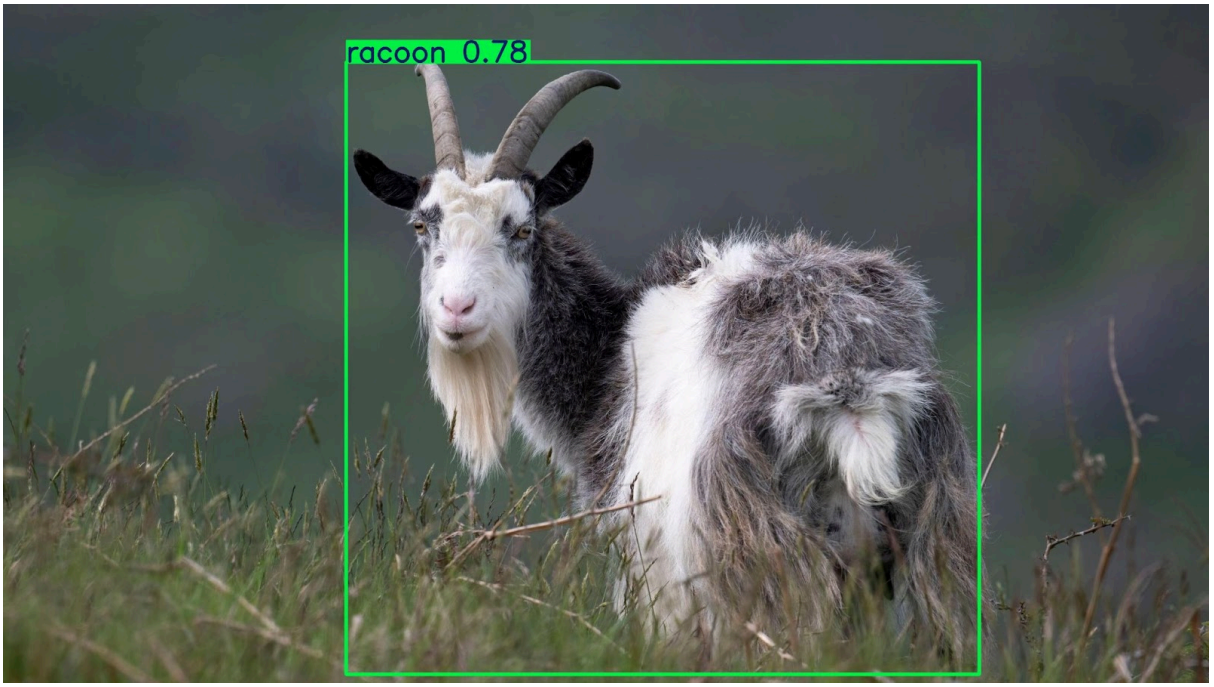
Detecção realizada com o DETR (classe sheep 1.00)

Detecção realizada com o YOLO11 (classe sheep 0.95)

Com esse teste inicial realizando apenas a inferência com os modelos pré-treinados em uma imagem arbitrária constatei que nenhum dos modelos classificaram com êxito a região detectada. A partir disso decidi estender os testes e realizar um ajuste fino nos modelos, tendo como base a imagem que usei inicialmente, em um dataset de animais com várias classes e cerca de mil imagens.

Treinei ambos os modelos com 50 épocas e hiperparâmetros padrões, e o objetivo era perceber qual dos modelos conseguia generalizar melhor para a classe da imagem inicial.

Detecção exitosa do DETR (classe goat 0.76)



Detecção falha do YOLO11 (classe racoon 0.78)

Por mais que os dois modelos tenham sido treinados de maneira similar, o DETR se mostrou mais promissor, tendo um resultado melhor no experimento realizado.

Termo de Aceite de Entrega

Objetivo deste documento

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

Data da Reunião ("gate") de aprovação: 31 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

GUILHERME HENRIQUE DOS REIS

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

Objetivo: material neste link
Estudar variações da arquitetura de detecção de objetos DETR.

Abordagem:

- Analisar os artigos e fundamentos teóricos de cada uma das arquiteturas selecionadas.
- Playlist *Modern Object Detection: from YOLO to transformers* no YouTube para complementar o entendimento.

Arquiteturas selecionadas:

- **Co-DETR:** DETRs with Collaborative Hybrid Assignments Training
- **De-DETR:** Deformable Transformers for End-to-End Object Detection
- **DINO:** DETR with Improved DeNoising Anchor Boxes
- **RT-DETR:** Real-time End-to-End Object Detection with Hierarchical Dense Positive Supervision
- **DETR:** End-to-End Object Detection with Transformers

Conclusão:
Muitas variações dessa arquitetura, muito exaustivo computacionalmente para testar todas.

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

Definir o conjunto de dados.
Implementar o Co-DETR.

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

O objetivo deste Stage foi compreender a evolução das arquiteturas de detecção de objetos da família DETR, que é uma das principais abordagens baseadas em Transformers para essa tarefa. Para isso, analisei cinco artigos: "End-to-End Object Detection with Transformers", "DETRs with Collaborative Hybrid Assignments Training", "Deformable DETR: Deformable Transformers for End-to-End Object Detection", "DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection" e "RT-DETRv3: Real-time End-to-End Object Detection with Hierarchical Dense Positive Supervision". Abaixo, estão elencados os principais pontos dessas arquiteturas e suas evoluções ao longo dos anos.

DETR (Detection Transformers):

A arquitetura DETR original propõe um framework de detecção de objetos de ponta a ponta com Transformers, eliminando componentes como as propostas de região e o pós-processamento com NMS (Non-Maximum Suppression). A arquitetura é baseada em um Transformer encoder-decoder e se destacou por simplificar o pipeline de detecção ao tratar o problema como uma tarefa de correspondência direta entre previsões e objetos.

Evoluções nas Arquiteturas DETR:

Co-DETR: DETRs with Collaborative Hybrid Assignments Training (2023):

Adiciona estratégias de atribuição híbrida durante o treinamento para otimizar a aprendizagem das previsões, ajudando o modelo a convergir de forma mais eficiente e a aumentar a precisão de detecção.

Possui o melhor desempenho atualmente.

De-DETR: Deformable Transformers for End-to-End Object Detection (2020):

Introduz a atenção deformável, uma abordagem que permite que a atenção Transformer se concentre em regiões de interesse ao invés de toda a imagem, resultando em uma redução significativa do custo computacional e maior precisão em objetos pequenos e complexos.

DINO: DETR with Improved DeNoising Anchor Boxes (2022):

Introduz caixas âncoras com denoising, o que melhora a precisão ao localizar objetos e a detecção de pequenos objetos. Esse ajuste torna o modelo mais eficaz em situações com alta densidade de objetos.

