UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
CURSO DE GRADUAÇÃO EM ENGENHARIA ELETRÔNICA

Alexandre Soli Soares

Uma Ferramenta para auxílio a terapias do Transtorno do Espectro Autista usando Aprendizado de Máquina

Florianópolis
2023

Alexandre Soli Soares

Uma Ferramenta para auxílio a terapias do Transtorno do Espectro Autista usando Aprendizado de Máquina

Trabalho de Conclusão do Curso de Graduação em Engenharia Eletrônica do Centro Tecnológico da Universidade Federal de Santa Catarina para a obtenção do título de Engenheiro Eletrônico.
Orientador: Prof. Jônata Tyska Carvalho, Ph.D.

Florianópolis
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Soares, Alexandre Soli
Uma Ferramenta para auxílio a terapias do Transtorno do Espectro Autista usando Aprendizado de Máquina / Alexandre Soli Soares ; orientador, Jônata Tyska Carvalho, 2023.
65 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia Eletrônica, Florianópolis, 2023.

Inclui referências.

1. Engenharia Eletrônica. 2. Transtorno do Espectro Autista. 3. Visão Computacional. 4. Aprendizado de Máquina. 5. Sistema Web. I. Carvalho, Jônata Tyska. II. Universidade Federal de Santa Catarina. Graduação em Engenharia Eletrônica. III. Título.

Alexandre Soli Soares

Uma Ferramenta para auxílio a terapias do Transtorno do Espectro Autista usando Aprendizado de Máquina

Este Trabalho de Conclusão foi julgado adequado para obtenção do Título de "Engenheiro Eletrônico" e aprovado em sua forma final pelo Curso de Graduação em Engenharia Eletrônica.

Florianópolis, 13 de fevereiro de 2023.

Documento assinado digitalmente
Fernando Rangel de Sousa
Data: 04/07/2023 20:51:50-0300
CPF: ***.649.114-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Fernando Rangel de Sousa, Dr.
Coordenador do Curso

Banca Examinadora:

Documento assinado digitalmente
Jonata Tyska Carvalho
Data: 03/07/2023 08:25:07-0300
CPF: ***.301.250-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Jônata Tyska Carvalho, Ph.D.
Orientador

Documento assinado digitalmente
Danilo Silva
Data: 04/07/2023 11:17:08-0300
CPF: ***.557.924-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Danilo Silva, Ph.D.
Avaliador
EEL - UFSC

Documento assinado digitalmente
CLAUDIO MACHADO DINIZ
Data: 03/07/2023 10:30:40-0300
CPF: ***.311.690-**
Verifique as assinaturas em https://v.ufsc.br

Prof.(a) Cláudio Diniz, Dr.
Avaliador
INF - UFRGS

Dedico esse trabalho aos meus pais Adriana e Soli,
à minha namorada Ursula e à minha cachorrinha Bolinha.

AGRADECIMENTOS

Agradeço aos meus pais, Adriana e Soli, à minha namorada Ursula, à minha estrelinha que está no céu Bolinha, aos meus dois melhores amigos Leonardo e Melchior, aos meus tios Manoel e Sirlene, e aos meus orientadores e amigos do laboratório de pesquisa Jônata, Mateus e Guilherme. Obrigado pai e mãe pelo amor, presença e incentivo, sempre. Amor, obrigado por ser minha metade e companheira. Bolinha, obrigado por me ensinar a apreciar o vento e ser minha amiga. Guris, obrigado por 10 anos das melhores lan parties e amizade. Tio e tia, obrigado pelo convite e estadia com vocês em Florianópolis e apoio. Orientadores e amigos do lab, obrigado pela oportunidade de pesquisa e ajuda nessa trajetória.

"The world of Hell is a state in which life is itself painful; where anything you see makes you feel miserable. People in this state need someone, anyone, to be at their side. They need someone who will be there for them, to listen and offer even just a few words of encouragement. That's all it may take for the flame of life to burn brightly once again in the heart of someone who is suffering deeply. Just knowing that someone cares about them makes their heart expand."

(Daisaku Ikeda)

RESUMO

O transtorno do espectro autista (TEA) afeta o desenvolvimento cognitivo e as habilidades de comunicação em indivíduos de todas as idades, limitando sua capacidade de interação social, comunicação, seu comportamento e fala. A alta demanda por acompanhamentos terapêuticos gera uma grande quantidade de dados qualitativos e quantitativos sobre a evolução do paciente, dados que muitas vezes são despercebidos em meio a análise manual por parte dos terapeutas. Essa análise acontece durante a sessão de terapia, onde o profissional da saúde divide a atenção entre o paciente e suas anotações, ou acontece após a seção, revendo-a caso a mesma tenha sido gravada em vídeo. O problema dessa última alternativa, apesar de possibilitar uma efetiva interação com o paciente, é o tempo necessário para encontrar e rever todos os momentos importantes no vídeo, que pode ser muito mais que a própria duração da sessão. Dessa forma, a análise manual traz um déficit e um atraso no tratamento e evolução do paciente com TEA. O diagnóstico e tratamento do TEA auxiliados por técnicas computacionais representam um poderoso aliado, reduzindo a carga de trabalho dos profissionais e permitindo uma melhor experiência terapêutica para o paciente. Esse trabalho investiga como técnicas de aprendizado de máquina e visão computacional podem ajudar os especialistas ao fornecer uma análise automatizada de sessões de terapia gravadas em vídeo com crianças dentro do espectro do autismo. Para isso, propomos uma ferramenta capaz de processar grandes quantidades de dados de vídeo, filtrando de forma automatizada quadros relevantes de acordo com eventos de interesse pré-estabelecidos por profissionais da área. Utilizamos um conjunto de dados de 819 quadros de vídeos, disponibilizado pelo Instituto Italiano de Ciências e Tecnologias Cognitivas (ISTC), que apresentam três integrantes, um terapeuta, uma criança (paciente) e um ursinho interativo chamado *PlusMe*. Com o objetivo de detectar eventos de interesse baseados em interação por meio de toque, utilizamos esse conjunto para a criação de um modelo de detecção de objetos capaz de gerar caixas delimitadoras ao redor dos três integrantes de forma automática, que por sua vez são processadas por uma heurística de sobreposição de caixas delimitadoras fazendo predições sobre a existência de uma interação. Validamos essa heurística ao compararmos as predições feitas com os eventos reais presentes nos 200 quadros de vídeo disponíveis do conjunto de teste. Nossos resultados mostram que nesses 200 quadros a detecção de eventos de interesse é capaz de reduzir análises manuais entre 9 a 90% do tempo total do vídeo, considerando um equilíbrio entre redução de tempo de análise, nível de interação e desempenho de detecção, que pode trazer uma redução significativa da carga de trabalho para os especialistas em saúde. Este trabalho também apresenta a implementação da ferramenta proposta em uma plataforma web, permitindo que terapeutas e pesquisadores organizem e mantenham um repositório de sessões, onde é possível o manuseio dos vídeos, geração de estatísticas e histórico evolutivo de cada paciente ao longo das terapias, buscando trazer uma nova perspectiva em decisões clínicas e promover uma melhor experiência para o paciente.