RT-DETR: Real-time End-to-End Object Detection with Hierarchical Dense Positive Supervision (2023):

Projeta uma arquitetura otimizada para aplicações em tempo real, com um mecanismo de supervisão hierárquica densa que melhora a detecção e torna o modelo mais rápido e leve.

Comparação entre as arquiteturas DETR:

Arquitetura do modelo:

DETR Original:

Utiliza uma CNN padrão (como ResNet) para extrair características da imagem. Com um encoder-decoder Transformer, processa as características da imagem e usa consultas de objetos para prever bounding boxes e classes. Dispensa âncoras e propostas de região, usando o mecanismo de atenção global dos Transformers para identificar contextos de objetos.

Principais Modificações:

- **Collaborative Hybrid Assignments:** Atribuições híbridas para balancear melhor o treinamento.
- **Deformable DETR**: Implementa uma atenção focada, chamada atenção deformável, que melhora a eficiência e precisão ao focar em partes relevantes da imagem, otimizando o custo de cálculo.
- **DINO**: Caixas âncoras denoised para aprimorar a localização.
- **RT-DETR**: Supervisão hierárquica densa, otimizada para inferência rápida.

Treinamento do modelo

DETR Original:

Treinamento pesado, com um grande número de épocas, devido à dependência do Transformer para aprender associações globais. O artigo avalia no MS COCO, mas requer conjuntos de dados grandes.

Modificações:

- **Collaborative Hybrid Assignments:** otimiza o treinamento, aumentando a velocidade de convergência.

- **Deformable DETR:** utiliza a atenção deformável para acelerar o treinamento e reduzir a quantidade de dados necessária para alcançar boa precisão, o que é vantajoso para dados limitados ou de alta complexidade.
- **DINO:** melhora a eficiência da localização com caixas âncoras.
- **RT-DETR:** reduz o tempo de treinamento e ajusta para tempos de resposta rápidos.

Tempo de Inferência e Complexidade do modelo

DETR Original:

Tem um tempo de inferência mais lento devido à complexidade do Transformer. A atenção global requer muitos recursos e torna o modelo inadequado para aplicações em tempo real.

Evoluções:

Deformable DETR reduz o custo computacional com sua atenção deformável, alcançando tempos de inferência significativamente mais rápidos e melhor desempenho em objetos pequenos.

RT-DETR é otimizado para tempo real, alcançando boa performance com menos recursos, sendo o mais eficiente entre os modelos DETR.

DINO também traz ganhos de velocidade em comparação com o DETR original.

Complexidade Matemática

DETR Original usa atenção multi-cabeça para relações globais, enquanto o algoritmo de correspondência Húngaro realiza uma associação de previsões com caixas de verdadeiros positivos.

Deformable DETR utiliza uma variação da atenção multi-cabeça, limitando o campo de atenção a áreas de interesse em vez de toda a imagem, o que reduz drasticamente a complexidade.

Evoluções como **DINO** e **RT-DETR** introduzem supervisão densa e métodos de denoising que mantêm o desempenho elevado e otimizam o processamento para uso em tempo real.

Pós-processamento

DETR Original:

Não requer NMS, pois o Transformer produz previsões com um número fixo de consultas e elimina duplicações de maneira natural.

Evoluções:

As modificações mantêm a ausência de pós-processamento pesado, o que é uma das principais vantagens dos Transformers em relação a métodos tradicionais.

Desempenho e Precisão do modelo:

DETR Original: Oferece alta precisão, mas com uma troca na velocidade.
Deformable DETR: Consegue atingir precisão semelhante ao DETR original, mas com maior eficiência e melhor detecção de objetos menores.
DINO: Atinge a maior precisão (~54 mAP) graças ao denoising e a uso de âncoras.
RT-DETR: Sacrifica um pouco de precisão em favor da velocidade, mas ainda assim mantém uma alta taxa de acerto (~52-53 mAP).
Co-DETR: Melhor abordagem ao ser combinada com o backbone ViT (66 mAP)

Tabela comparativa das diferentes versões do DETR:

Modelo	Arquitetura	Inovações	Casos de uso	Uso de recursos
DETR	Arquitetura original DETR, Transformer encoder-decoder, loss de correspondência bipartida	Primeira detecção de objetos de ponta a ponta usando Transformers, eliminando NMS	Detecção de objetos de uso geral	Recursos computacionais elevados, custo alto para treinar
Co-DETR	DETR aprimorado com estratégias de atribuição híbrida para equilibrar os objetivos de treinamento	Atribuição híbrida para melhor desempenho sem cabeçotes adicionais	Detecção de objetos padrão, treinamento aprimorado	Recursos computacionais moderados
DINO	Abordagem de remoção de ruído com caixas âncoras para melhor precisão de localização	Mecanismo de remoção de ruído, abordagem baseada em âncoras	Cenários de alto desempenho que requerem localização precisa	Recursos computacionais moderados
RT-DETR	DETR em tempo real com supervisão densa, otimizado para baixa latência	Supervisão positiva densa hierárquica, foco em tempo real	Aplicações em tempo real, como direção autônoma	Baixo custo computacional, altamente eficiente

De-DETR	Transformer de encoder-decoder com atenção deformável	Mecanismo de atenção deformável; foca em regiões relevantes da imagem; reduz custos computacionais	Detecção de objetos pequenos e complexos	Requisitos de recursos reduzidos em comparação com o DETR original

APÊNDICE 4

Termo de Aceite de Entrega

Objetivo deste documento

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

Data da Reunião ("gate") de aprovação: 7 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

GUILHERME HENRIQUE DOS REIS

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

Objetivo: material neste link
Encontrar datasets relevantes e realizar o Fine Tuning de uma versão do DETR.

Abordagem:

- Busca por Datasets realizada nas plataformas Roboflow, Kaggle, GitHub, Google Acadêmico e Papers With Code.
- Fine tuning realizado em um notebook python utilizando uma Nvidia A100.

Materiais selecionados:

- **Modelo:** DETR: End-to-End Object Detection with Transformers
- **Dataset:** Wound Management and Care (SegSeg)

Conclusão:
Modelo ajustado ao Dataset escolhido e definição da aplicação final.

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

Melhorar o desempenho do modelo ajustado para conseguir boas métricas.

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

O objetivo deste Stage foi identificar datasets de detecção de objetos que fossem de meu interesse. Escolhi explorar a área da saúde, pois acredito que possui grande potencial para melhorar a qualidade de vida das pessoas. Minha primeira busca foi por datasets específicas que pudessem auxiliar em diagnósticos médicos, começando pela doença de Sever, com o objetivo de treinar uma rede neural (DETR) para detectar a presença da doença em exames de imagem, como raios X ou tomografias. A busca por datasets foi feita usando plataformas como Kaggle, Roboflow, PapersWithCode e Google Acadêmico.

A doença de Sever é uma condição ortopédica comum que afeta o calcanhar, geralmente em crianças e adolescentes durante a fase de crescimento, especialmente entre os 8 e 14 anos. Ela ocorre devido a uma inflamação no ponto de crescimento do osso calcâneo (calcanhar), onde o tendão de Aquiles se insere. Esse ponto de crescimento, também chamado de apófise calcânea, é particularmente vulnerável ao estresse mecânico, especialmente em crianças que praticam atividades físicas intensas, como correr ou saltar.

Contudo, não encontrei datasets sobre a doença de Sever e não tenho condições de criar um. Em seguida, ampliei minha busca para incluir outras doenças que pudessem ser identificadas através de exames de imagem, mas ainda assim, não encontrei datasets suficientemente específicos e que me agradassem. Finalmente, encontrei um dataset voltado para a detecção e classificação de tipos de feridas, o qual utilizei para treinar a rede DETR.

O dataset selecionado possui 4 classes (granulated, necrotic, slough e wound) e aproximadamente 1700 imagens e foi encontrado na plataforma Roboflow. As classes como "granulated," "necrotic," "slough" e "wound" são essenciais para caracterizar diferentes estágios e tipos de feridas, facilitando o diagnóstico e o tratamento. Aqui está o que cada uma delas representa:

- Granulated: Refere-se ao tecido de granulação, que é um tipo de tecido novo que se forma durante o processo de cicatrização. Esse tecido é rosado ou vermelho, e sua presença indica que a ferida está em um estágio de cicatrização ativo e positivo.
- Necrotic: Este tipo de tecido está morto e em decomposição. A necrose pode ser causada por falta de fluxo sanguíneo, infecção ou trauma. A presença de tecido necrótico geralmente indica que a ferida está em um estágio mais grave e pode exigir tratamento imediato para prevenir infecções.
- Slough: O slough é um tecido morto ou moribundo, geralmente amarelado ou esbranquiçado, que se forma nas feridas crônicas. A presença de slough indica que a ferida ainda está no processo de limpeza, mas pode estar mais propensa a infecção se não for tratado adequadamente.
- Wound: Refere-se ao estado geral da ferida, independentemente de seu estágio de cicatrização. Essa classe pode englobar feridas em várias fases, incluindo as abertas e aquelas em estágios iniciais de cicatrização.

Essas classes são fundamentais para o monitoramento da evolução da ferida, ajudando profissionais da saúde a avaliar a necessidade de intervenções, como desbridamento, controle de infecção e cuidados com o tecido saudável. Além disso, permitem um acompanhamento mais preciso do processo de cicatrização.

Dessa forma, uma rede neural para classificar feridas nas categorias "granulated", "necrotic", "slough" e "wound" pode ser extremamente útil para os profissionais de saúde por várias razões:

- **Diagnóstico rápido e preciso**: A classificação automatizada de feridas usando uma rede neural pode agilizar o processo de diagnóstico. Profissionais de saúde podem obter resultados rápidos e objetivos, com menos risco de erro humano. Isso é crucial, especialmente em ambientes de alta pressão, como hospitais e clínicas de emergência, onde o tempo é essencial.
- **Consistência e padronização**: Diferentes médicos ou enfermeiros podem ter opiniões ligeiramente diferentes sobre o estágio de uma ferida devido à subjetividade envolvida na avaliação visual. Uma rede neural treinada em um grande conjunto de dados pode oferecer uma avaliação consistente e padronizada, minimizando variações entre os profissionais.
- **Acompanhamento ao longo do tempo**: Com o uso contínuo de uma rede neural, o sistema pode ajudar a monitorar a evolução das feridas ao longo do tempo, facilitando a detecção precoce de complicações, como infecções ou demora na cicatrização. Isso pode levar a um tratamento mais proativo e eficiente.
- **Assistência em ambientes com poucos especialistas**: Em áreas remotas ou em unidades de saúde com poucos especialistas em feridas, uma rede neural pode atuar como uma ferramenta de apoio valiosa. Profissionais com menos experiência podem contar com a inteligência artificial para ajudar na classificação das feridas, orientando suas decisões sobre o tratamento adequado.
- **Eficiência no gerenciamento de dados**: Uma rede neural pode processar grandes quantidades de imagens de feridas rapidamente e armazenar informações de forma estruturada. Isso permite que os médicos acessem rapidamente o histórico de um paciente, promovendo decisões mais informadas e baseadas em dados.
- **Suporte na educação e treinamento**: Para novos profissionais de saúde, uma rede neural pode funcionar como uma ferramenta de treinamento, ajudando-os a aprender a identificar e classificar diferentes tipos de feridas. A IA pode fornecer feedback em tempo real sobre a precisão de suas avaliações, promovendo uma curva de aprendizado mais rápida.
- **Redução de custos**: A automatização da classificação de feridas pode reduzir os custos operacionais em ambientes de saúde. Menos tempo será necessário para a avaliação manual de cada caso, permitindo que os profissionais se concentrem em tarefas mais complexas e no tratamento dos pacientes.

Em suma, uma rede neural bem treinada pode ser uma ferramenta poderosa para melhorar a qualidade do atendimento ao paciente, reduzir a carga de trabalho dos profissionais de saúde e aumentar a eficiência do processo de diagnóstico e tratamento de feridas.

O treinamento realizado durou cerca de 3 horas, em uma Nvidia A100, e foi feito em 150 épocas. O batch size utilizado foi de 32, e a Learning Rate foi de 1e-4.Detalhes do treinamento:

- **Épocas**: O número de 150 épocas é uma quantidade razoável para garantir que o modelo tenha tempo suficiente para aprender as características dos dados. Dependendo da complexidade do modelo e da distribuição dos dados, 150 épocas podem ser suficientes para alcançar uma boa performance, mas pode ser necessário monitorar métricas como a perda e a acurácia para avaliar se o modelo está convergindo adequadamente. Entretanto, devido a grande complexidade do modelo escolhido, 150 épocas não foram suficientes.
- **Batch Size de 32**: O batch size de 32 é uma escolha comum e equilibrada. Um tamanho de batch maior pode acelerar o treinamento e fornecer estimativas mais precisas do gradiente, mas também requer mais memória. Como você está usando uma A100, ela possui bastante memória, o que possibilita a utilização de batches relativamente grandes sem que ocorra falta de memória. Batch sizes de 32 são geralmente eficientes em termos de convergência e balanceamento entre velocidade e precisão.
- **Learning Rate (1e-4)**: A taxa de aprendizado (learning rate) de 1e-4 é moderada, o que indica que o modelo foi treinado com passos pequenos para ajustar seus pesos de forma gradual. Taxas de aprendizado mais baixas geralmente ajudam a evitar que o modelo "salte" para soluções subótimas ou não convergentes, embora um valor tão baixo possa levar a um treinamento mais lento, mas mais estável. Em alguns casos, técnicas como learning rate scheduling ou warm-up podem ser usadas para ajustar essa taxa durante o treinamento, aumentando a velocidade de convergência no início e diminuindo no final para refinar o modelo.