Palavras-chave: transtorno do espectro autista. aprendizado de máquina. visão computacional. terapia. ferramenta web.

ABSTRACT

Autism Spectrum Disorder (ASD) has an impact on the cognitive development and communication skills of individuals across all age groups, resulting in challenges in social interaction, communication, behavior, and speech. The high demand for therapeutic accompaniments generates a large amount of qualitative and quantitative data on the patient's progress, data that often go unnoticed in the midst of manual analysis by therapists. This analysis takes place during the therapy session, where the health professional divides attention between the patient and their notes, or it takes place after the session, reviewing it if it has been recorded on video. The problem with the last alternative, despite allowing an effective interaction with the patient, is the time needed to find and review all the important moments in the video, which can be much longer than the duration of the session itself. Thus, manual analysis brings a deficit and delay in the treatment and evolution of patients with ASD. The diagnosis and treatment of ASD assisted by computational techniques represent a powerful ally, reducing the workload of professionals, allowing a better therapeutic experience for the patient. This work investigates how machine learning and computer vision techniques can help experts in the field by providing automated analysis of video recorded therapy sessions with children on the autism spectrum. For this, we propose a tool capable of processing large amounts of video data, automatically filtering relevant frames according to events of interest pre-established by professionals in the field. We used a dataset of 819 video frames, provided by the Italian Institute of Cognitive Sciences and Technologies (ISTC), which features three members, a therapist, a child (patient) and an interactive teddy bear named *PlusMe*. In order to detect events of interest based on interaction through touch, we used this dataset to create an object detection model capable of automatically generating bounding boxes around the three members, which in turn are processed by a heuristic of overlapping bounding boxes making predictions about the existence of an interaction. We validate this heuristic by comparing the predictions made with the actual events present in 200 video frames available in the test set. Our results show that in these 200 frames, the detection of events of interest is capable of reducing manual analysis between 9 to 90% of the total video lenght, considering a balance between analysis time reduction, interaction level and detection performance, that can bring a significant reduction in the workload for health experts. This work also presents the implementation of the proposed tool on a web platform, allowing therapists and researchers to organize and maintain a repository of sessions, where it is possible the handling of videos, generation of statistics and progress history of each patient throughout therapies, seeking to bring a new perspective on clinical decisions and promote a better patient experience.

Keywords: autism spectrum disorder. machine learning. computer vision. therapy. web app.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O transtorno do espectro autista (TEA) é uma condição de desenvolvimento complexa que afeta as habilidades de comunicação e aprendizagem. O diagnóstico de autismo em crianças tem aumentado globalmente, trazendo desafios para a área clínica, familiar e social (KOŁAKOWSKA *et al.*, 2017). Pessoas com TEA possuem dificuldades de comunicação e interação social, déficit cognitivo, diminuição da velocidade de processamento mental, aprendizado verbal e memória, e diversos outros problemas (MAENNER *et al.*, 2021).

O aumento dos casos, junto com a necessidade de auxílio e suporte ao longo da vida que grande parte dos indivíduos com TEA necessita em vários aspectos cotidianos, faz com que esse transtorno seja de grande importância para a sociedade.

Diagnosticar o TEA o quanto antes é fundamental, para que se comece desde cedo, a realização de intervenções terapêuticas que melhorem a funcionalidade social e cognitiva de indivíduos dentro do espectro (RAMIREZ-DUQUE; FRIZERA-NETO; BASTOS, 2018). Segundo Sharma, Gonda e Tarazi (2018), existem vários tipos de terapias para o TEA dentro de duas divisões principais, farmacológica e não farmacológica. No primeiro caso temos como exemplos os acompanhamentos realizados com medicamentos, ervas medicinais e técnicas nutricionais. Para o segundo caso, alguns exemplos são musicoterapia, terapia cognitivo-comportamental e terapia social comportamental.

O acompanhamento terapêutico é de suma importância pois promove melhoras na comunicação e interação social, trabalha a coordenação motora e desenvolve autoconfiança (SHARMA; GONDA; TARAZI, 2018). A validação desses resultados se dá geralmente com a presença do médico em sessões de terapia cotidianas, onde o mesmo tem a liberdade para anotar o avanço do paciente conforme lhe convém e propor novos meios e técnicas de tratamento.

Porém, essa forma tradicional de obter informações comportamentais por meio de observação direta e manual representa um atraso na evolução do tratamento. Muitas informações são sutis demais para serem percebidas pelo terapeuta durante uma sessão (CABIBIHAN *et al.*, 2013), assim como muitos dados deixam de ser coletados pela não existência de um padrão de registro consistente de profissional para profissional.

O desenvolvimento de pessoas dentro do espectro envolve sessões de terapia acompanhadas por um profissional qualificado, muitas das quais são gravadas para posterior análise. Novas metodologias de análise observacional permitem que os pesquisadores obtenham dados quantitativos a partir de gravações e utilizem abordagens computacionais avançadas para uma observação sistemática do comportamento. Essas técnicas incluem Modelos Ocultos de Markov (HMM), Redes Neurais Artificiais (RNA), abordagens espaço-temporais e outras, conforme mencionado por Nigam, Singh e Misra (2018) em seu estudo.

O uso de visão computacional pode auxiliar na automatização da análise de dados

em sessões de terapia gravadas, proporcionando uma redução no tempo de avaliação manual para os profissionais de saúde mental, além de permitir a detecção de mudanças comportamentais e a evolução dos pacientes ao longo das sessões de terapia, proporcionando-os melhores intervenções terapêuticas.

Por consequência disso, muitas das técnicas adotadas em trabalhos recentes se baseiam na visão computacional, pois ela tem ajudado esses estudos observacionais de diversas maneiras (THEVENOT; LÓPEZ; HADID, 2018). Como citado por Ribeiro, Soares, Carvalho e Grellert (2022), alguns destes estudos incluem classificação e segmentação de imagens (WU; LIU; LIU, 2019), detecção de objetos (ZHAO *et al.*, 2019), análise de vídeo (OPREA *et al.*, 2020), análise de expressão facial (ABDULLAH; ABDULAZEEZ, 2021) e detecção de padrões comportamentais (LU; NGUYEN; YAN, 2020) por exemplo.

A proposta deste trabalho aplica visão computacional para detectar interações entre os participantes, também chamados de atores, em sessões de terapia. E tem como motivação o auxílio a profissionais de saúde mental através da redução do tempo necessário para analisar e perceber mudanças entre as sessões de terapia, proporcionando aos pacientes com TEA intervenções terapêuticas mais focadas e um melhor tratamento.