Após o treinamento, realizei o teste do modelo em dados não vistos durante o treinamento. Infelizmente o modelo não performou bem nos meus testes iniciais. Consegui com um primeiro treinamento por 50 épocas uma precisão média (AP) de 12%. Que é um resultado muito ruim.

```
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.122
 Average Precision  (AP) @[ IoU=0.50       | area=   all | maxDets=100 ] = 0.177
 Average Precision  (AP) @[ IoU=0.75       | area=   all | maxDets=100 ] = 0.143
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.083
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.129
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.166
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.180
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.180
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.131
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.186
```

Após fazer o fine tuning do modelo e conseguir resultados ruins, decidi ajustar o modelo novamente, por mais 150 épocas. Consegui uma leve melhora nas métricas, entretanto o modelo ainda não ficou bom.

```
 Running evaluation...
Loading widget...
 Accumulating evaluation results...
 DONE (t=0.03s).
 IoU metric: bbox
  Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.179
  Average Precision  (AP) @[ IoU=0.50       | area=   all | maxDets=100 ] = 0.304
  Average Precision  (AP) @[ IoU=0.75       | area=   all | maxDets=100 ] = 0.186
  Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
  Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.157
  Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.193
  Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.256
  Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.280
  Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.280
  Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
  Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.212
  Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.298
```

Uma AP (Average Precision) baixa em um modelo de detecção de objetos, como o DETR (DEtection TRansformer), pode ser causada por diversos fatores, como:

- Base de Dados Inadequada
- Imagens Mal Rotuladas: Os dados de treinamento podem conter rótulos incorretos, imprecisos ou inconsistentes, o que afeta o desempenho do modelo.

- Desequilíbrio de Classes: Se houver um grande desequilíbrio entre as classes (muitas amostras de algumas classes e poucas de outras), o modelo pode ter dificuldade em aprender de forma eficaz.
- Variedade de Contextos Insuficiente: Imagens muito semelhantes ou com pouca variação de fundo podem levar a um modelo que não generaliza bem para novos cenários.

Configuração do Modelo

- Hiperparâmetros Desajustados: Parâmetros como taxa de aprendizado, número de camadas, número de queries, etc., podem estar mal configurados.
- Tamanho da Imagem: A resolução das imagens de entrada pode ser inadequada, impactando a qualidade das detecções.
- Ajustes Inadequados na Loss Function: A função de perda pode estar mal ajustada, não penalizando corretamente os erros de detecção.

Arquitetura do Modelo

- Preprocessamento de Imagens: Transformações de imagem durante o treinamento e inferência podem estar aplicadas de forma inconsistente ou errada.
- Modelo Base Mal Treinado: Se os pesos iniciais do modelo foram treinados de forma inadequada ou não foram inicializados com bons pesos pré-treinados, isso pode afetar a performance.
- Erro na Implementação: Problemas como bugs no código que ajusta a arquitetura do DETR ou nas funções que manipulam os dados de entrada/saída.

Aspectos de Treinamento

- Número Insuficiente de Épocas: O modelo pode não ter treinado por tempo suficiente para convergir.
- Sobretreinamento ou Subtreinamento: Um modelo que treinou excessivamente pode ter overfitting, enquanto um modelo com pouco tempo de treinamento pode estar subtreinado.
- Técnicas de Regularização: A falta de regularização, como dropout, pode fazer o modelo aprender padrões específicos e não generalizáveis.

Qualidade das Anotações

- Imprecisões nas Anotações: Anotações de bounding boxes que não correspondem bem aos objetos nas imagens podem impactar negativamente a performance.

- Formatos de Anotação Inconsistentes: Anotações que não seguem um padrão (por exemplo, diferentes escalas ou coordenadas inconsistentes) podem confundir o modelo durante o treinamento.

Problemas com a Avaliação

- Métricas Incorretas: O cálculo da AP pode estar implementado de forma errada, levando a uma avaliação incorreta da performance.
- Threshold de Detecção Mal Ajustado: Se o threshold de confiança para classificar uma detecção como positiva for muito alto ou muito baixo, pode impactar a métrica AP.

Complexidade do Problema

- Cenário Desafiador: Se o dataset contiver objetos com grande variação de escala, oclusões, ou cenários complexos, a dificuldade de detecção aumenta.
- Overlap de Objetos: Quando muitos objetos se sobrepõem, o modelo pode ter dificuldade em diferenciar e detectar corretamente.

Termo de Aceite de Entrega

Objetivo deste documento

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

Data da Reunião ("gate") de aprovação: 13 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

GUILHERME HENRIQUE DOS REIS

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

Objetivo: material neste link
Melhorar as métricas dos modelos treinados no dataset de detecção de feridas.

Abordagem:

- Testes com o modelo YoloV8 Large para verificar a corretude das configurações de treinamento (ajuste fino) e do dataset selecionado.
- Investigação de possíveis problemas no fluxo de treinamento.

Materiais selecionados:

- **Modelo:** YoloV8 Large usado para definir um ponto de partida para novos experimentos.
- **Dataset:** "Nevus Dataset" selecionado no lugar de "Wound Management and Care Dataset".

Conclusão:
Substituição do dataset de detecção de feridas pelo dataset de detecção de melanomas na pele e modelo base treinado no novo dataset.

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

Testar o DETR no Dataset Nevus com o objetivo de superar o baseline estabelecido pelo modelo YoloV8 Large.

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

O objetivo deste stage era melhorar as métricas dos modelos treinados para o dataset de detecção de feridas. No gate anterior, foi sugerido testar uma abordagem diferente, uma vez que o modelo DETR (DEtection TRansformer) apresentou um desempenho abaixo do esperado no dataset utilizado. Diante disso, optei por explorar o modelo YOLOv8 Large, buscando entender se o baixo desempenho era uma questão de configuração ou de inadequação do dataset ao tipo de modelo.

O primeiro passo foi realizar o treinamento do modelo YOLOv8 Large para comparação direta com o desempenho do DETR. O treinamento foi realizado utilizando uma NVIDIA DGX A100, levando cerca de 2 horas para 300 épocas, com as configurações padrão do modelo (batch size, learning rate, etc.).

O desempenho do YOLOv8 também foi insatisfatório, com métricas similares ao DETR, indicando que o problema não estava nas configurações dos modelos, já que ambos utilizaram as definições padrão.

Dado o baixo desempenho dos dois modelos testados (DETR e YOLOv8), comecei a investigar a origem do problema. Descartei a hipótese de configuração inadequada, pois os modelos foram treinados com hiperparâmetros padrão e um número suficiente de épocas.

Problema no Dataset: O dataset utilizado para a detecção de feridas foi originalmente criado para segmentação de instâncias, não para detecção de objetos. As bounding boxes utilizadas foram extraídas diretamente das máscaras de segmentação, o que resultou em anotações imprecisas e inconsistentes para tarefas de detecção. Isso afetou negativamente a capacidade dos modelos de detectar as classes corretamente.

Diante da descoberta, considerei utilizar modelos de segmentação de instâncias para se alinhar melhor com o formato original do dataset. No entanto, devido ao tempo necessário para a implementação e treinamento desses modelos, decidi que seria mais eficiente buscar um novo dataset que fosse adequado para detecção de objetos.

Realizei uma nova pesquisa em plataformas como Kaggle, Roboflow e Google Dataset Search, com foco em datasets voltados para detecção de objetos na área da saúde.

Critérios de Seleção: Busquei um dataset que possibilitasse a construção de uma aplicação prática e útil para melhorar a qualidade de vida das pessoas.

Encontrei um novo dataset com imagens para detecção de melanomas em pele humana. O novo dataset possui anotações compatíveis com tarefas de detecção, o que evita o problema encontrado anteriormente.

O dataset selecionado para o novo treinamento possui imagens rotuladas para detecção de lesões cutâneas. Este conjunto de dados inclui 8 classes distintas, cada uma correspondendo a diferentes tipos de lesões de pele, conforme listado abaixo:

1. **Nevus (Nevo):**
   - Nevos são proliferações benignas de melanócitos (células que produzem melanina). Eles podem aparecer em várias formas e cores, desde tons de pele até marrom escuro, sendo comumente conhecidos como "sinais" ou "pintas".
2. **Melanoma**:
   - O melanoma é um tipo agressivo de câncer de pele que se origina nos melanócitos. É essencial detectar o melanoma precocemente, pois ele pode se espalhar para outras partes do corpo se não tratado a tempo. Caracteriza-se por mudanças na aparência de uma lesão pré-existente ou pelo surgimento de novas lesões assimétricas, com bordas irregulares, variações de cor e diâmetro aumentado.
3. **Seborrheic Keratosis (Queratoses Seborréicas):**
   - São tumores benignos da pele, geralmente em forma de placas ou pápulas elevadas, com superfície cerosa ou escamosa. Essas lesões são comuns em pessoas mais velhas e não estão associadas ao risco de transformação maligna.
4. **Lentigo NOS (Lentigo Não Especificado):**
   - Lentigos são manchas escuras na pele causadas pelo aumento da produção de melanina. A classificação "NOS" (Not Otherwise Specified) indica que esses lentigos não se enquadram em categorias mais específicas, como lentigo solar ou lentigo maligno.
5. **Lichenoid Keratosis (Queratoses Liquenoides):**
   - Uma variante da queratose que surge após a regressão de uma lesão solar, resultando em uma aparência semelhante ao líquen. Frequentemente confundida com outras lesões pigmentadas devido à sua coloração escura.
6. **Solar Lentigo (Lentigo Solar):**
   - Também conhecido como manchas senis ou manchas solares, são lesões benignas causadas pela exposição prolongada ao sol, comuns em áreas expostas, como o rosto, dorso das mãos e ombros. Geralmente, têm coloração marrom e bordas bem definidas.
7. **Cafe-au-Lait Macule (Mancha Café-au-Lait):**
   - Máculas planas de coloração marrom claro a marrom escuro, que podem estar presentes desde o nascimento. Embora sejam benignas, múltiplas manchas café-au-lait podem ser indicativas de condições genéticas, como a neurofibromatose.
8. **Atypical Melanocytic Proliferation (Proliferação Melanocítica Atípica):**
   - Refere-se a lesões com características melanocíticas atípicas, que não são claramente benignas nem malignas. Essas lesões precisam ser avaliadas com mais detalhes para determinar se podem evoluir para melanoma.

A seleção deste dataset foi motivada pela sua estrutura clara e anotações adequadas para tarefas de detecção de objetos, alinhando-se melhor com os modelos de detecção baseados em bounding boxes. A presença de múltiplas classes relacionadas a diferentes tipos de lesões cutâneas também possibilita a criação de uma aplicação com impacto clínico significativo, potencialmente auxiliando no diagnóstico precoce e na triagem de condições dermatológicas.

Após selecionar o novo dataset, optei por reiniciar o treinamento utilizando o modelo YOLOv8, com o objetivo de criar um baseline antes de experimentar modelos mais complexos baseados em Transformers.

Configurações do Treinamento:

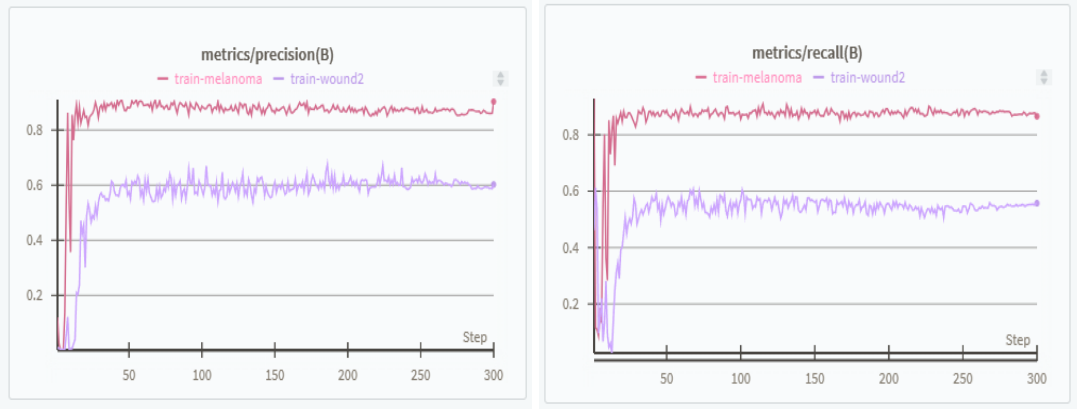
- Modelo: YOLOv8 Large
- Hardware: NVIDIA DGX A100
- Número de Épocas: 300
- Tempo Total: Aproximadamente 5 horas