O projeto envolve múltiplas instituições de pesquisa nacionais e internacionais especialistas em TEA e Inteligência Artificial. A rede de pesquisa é formada por pesquisadores da Universidade Federal de Santa Catarina (UFSC), Universidade Federal do Rio Grande do Sul (UFRGS), Universidade Católica de Pelotas (UCPEL), Universidade de São Paulo (USP), Centro de Excelência e Inovação em Autismo - Instituto Farol, e pelo Instituto de Ciências e Tecnologias Cognitivas (ISTC) do Conselho Nacional de Pesquisa Italiano (CNR).

1.1 OBJETIVOS

O objetivo deste trabalho foi investigar como técnicas de aprendizado de máquina, visão computacional e mineração de dados podem ser utilizadas no auxílio em terapias de pacientes com transtorno do espectro autista (TEA), possibilitando um acompanhamento mais detalhado do processo evolutivo dos pacientes. Espera-se, no final desse projeto, fornecer uma ferramenta a pesquisadores, terapeutas e especialistas que ajude na identificação automatizada de características e eventos de interesse nos dados, utilizando técnicas de inteligência artificial desenvolvidas no contexto deste projeto.

1.1.1 Objetivos Específicos

Os objetivos específicos deste trabalho são:

1. Coleta de vídeos de sessões de terapia do TEA.
2. Rotular o posicionamento dos atores em cada quadro de vídeo por meio de caixas

delimitadoras para a construção de um conjunto de dados que permita o treinamento de modelos de aprendizado de máquina, assim como validação e teste.

3. Treinamento e otimização de modelos de detecção dos atores.

4. Implementação de heurística de sobreposição de caixas delimitadoras para predição de eventos de interesse.

5. Avaliação de desempenho de modelos de aprendizado de máquina e do sistema com heurística.

6. Criação de uma plataforma web que permita que o usuário organize e mantenha um repositório de gravações de sessões de terapia, onde será possível o manuseio dos vídeos, geração de estatísticas, dashboards e histórico evolutivo de cada paciente.

1.2 ORGANIZAÇÃO DO TRABALHO

A seguir, o capítulo 2 *(Fundamentação Teórica e Revisão da Literatura)* fornece uma base teórica sobre o diagnóstico e terapia do TEA e os problemas da avaliação manual atrelados às técnicas de observação corriqueiras, além de apresentar como técnicas de aprendizado de máquina podem ajudar. O capítulo 3 *(Solução Proposta)* apresenta o conjunto de dados e abordagem utilizados em todo o trabalho. O capítulo 4 *(Resultados)* apresenta a sequência de passos tomados e os resultados obtidos envolvendo treinamento de modelos de aprendizado de máquina, detecção de eventos e métricas avaliativas. O capitulo 5 *(Construção da Ferramenta)* mostra a implementação da ferramenta em uma plataforma web e os resultados na interface de usuário. O capítulo 6 *(Conclusão)* retoma os principais resultados e apresenta ideias para trabalhos futuros, e por fim, o capítulo 7 *(Considerações Finais)* traz considerações sobre todo o desenvolvimento deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA E REVISÃO DA LITERATURA

Este capítulo traz uma revisão teórica sobre o TEA, sua definição e abrangência. É apresentado também como terapias observacionais ajudam na coleta de dados clínicos e como técnicas de visão computacional e aprendizado de máquina auxiliam o trabalho do terapeuta ao automatizar o processo de extração e processamento desses dados.

2.1 TRANSTORNO DO ESPECTRO AUTISTA

De acordo com o Manual Diagnóstico e Estatístico de Transtornos Mentais, 5ª Edição (DSM-5) (EDITION *et al.*, 2013), o, transtorno do espectro autista (TEA) é uma condição que afeta a maneira como uma pessoa percebe, interage e se comunica com o mundo ao seu redor. É um distúrbio do neurodesenvolvimento que pode apresentar uma ampla gama de características e sintomas individuais. Essas características podem incluir dificuldades na comunicação e interação social, padrões de comportamento repetitivos, e sensibilidades sensoriais (MAENNER *et al.*, 2021).

O TEA engloba uma ampla gama de sintomas que podem variar em intensidade, habilidades e características de pessoa para pessoa. Essas características tornam o diagnóstico e a avaliação do progresso terapêutico desafiadores, mas também são fundamentais para determinar a eficácia das intervenções (KOŁAKOWSKA *et al.*, 2017). Estudos indicam que o TEA está associado a diversas comorbidades, como epilepsia, problemas de atenção, distúrbios gastrointestinais, comportamento opositivo, ansiedade, depressão, distúrbios do sono e distúrbios alimentares. Estima-se que entre 30 e 50% das pessoas com TEA também apresentam algum grau de deficiência intelectual (ROGGE; JANSSEN, 2019).

2.1.1 Eventos de Interesse em Terapias

Como abordado nos resultados preliminares de Ribeiro, Soares, Carvalho e Grellert (2022), o sucesso de uma terapia está diretamente ligado ao feedback que o paciente oferece, principalmente em terapias observacionais (BERTAMINI *et al.*, 2021) onde paciente é acompanhado ao vivo pelo terapeuta. Em sessões não gravadas, o feedback é limitado à percepção imediata e à memória do terapeuta, o que pode levar a interpretações errôneas e ao esquecimento. No entanto, ao gravar uma sessão em vídeo, o terapeuta tem a oportunidade de revisitar eventos de interesse percebidos durante a sessão ao vivo, bem como perceber novos eventos previamente não detectados, gerando assim uma espécie de registro (HAILPERN *et al.*, 2009).

Nesse registro, cada terapeuta é livre para anotar os eventos de interesse que mais fazem sentido para si como um profissional do TEA, eventos qualitativos e/ou quantitativos, como por exemplo, informação se a criança teve uma boa evolução em

relação às sessões passadas, ou quantas vezes ela interagiu com o terapeuta e o ambiente, quanto tempo durou a sessão, entre muitos outros.

No entanto, dois problemas interessantes surgem nesse cenário, sendo o primeiro o tempo gasto nessa análise manual. Revisar e anotar uma sessão de terapia pode levar mais tempo do que a própria sessão, o que acaba sendo inviável para uma grande quantidade de dados e/ou quando o terapeuta responsável atende outros pacientes. E se os dados não forem anotados pelo mesmo terapeuta que realizou a sessão, o problema é o padrão de registro. Para um profissional, certos eventos de interesse são mais importantes e descritivos do que para outros, então o tipo, formato e métricas de cada registro rapidamente diferem e se tornam inconsistentes de profissional para profissional.

2.1.2 Automação da Coleta e Processamento de Dados em Terapias

Nos últimos anos, diversas estratégias têm sido propostas para aprimorar diagnósticos e terapias que utilizam métodos de aprendizado de máquina e visão computacional (PARIKH; LI; HE, 2019; NOGAY; ADELI, 2020; KOŁAKOWSKA *et al.*, 2017; RAMIREZ-DUQUE; FRIZERA-NETO; BASTOS, 2018).