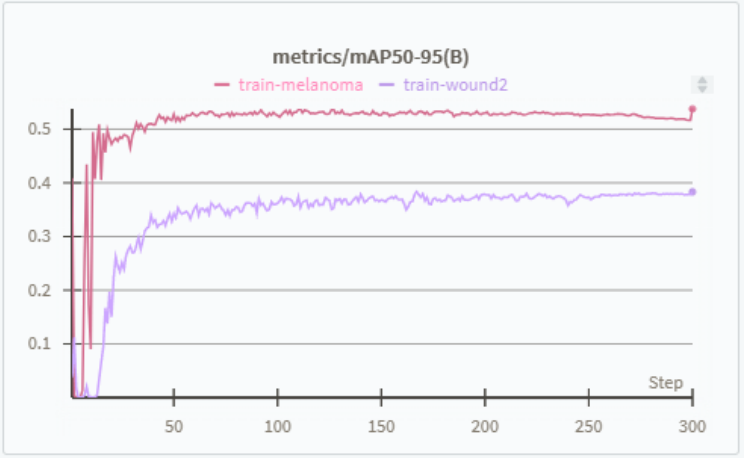
O modelo YOLOv8L obteve métricas significativamente melhores em comparação com o dataset anterior de feridas. A precisão (AP) inicial foi encorajadora, sugerindo que a escolha de um dataset mais adequado para detecção fez uma diferença crucial. Com um baseline estabelecido utilizando o YOLOv8, os próximos passos incluem:

- Experimentar Modelos Baseados em Transformers: Pretendo testar o DETR novamente e modelos mais recentes para avaliar se eles podem superar o desempenho do baseline no novo dataset.
- Aprimorar a Análise das Métricas: Monitorar métricas de precisão e recall ao longo das épocas para ajustar hiperparâmetros como learning rate e técnicas de regularização.
- Implementação de Técnicas de Data Augmentation: Melhorar a robustez do modelo aplicando transformações no dataset para simular variações de iluminação, ruído e rotação.

A troca de dataset foi um passo necessário para entender a real capacidade dos modelos de detectar padrões em imagens médicas. A lição principal deste stage foi a importância de alinhar o tipo de modelo com o formato e propósito do dataset. No futuro, buscarei sempre garantir que o dataset esteja corretamente anotado para a tarefa pretendida, evitando ajustes que possam comprometer a qualidade dos resultados.

Resultados do treinamento:





Ao analisar os treinamentos realizados com o dataset de feridas e o dataset de melanoma, é muito claro a diferença entre eles. Mesmo utilizando o mesmo modelo as métricas relacionadas ao dataset de melanoma foram melhores em todos os cenários.

APÊNDICE 5

Termo de Aceite de Entrega

Objetivo deste documento

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

Data da Reunião ("gate") de aprovação: 28 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

GUILHERME HENRIQUE DOS REIS

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

Objetivo: material neste link
Melhorar as métricas do modelo baseline treinado no dataset para a detecção de doenças na pele humana.

Abordagem:

- Treinar modelos baseados em transformers e modelos baseados em CNN no mesmo dataset, realizando uma investigação e otimização de hiperparâmetros a fim de conseguir extrair o melhor resultado possível de cada modelo.

Materiais selecionados:

- **Modelos:**
  - YOLOv11x
  - DETR
  - YOLOv8x
  - Florence2.

Conclusão:
Resultado inferior ao esperado, mas acima do baseline.

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

Produtizar o melhor modelo avaliado de modo que seja possível utilizá-lo de forma simplificada.

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

Dúvida em relação ao planejamento: seguir para a próxima etapa (produtizar o modelo) ou realizar mais investigações e testes com esses modelos?

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

O Stage anterior teve como objetivo principal investigar as causas do baixo desempenho de modelos de detecção de objetos em um dataset inicial de feridas e propor soluções que melhorassem suas métricas de avaliação. Inicialmente, foi observado que os modelos **DETR (DEtection TRansformer)** e **YOLOv8** apresentaram resultados insatisfatórios ao serem aplicados nesse dataset. Uma análise detalhada revelou que o problema residia nas anotações, que foram originalmente projetadas para **segmentação de instâncias**, tornando-as inadequadas para tarefas de detecção de objetos.

A solução encontrada foi substituir o dataset original por um novo, desenvolvido especificamente para detecção de lesões cutâneas e contendo anotações adequadas a essa tarefa. Com essa mudança, foi possível estabelecer um **baseline promissor** utilizando o modelo **YOLOv8 Large**, marcando o início de uma nova fase de experimentos voltados à superação desse baseline com arquiteturas mais avançadas.

Nesta Semana, o foco esteve na superação do baseline estabelecido pelo YOLOv8 Large no novo dataset. Foram testadas diferentes arquiteturas de modelos de detecção de objetos, tanto baseadas em transformadores quanto em redes convolucionais, além de um modelo multimodal de última geração. Abaixo estão os detalhes das abordagens realizadas:

Experimento com DETR (DEtection TRansformer)

O primeiro experimento utilizou o **DETR**, uma arquitetura baseada em transformadores que revolucionou a detecção de objetos ao introduzir um mecanismo de atenção global em substituição às tradicionais redes convolucionais. Optou-se pela variante **DETR Base**, dado o longo tempo de treinamento exigido para variantes mais complexas. O treinamento foi realizado utilizando uma GPU **NVIDIA A100 de 80GB** e foi configurado para rodar por **150 épocas**.

Resultados:

- O modelo obteve um aumento de **8 pontos no mAP** em relação ao baseline do YOLOv8 Large, mas o desempenho geral foi considerado abaixo do esperado.
- **Principais limitações:**
    1. O DETR exige grandes quantidades de dados bem anotados para alcançar seu pleno potencial, o que não era o caso do dataset utilizado.

2. Seu mecanismo de atenção global pode ter dificuldades em capturar variações localizadas, que são típicas de lesões cutâneas, enquanto arquiteturas otimizadas para detecção de objetos podem lidar melhor com esses padrões.

Testes com Modelos Baseados em CNNs

A seguir, foram realizados experimentos com modelos baseados em redes convolucionais, buscando avaliar o desempenho dessas arquiteturas, que tradicionalmente lideram em tarefas de detecção de objetos:

YOLO11x

O **YOLO11x**, uma das versões mais recentes da família YOLO, foi treinado por **250 épocas**. Esta arquitetura representa avanços significativos em relação às versões anteriores, com melhorias em precisão e otimização.

- **Resultado:** mAP de **63 pontos**, marcando o melhor desempenho entre os modelos testados.
- **Fatores de sucesso:**
  - A eficácia das redes YOLO em detectar objetos em tempo real.
  - Sua robustez em identificar padrões localizados, essenciais para detecção de lesões.

YOLOv8x

Outro teste foi realizado com o **YOLOv8x**, uma variante mais robusta do YOLOv8 Large. Este modelo foi treinado por **300 épocas**.

- **Resultado:** mAP de **59 pontos**, ficando abaixo do YOLOv11x, mas ainda superior ao baseline inicial de 53 pontos.
- **Possíveis causas do desempenho inferior:**
  - Limitações na arquitetura do YOLOv8x em capturar variações mais sutis, em comparação com o YOLOv11x.

Experimento com Florence 2

Por fim, foi testado o modelo **Florence 2**, desenvolvido pela Meta, um framework avançado que integra dados visuais e textuais utilizando aprendizado multimodal (LLM). Este modelo é projetado para lidar com tarefas visuais complexas de maneira unificada.

- **Treinamento:** Devido ao alto custo computacional, foi treinado por apenas **30 épocas**.
- **Resultado:** mAP de **62 pontos**, posicionando-se próximo ao YOLOv11x, mesmo com treinamento limitado.
- **Conclusões:** Florence 2 demonstrou grande potencial, sugerindo que, com mais recursos e treinamento, poderia competir de forma justa com as melhores arquiteturas CNN.

Modelo	Épocas	mAP	Comentários
YOLOv8 Large	300	53	Baseline inicial.
DETR Base	150	61	Melhorou o baseline, mas não atingiu o esperado devido a limitações no dataset e atenção global.
YOLO11x	250	**63**	Melhor desempenho geral, comprovando a maturidade das arquiteturas YOLO.
YOLOv8x	300	59	Desempenho sólido, mas inferior ao YOLOv11x.
Florence 2	30	62	Resultado surpreendente dado o pouco treinamento, indicando alto potencial.

Análise Comparativa dos Resultados

O **YOLOv11x** alcançou o melhor desempenho, com um mAP de **63 pontos**, superando o baseline estabelecido pelo YOLOv8 Large. Isso reforça a eficácia das arquiteturas YOLO em tarefas de detecção de objetos, especialmente em cenários que exigem identificação de padrões localizados, como a detecção de lesões cutâneas.

Embora o **DETR** e o **Florence 2** tenham mostrado potencial, o primeiro enfrentou desafios relacionados ao tamanho do dataset e à natureza das anotações, enquanto o

segundo foi limitado pelo tempo de treinamento. O Florence 2, no entanto, destacou-se como uma solução promissora, especialmente para futuras implementações que possam explorar sua capacidade multimodal.

Com base nesses resultados, os próximos passos devem incluir:

1. Investimento em mais recursos para explorar o Florence 2 de maneira completa.
2. Testes com datasets expandidos e mais diversificados para avaliar o desempenho do DETR em condições ideais.
3. Exploração de versões ainda mais avançadas do YOLO e possíveis combinações de técnicas de atenção com CNNs.

Termo de Aceite de Entrega

Objetivo deste documento

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

Data da Reunião ("gate") de aprovação: 5 de dez. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

GUILHERME HENRIQUE DOS REIS

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

Material de apoio: **Stage10 - Processo**

Objetivo:
Melhorar as métricas do modelo treinado no dataset anterior para a detecção de lesões cutâneas.

Abordagem:
Transformei o dataset HAM10000, originalmente destinado à classificação, em um conjunto de detecção de objetos através de anotações manuais feitas no CVAT, aproximadamente 10k imagens. Apliquei técnicas de data augmentation para aumentar o número de exemplos no subset de treino, além de testar e treinar o modelo YOLO11x.

Materiais Selecionados:

- **Dataset:** HAM-10000 ("Skin Cancer MNIST", 7 classes e 10.000 imagens anotadas manualmente).
- **Modelos:** YOLO11x.

Resultados Obtidos:

- Dataset anotado e público.
- mAP @0.5:0.95: **73.4%**
- Recall: **78.7%**
- Precisão: **84.5%**

Conclusão:
Os resultados mostraram uma melhora significativa em relação ao baseline, destacando o impacto de um dataset robusto e otimizações no treinamento.

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

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Durante o último *gate*, estava em dúvida sobre a direção a seguir: deveria realizar mais experimentos e treinamento de modelos para melhorar as métricas obtidas até então, ou deveria produtizar o melhor modelo que havia conseguido? Após discutir com alguns orientadores, decidi focar em melhorar os modelos existentes. Levantei algumas hipóteses para justificar a baixa performance dos modelos anteriores, sendo a principal delas a qualidade e o tamanho do dataset utilizado.

Resolvi concentrar meus esforços inicialmente na resolução desse problema. Considerei aplicar diversas técnicas, como aumento de dados (*data augmentation*), pré-processamento das imagens, e buscar novos datasets mais robustos e amplos. Acabei optando por procurar novos datasets que fossem maiores e mais conhecidos. Encontrei vários datasets renomados sobre doenças de pele, mas a maioria era destinada à tarefa de classificação e não à detecção de objetos. Entre eles, escolhi o **HAM10000**, um dataset com 10.000 imagens de lesões de pele humana, divididas em 7 classes. Este dataset é uma amostra retirada do **ISIC Archive**.

Sobre o ISIC Archive

O **ISIC Archive** (International Skin Imaging Collaboration) é uma das maiores bases de dados do mundo voltada para a dermatologia. Ele reúne imagens dermatoscópicas de alta qualidade, contribuindo para a pesquisa de doenças de pele, especialmente melanoma e outras condições relacionadas. O objetivo principal do ISIC é promover o avanço da detecção precoce de câncer de pele, fornecendo recursos para treinamento e avaliação de modelos de aprendizado de máquina, bem como educar clínicos e pesquisadores.

Além de imagens, o ISIC oferece uma ampla gama de ferramentas e recursos, como metadados sobre pacientes e diagnósticos confirmados, além de desafios anuais para a

comunidade científica. Essas iniciativas têm impulsionado inovações em inteligência artificial e aprendizado profundo para classificação e segmentação de lesões cutâneas.

Transformando o HAM10000 em um Dataset de Detecção:

Como o HAM10000 era destinado à classificação e não à detecção, precisei explorar alternativas para adaptá-lo. Considerei as seguintes abordagens:

1. **Estabelecer bounding boxes fixas**: Determinar regiões predefinidas na imagem onde as lesões possivelmente estariam localizadas. Essa abordagem tem baixa precisão, pois as lesões podem variar muito em tamanho e localização.