O aumento do número de diagnósticos do TEA e a grande quantidade de dados gerados pelos tratamentos relacionados a essa condição estimulam o interesse em pesquisas desse tipo, pois com o uso dessas tecnologias digitais e a abundância de dados disponíveis é possível automatizar, pelo menos parcialmente, a análise desses dados, visando aprimorar tanto o diagnóstico quanto as terapias relacionadas ao TEA.

O primeiro trabalho citado, de Parikh, Li e He (2019), propõe a realização de diagnóstico do TEA a partir de atributos retirados de características pessoais, como idade, sexo, lateralidade (destro ou canhoto) e índices de *QI*. Diferente de Nogay e Adeli (2020), que traz um compilado de estudos que extraem atributos a partir de imagens de ressonância magnética cerebrais. Ambas as abordagens fazem uso de classificadores baseados em machine learning e demonstram sucesso em diferenciar aspectos característicos e não característicos do TEA.

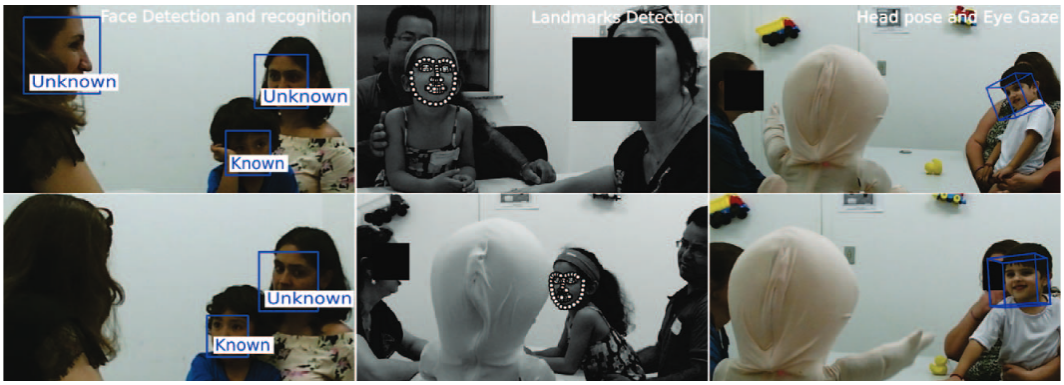
O terceiro trabalho, de Kołakowska *et al.* (2017), propõe uma ferramenta para o acompanhamento terapêutico de crianças com TEA, baseado na coleta de dados provenientes de sensores em *tablets* com diversos jogos que as crianças podem brincar. A ideia desse estudo é recolher, dentro da aplicação, com o acelerômetro, giroscópio e outros sensores do próprio tablet, o nível de indicadores comportamentais característicos de crianças atípicas, como alta força de impacto da tela, movimentações laterais, repetição de movimentos, e usar classificadores de aprendizado de máquina que analisam os parâmetros de sessões anteriores com os da sessão atual para prever progresso ou não. E mostrou que é possível acompanhar, em algumas áreas do desenvolvimento, o progresso da criança com TEA entre sessões.

O último trabalho, de Ramirez-Duque, Frizera-Neto e Bastos (2018), utiliza vi-

são computacional e machine learning em intervenções terapêuticas auxiliadas por robô para diagnosticar o TEA. Nele, o conjunto de dados usados são vídeos de intervenções terapêuticas onde a criança interage com o terapeuta e com um robô. Modelos de visão computacional estimam a posição e movimentação da cabeça e dos olhos assim como a atenção conjunta (*joint attention*), e com esse histórico de movimentação e o pareamento com situações pré estabelecidas, pode-se classificar a criança como neuroatípica ou não.

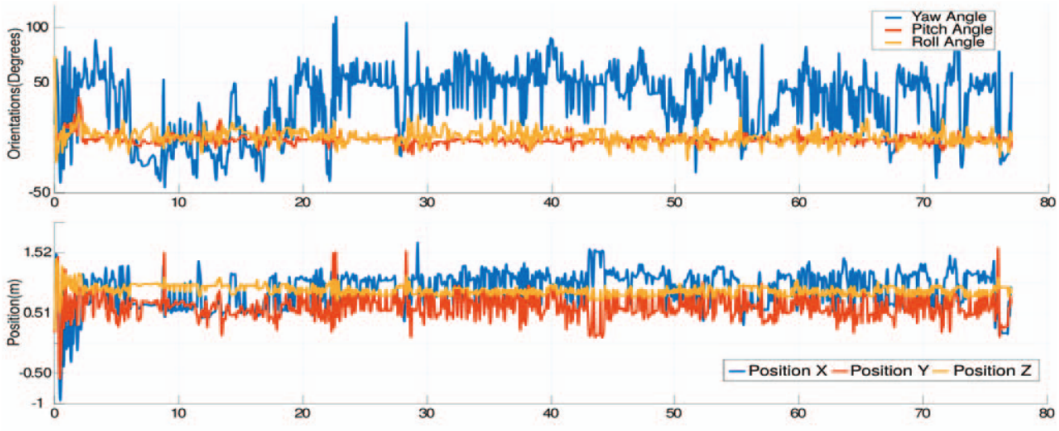
As Figuras 1 e 2 mostram o ambiente de intervenção e resultados de movimentação da cabeça da criança.

Figura 1 – Intervenção Terapêutica para Diagnóstico com Robô.



Fonte: Ramirez-Duque, Frizera-Neto e Bastos (2018).

Figura 2 – Resultados para Movimentação da Cabeça em Intervenção com Robô.



Fonte: Ramirez-Duque, Frizera-Neto e Bastos (2018).

Dentro os estudos citados, o que mais se aproxima dos nossos objetivos é na verdade uma mescla entre o trabalho de acompanhamento do progresso terapêutico, Kołakowska *et al.* (2017), e do de intervenção robótica, Ramirez-Duque, Frizera-Neto e Bastos (2018). Uma vez que objetivamos detectar eventos de interesse de forma automatizada durante sessões de terapia que envolvem um ursinho de pelúcia robótico, gerando estatísticas que elucidem o progresso do paciente.

2.2 APRENDIZADO DE MÁQUINA

Aprendizado de máquina, ou Machine Learning (ML) em inglês, é uma técnica computacional e estatística que permite que modelos computacionais "aprendam" diversas tarefas sem que suas etapas de decisão tenham sido explicitamente programadas. Isso é realizado por meio da adequação desses modelos em dados disponíveis sobre o objetivo, permitindo que realizem predições com base no que aprenderam (BI *et al.*, 2019a).

Esses modelos são utilizados em diversas áreas de pesquisa e de produção, algumas de suas aplicações incluem a detecção de spam, reconhecimento de imagens, vídeos e fala, diagnósticos médicos e controle de robôs (JORDAN; MITCHELL, 2015).

2.2.1 Processos de Aprendizado

O processo de aprendizado desses modelos, também chamado de treinamento, se dá de três maneiras principais, de forma supervisionada, não supervisionada e por reforço (AYODELE, 2010).

No treinamento supervisionado, o modelo recebe como entrada um par de informações que corresponde ao dado em si e a anotação do dado, onde esses pares formam o conjunto de treinamento (dado + objetivo). A cada iteração, o modelo faz uma predição que tenta relacionar o dado com sua anotação, que pode ser um valor numérico como tarefas de regressão ou um rótulo, como em tarefas de classificação. Caso essa predição esteja errada, o modelo ajusta seus parâmetros com base numa *função perda* que tem o objetivo de reduzir esse erro, para que na próxima iteração possa tentar de novo (um modelo com bom desempenho tende a ter o erro diminuindo a cada iteração). O termo supervisionado se origina dessa frequente comparação que é feita entre a anotação real e a anotação prevista mais o ajuste de parâmetros. Um exemplo de modelo assim é o regressor linear que tenta adequar uma função tendência a um conjunto de dados.

O treinamento não supervisionado é utilizado na descoberta de padrões e estruturas nos dados sem anotações pré-definidas e informações do objetivo final, ou seja, sem orientação externa. Isso traz um dificuldade extra no treinamento dos modelos pois não é definido no modelo uma função perda que em todo o treinamento é comparada com um valor real. Exemplos desse tipo de treinamento são os algoritmos de *K-Means Clustering* (SINAGA; YANG, 2020), Redução de Dimensionalidade (CARREIRA-PERPINÁN, 1997) e Modelos Generativos (GM *et al.*, 2020), extremamente relevantes no cenário atual como ChatGPT[1], DALL-E[2] e outros.

O treinamento por reforço é uma técnica muito usada na formação de agentes inteligentes, que precisam saber reagir a mudanças no estado em que se encontram por meio de "recompensas" que são atualizadas conforme suas decisões. Algumas aplicações

[1] https://chat.openai.com/
[2] https://labs.openai.com/

são carros autônomos, processamento de linguagem natural e robótica (POLYDOROS; NALPANTIDIS, 2017; LUKETINA *et al.*, 2019).

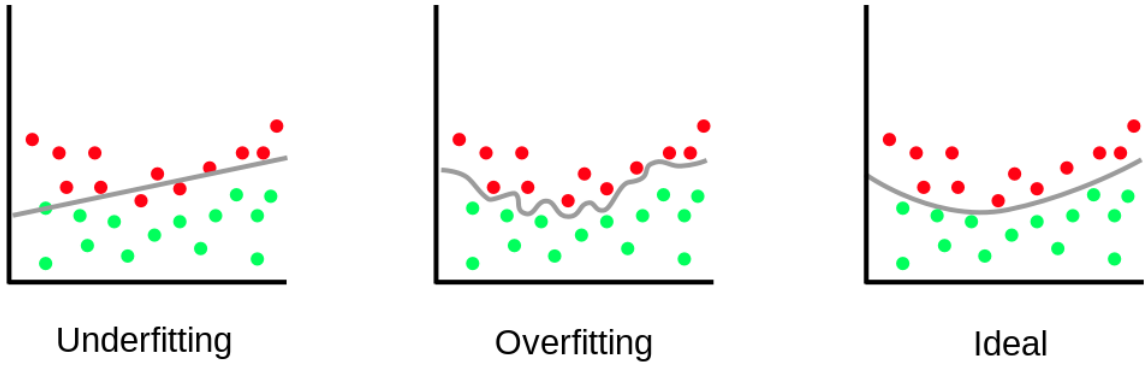
2.2.2 Overfitting e Underfitting

Como mencionado, durante o treinamento supervisionado, o modelo aprende ao relacionar os dados de entrada com suas anotações, fazendo ajustes conforme necessário. Uma abordagem ingênua consiste em utilizar os mesmos dados do treinamento para a avaliação do modelo. Essa abordagem traz uma desvantagem, ela não permite avaliar como o modelo irá generalizar, i.e., como ele irá se comportar com dados diferentes daqueles utilizados no seu treinamento. É possível que o modelo tenha se especializado demais para minimizar a função de erro, reduzindo sua capacidade de generalização. Esse fenômeno indesejado é chamado de *overfitting*. Uma analogia interessante para entender esse fenômeno consiste em dizer que o modelo apenas "decorou" as respostas ao invés de realmente entender a essência do problema sendo modelado.

Por outro lado, quando um modelo não consegue se adaptar nem ao próprio conjunto apresentado, é dito que houve *underfitting*. Esse tipo de situação é causada quando os dados utilizados não possuem informações relevantes para modelar o problema em questão ou quando o modelo possui uma capacidade de representação limitada para lidar com a complexidade dos dados, não conseguindo discriminar de forma satisfatória as características essenciais do conjunto.

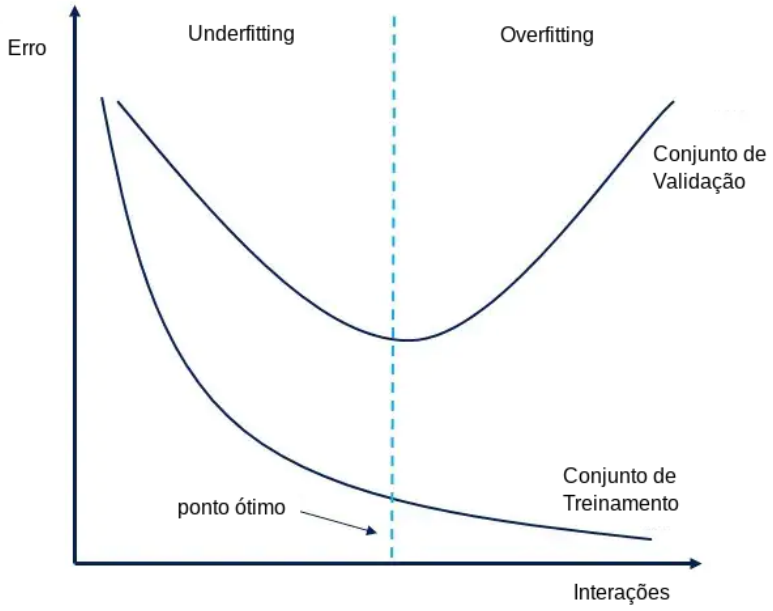
A Figura 3 ilustra essas duas ocorrências juntamente com o estado *ideal*, onde o modelo aprende a fazer uma divisão "equilibrada" (entre os grupos verde e vermelho) que não se adapta exageradamente. Já a Figura 4 ilustra como podemos identificar esses 3 momentos ao olharmos para a perda (erro) no conjunto de treinamento e validação. Onde o conjunto de validação é um conjunto a parte, não visto pelo modelo mas que compartilha de características semelhantes ao conjunto de treinamento, onde é possível realizar a análise de convergência conjunta ou divergência das perdas (funções perda/erros) desses dois conjuntos (XU; GOODACRE, 2018). Uma perda de validação muito próxima da de treinamento (convergência conjunta), indica generalização, o que é o estado ideal. Uma perda de validação estagnada ou divergente da de treinamento significa overfitting. Já uma perda de treinamento alta, indica que o modelo está tendo underfitting.