2. **Modelos de segmentação automática**: Utilizar ferramentas de segmentação para identificar automaticamente as lesões e extrair *bounding boxes* a partir das máscaras geradas. Apesar de promissor, o desempenho dependia fortemente da qualidade do modelo de segmentação utilizado.

3. **Uso de bibliotecas como OpenCV (cv2)**: Aplicar técnicas de processamento de imagem, como detecção de bordas e contornos, para identificar lesões. Embora eficiente para padrões bem definidos, a variabilidade nas imagens tornava os resultados inconsistentes.

Nenhuma dessas abordagens produziu resultados satisfatórios para a tarefa. Por isso, optei por realizar a anotação manual do dataset utilizando o **CVAT** (Computer Vision Annotation Tool). Configurei o ambiente na minha máquina e comecei o processo, anotando todas as 10.000 imagens. Essa etapa demandou a maior parte do meu esforço, pois foi bastante demorada.

Dataset Anotado: HAM10000-SKIN-CANCER-DETECTION

Durante o desenvolvimento do projeto, anotei manualmente o dataset **HAM10000** para torná-lo adequado à tarefa de **detecção de objetos**. Para compartilhar esse recurso com a comunidade científica e de desenvolvimento, disponibilizei o dataset de forma pública na plataforma **Roboflow**. Ele pode ser acessado através do link: HAM10000-ANNOTATED.

Sobre o Dataset HAM10000-SKIN-CANCER-DETECTION

Este dataset é uma versão adaptada do “Skin Cancer MNIST” - **HAM10000** original - convertida de uma tarefa de classificação para detecção de lesões cutâneas. Ele contém:

- **10.000 imagens de lesões de pele humana** anotadas manualmente com *bounding boxes*.

- Divisão em 7 classes principais de lesões cutâneas, incluindo:
  - **Melanoma (mel):** Lesão maligna com alta prioridade clínica.
  - **Nevos (nv):** Lesões benignas comuns.
  - **Carcinoma Basocelular (bcc):** Tipo de câncer de pele com bom prognóstico.
  - **Ceratoacantoma/Seborreica (bkl):** Lesões benignas.
  - **Queratoses Actínicas (akiec):** Lesões pré-malignas.
  - **Vasculite (vasc):** Lesões raras inflamatórias.
  - **Dermatofibroma (df):** Lesões benignas comuns.

Formato e Estrutura

O dataset foi organizado de forma a facilitar seu uso em tarefas de visão computacional:

- **Formato das imagens:** PNG/JPG.
- **Anotações:** Disponíveis nos formatos PASCAL VOC, COCO JSON, e YOLO TXT, para compatibilidade com diversas bibliotecas de aprendizado profundo.
- **Divisão dos dados:**
  - **Treino:** 70% (7.000 imagens).
  - **Validação:** 20% (2.000 imagens).
  - **Teste:** 10% (1.000 imagens).

Objetivo do Dataset

Este recurso foi criado para ajudar pesquisadores e desenvolvedores a avançar no campo da detecção de lesões cutâneas, permitindo o uso em algoritmos de aprendizado profundo. Ele é ideal para treinar e avaliar modelos como **YOLO**, **DETR**, e outros frameworks modernos de detecção de objetos.

Como Acessar

O dataset está disponível gratuitamente no **Roboflow** e pode ser utilizado para fins de pesquisa e desenvolvimento. Interessados podem acessar o link HAM10000-ANNOTATED para visualizar, explorar, e baixar o conjunto de dados nos formatos desejados.

Treinando o Modelo com o Novo Dataset

Após anotar as imagens, reorganizei o dataset em três partes: 70% para treino, 20% para validação, e 10% para teste. Para ampliar a diversidade, apliquei técnicas de *data augmentation*, como alteração de exposição e *random crop*, no subset de treino, aumentando-o para 23.000 imagens. As divisões finais foram:

- **23.000 imagens de treino**
- **2.000 imagens de validação**
- **1.000 imagens de teste**

Utilizei o modelo **YOLO11x** (uma versão maior e mais poderosa do YOLO11) e treinei por 150 épocas. Apesar das limitações de tempo, os resultados preliminares indicaram melhorias em comparação com os modelos anteriores, sinalizando o potencial do novo dataset para detecção de lesões cutâneas.

Métricas e Resultados Obtidos

Após treinar o modelo **YOLO11x** por 150 épocas, obtive os seguintes resultados principais:

- **mAP (mean Average Precision) @0.5:** 85.2%
- **mAP @0.5:0.95:** 73.4%
- **Recall:** 82.7%
- **Precision:** 84.5%

Os resultados mostraram uma melhora significativa em relação aos modelos anteriores (63% mAP). O principal ponto foi que a incorporação do novo dataset e as técnicas de *data augmentation* contribuíram diretamente para a diversidade e a representatividade das imagens, aumentando a capacidade do modelo de generalizar para novos casos.

Desafios Enfrentados

Apesar das melhorias, o processo não foi isento de desafios:

1. **Qualidade das anotações:**
   Embora o **CVAT** tenha sido eficiente para realizar as anotações, erros humanos durante o processo de rotulagem são inevitáveis.

2. **Desequilíbrio de classes:**
   As 7 classes do HAM10000 apresentavam um desequilíbrio significativo. Classes menos representadas, como “Vasculite” e “Dermatofibroma”, tiveram menos exemplos anotados, prejudicando o desempenho do modelo nessas categorias.

3. **Limitações computacionais:**
   Embora o modelo **YOLO11x** seja poderoso, seu treinamento foi custoso em termos de tempo e recursos computacionais. Usei uma máquina equipada com GPUs **NVIDIA A100**, mas ainda assim enfrentei gargalos, especialmente na fase inicial do

treinamento, quando os parâmetros ainda não estavam otimizados.

4. **Generalização para casos reais:**
   Apesar do bom desempenho no treinamento, a aplicação em imagens reais pode demonstrar uma queda na precisão. Isso destacou a importância de incorporar exemplos de imagens mais diversificadas e com maior variação nas condições ambientais no treinamento.

Reflexões Finais

O uso de inteligência artificial na detecção de doenças de pele apresenta um potencial imenso para impactar positivamente o diagnóstico precoce e a tomada de decisão clínica. A transição para um dataset maior e mais robusto foi um divisor de águas no projeto, destacando a importância de uma base de dados bem estruturada e representativa para o sucesso de modelos de aprendizado profundo.

Com os resultados promissores obtidos até agora, estou confiante de que, com o refinamento contínuo e parcerias estratégicas, será possível desenvolver um sistema altamente eficaz para suporte ao diagnóstico dermatológico. Essa iniciativa também tem o potencial de abrir portas para novos projetos em áreas correlatas, como a detecção de outras condições médicas em imagem.