Figura 3 – Ilustração - Underfitting, Overfitting e Ideal.



Fonte: Adaptado de Lin (2020).

Figura 4 – Ilustração - Indicação de Overfitting e Underfitting nas Perdas.

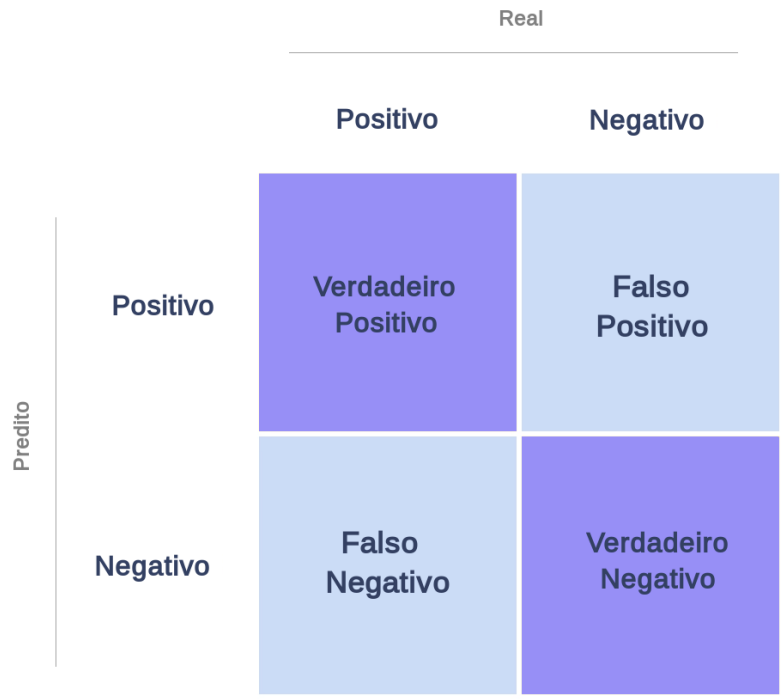


Fonte: Adaptado de Chatterjee (2022).

2.2.3 Métricas Avaliativas

Além de analisarmos o desempenho do modelo olhando para as perdas como mostra a Figura 4, podemos analisar métricas que sejam bem descritíveis do problema em si. Algumas das métricas mais comuns, são a Acurácia, Precisão, Revocação (*Recall* em inglês), *F1Score*, *AP* e *mAP* (HOSSIN; SULAIMAN, 2015; DALIANIS, 2018; PADILLA; NETTO; DA SILVA, 2020). Onde todas essas métricas podem ser obtidas de uma matriz de confusão (Figura 5).

Figura 5 – Exemplo de uma Matriz de Confusão.



Fonte: O Autor.

- Acurácia: Representa a proporção de previsões corretas no total.

  $\text{Acc} = \frac{VP+VN}{VP+VN+FP+FN}$

- Precisão: Representa a proporção de acerto da classe positiva dentre todas as predições positivas. Um modelo preciso tem sucesso em acertar várias de suas predições positivas (dentre todos os positivos).

  $\text{Pre} = \frac{VP}{VP+FP}$

- *Recall*: Representa a proporção de acerto da classe positiva dentre todas as classes positivas. Um modelo com alta Recall é "conservador" e acerta grande parte do que deveria ser acertado (entre os positivos).

$$\text{Rec} = \frac{VP}{VP+FN}$$

- F1Score: É definido como a média harmônica entre Precisão e Recall. Sendo um indicador **pontual** do equilíbrio dessas métricas.

  $$\text{F1} = \frac{2*Precisão*Recall}{Precisão+Recall} = \frac{2*VP}{2*VP+FP+FN}$$

- AP (Average Precision): A AP de uma classe é definida como a área sob a curva de Precisão e Recall (curva PR) dessa classe. Sendo um indicativo do quão equilibrado o preditor é em ser preciso e "conservador" **como um todo**, pois a curva PR é traçada para vários limiares de confiança entre 0 e 1.

  $$\text{AP} = \sum_n (R_n - R_{n-1}) P_n$$

- mAP (mean Average Precision): É definida como a média da AP calculado para todas as classes.

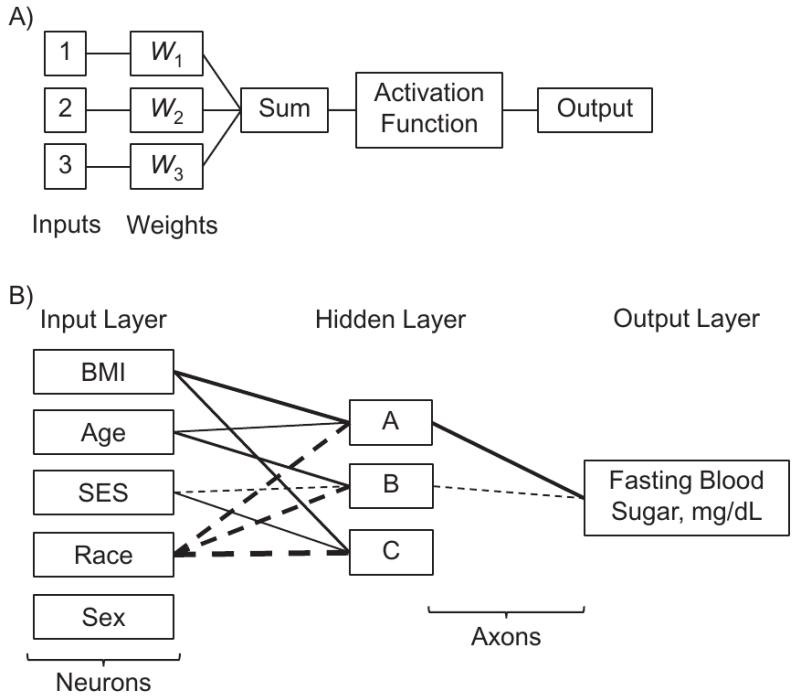
  $$\text{mAP} = \frac{1}{n} \sum_n AP_n$$

Cada uma dessas métricas, e muitas outras que não foram mencionadas, apresentam prós e contras que devem ser considerados durante a análise do desempenho do modelo. Como por exemplo, quando ocorre desbalanceamento do conjunto de dados. Basta imaginarmos um conjunto hipotético de 100 imagens, 10 imagens da classe A e 90 da classe B. Se o modelo prever que todas as imagens são da classe B, terá uma acurácia de 90% com um desempenho péssimo na classe A, dessa forma, dando uma impressão equivocada de quão bom o modelo realmente é em diferenciar essas classes.

2.3 APRENDIZADO PROFUNDO

Aprendizado profundo é uma sub categoria de Aprendizado de Máquina que explora a intricada conexão das Redes Neurais Artificiais. Essas redes são compostas por neurônios artificias, unidades de cálculos matemáticos que se ligam entre si e transmitem informação desde a entrada até o final do modelo (LECUN; BENGIO; HINTON, 2015). A Figura 6 mostra a construção básica de um neurônio, também conhecido como *perceptron* no item *A)*, que recebe as entradas em seus "axônios" (neste exemplo são 3 entradas). Essas entradas são multiplicadas por pesos, é feito um somatório dessas multiplicações e logo passado por uma função ativadora gerando o resultado (*output*). No item *B)* dessa mesma figura temos a demonstração de uma rede neural com 3 neurônios internos e 5 entradas que simbolizam valores demográficos. Cada neurônio interno faz seu cálculo individual e no final temos um resultado referente ao cálculo conjunto deles.

Figura 6 – Perceptron e Rede Neural Simples.



Fonte: Bi *et al.* (2019b).

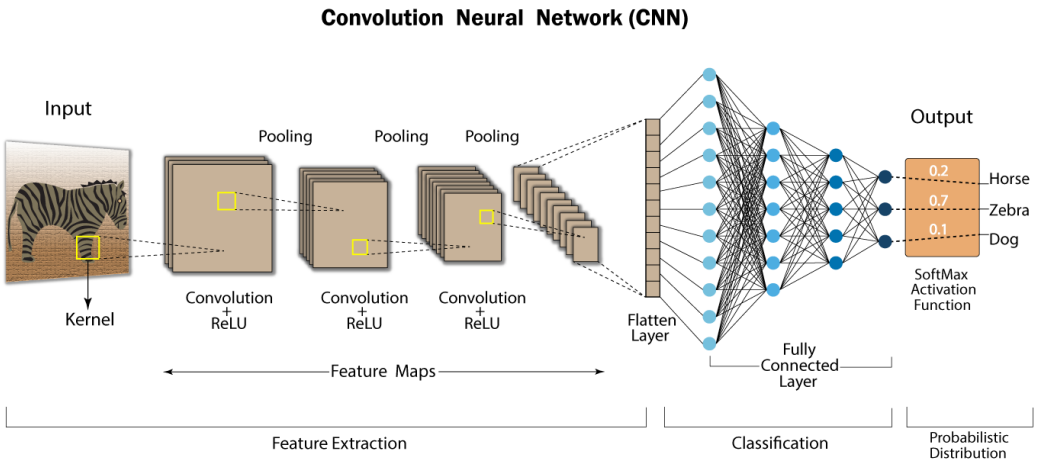
2.3.1 Redes Neurais Convolucionais

LeCun, Bengio e Hinton (2015) definem as Redes Neurais Convolucionais como Redes Neurais Profundas especializadas em processamento de imagens. Elas apresentam, logo após a entrada, uma sequência de camadas de filtros convolucionais (além de amostragem e ativação) que extraem atributos espaciais representativos do conteúdo da imagem, aumentando o nível de abstração em cada camada. A Figura 7 mostra a estrutura típica de uma rede convolucional. A rede ilustrada nessa figura é uma rede de classificação de imagens de animais, sua entrada é uma imagem (vetor de 2 ou mais dimensões) e sua saída são três valores numéricos que indicam a probabilidade de a imagem na entrada ser um cavalo, uma zebra ou um cachorro. Depois da última camada convolucional, seguem as ligações de uma rede neural artificial para a tarefa de classificação chegando ao final com o resultado desejado. E a Figura 8 ilustra os níveis de abstração em cada camada convolucional por meio do resultado dos filtros espaciais.

2.3.2 Usos das Redes Convolucionais

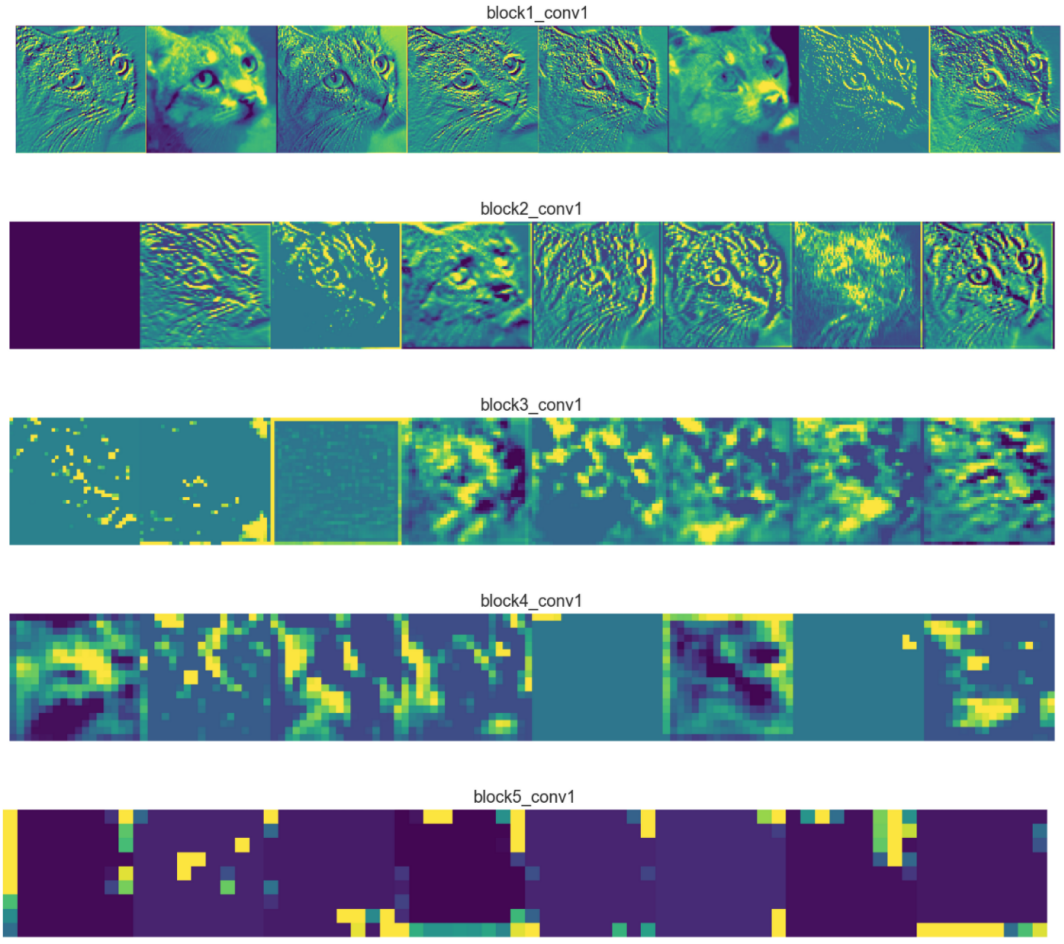
Como mencionado, uma das aplicações dessas redes na tarefa de visão computacional é a classificação do conteúdo de uma imagem, mas elas também são aplicadas em tarefas de localização, detecção e segmentação. Na classificação, como no exemplo da Figura 7, o resultado é a predição da classe presente. Na detecção, temos duas tarefas conjuntas, a classificação e determinação de localização por meio de caixas delimitadoras.

Figura 7 – Estrutura Típica de uma Rede Neural Convolucional.



Fonte: E (2020).

Figura 8 – Evolução dos Atributos Filtrados pelas Camadas Convolucionais.



Fonte: Dertat (2017).

E na segmentação temos a classificação de cada pixel do objeto limitado à sua silhueta.

Figura 9 – Tarefas Típicas de Redes Convolucionais.

Fonte: Adaptado de Patel (2020).

2.3.3 Transfer Learning

Quando os modelos desenvolvidos não alcançam um bom desempenho, seja por consequência de um dataset desbalanceado, baixa variedade dos dados ou poucas amostras para treino, pode se utilizar vários métodos para tentar contornar isso. Porém, um jeito rápido e muito bem estabelecido é o uso do Transfer Learning. Transfer Learning é uma técnica usada para transferir o aprendizado de um modelo treinado em outro conjunto de dados (WEISS; KHOSHGOFTAAR; WANG, 2016). Modelos treinados em competições de reconhecimento visual como a ImageNet[3] são treinados em centenas de classes e possuem um ótimo desempenho, e comumente esses modelos têm seus aprendizados reutilizados.

Uma vez que o modelo está treinado, os pesos que compõe cada uma das camadas convolucionais e das outras camadas podem ser "congelados" para uso posterior, fixando, dessa forma, o aprendizado de uma rede neural convolucional. Ao modificarmos a saída preditiva desses modelos de centenas de saídas (classes) para 3 saídas por exemplo como no exemplo da Figura 7 (cavalo, zebra e cachorro) e realizarmos um treinamento com o novo conjunto de dados, teremos uma chance de convergência muito maior e muito mais rápida. Isso acontece porque a rede neural que recebe esses pesos reusa características generalistas que a rede original aprendeu, adicionando novos aprendizados, combinando e mesclando com o que já foi aprendido, similar a como os seres humanos reutilizam informações para aprender (TORREY; SHAVLIK, 2010).

2.4 YOLO

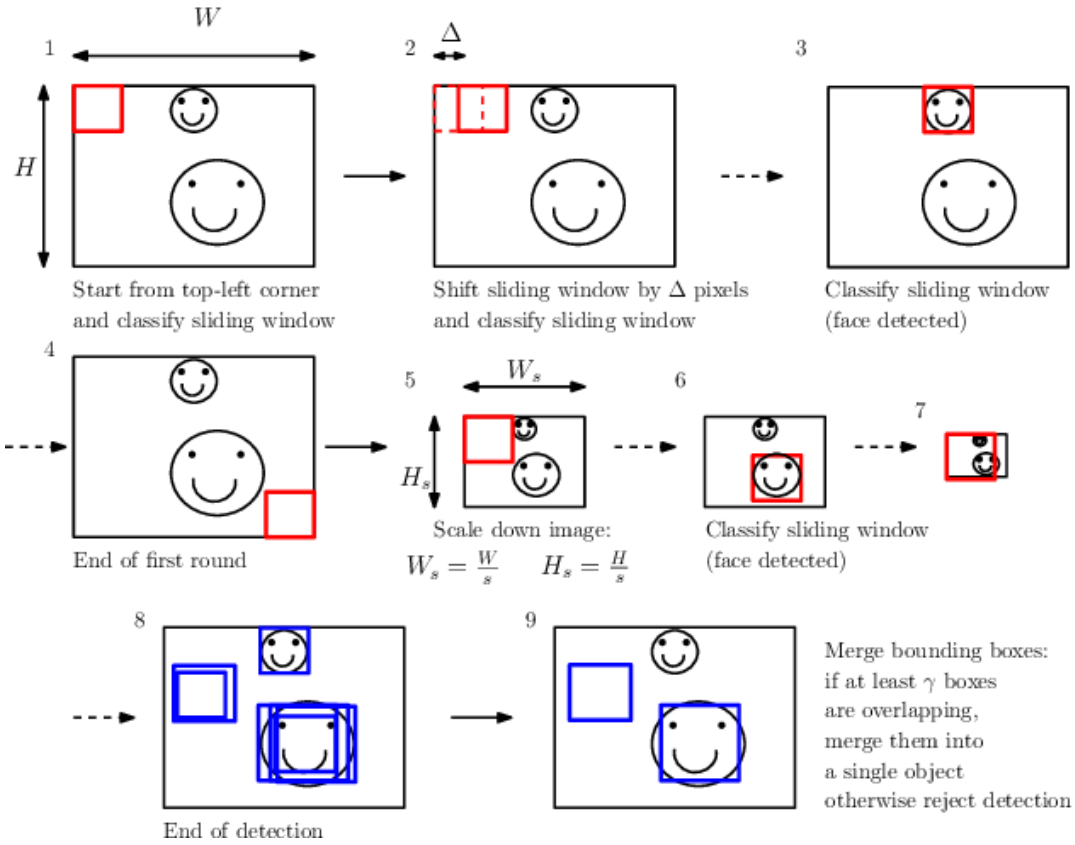
Desenvolvido por Redmon *et al.* (2016), YOLO é um algoritmo de detecção de objetos em tempo real. Tem como principal proposta o baixo custo computacional ao

[3] https://www.image-net.org/

realizar as detecções em uma "única etapa" na imagem, onde classificação e localização são feitas de uma só vez, e não em momentos diferentes como nos métodos de janelas deslizantes.

O método da janela deslizante para detecção de objetos (GOULD; GAO; KOLLER, 2009) é relativamente simples mas muito custoso. Como pode-se ver na Figura 10, uma região retangular (janela) percorre uma imagem em passos regulares, e a cada passo, uma tarefa de classificação é feita. Quando uma determinada classe é detectada, essa janela é automaticamente determinada como a caixa delimitadora, ou bounding box em inglês, desse objeto. O problema com essa técnica é que não existe um tamanho de janela que englobe as diversas dimensões de todos os objetos no conjunto de dados, sendo necessário a realização dessa etapa de classificação em toda imagem várias vezes com vários tamanhos diferentes de janelas.

Figura 10 – Ilustração da Detecção por Janela Deslizante.



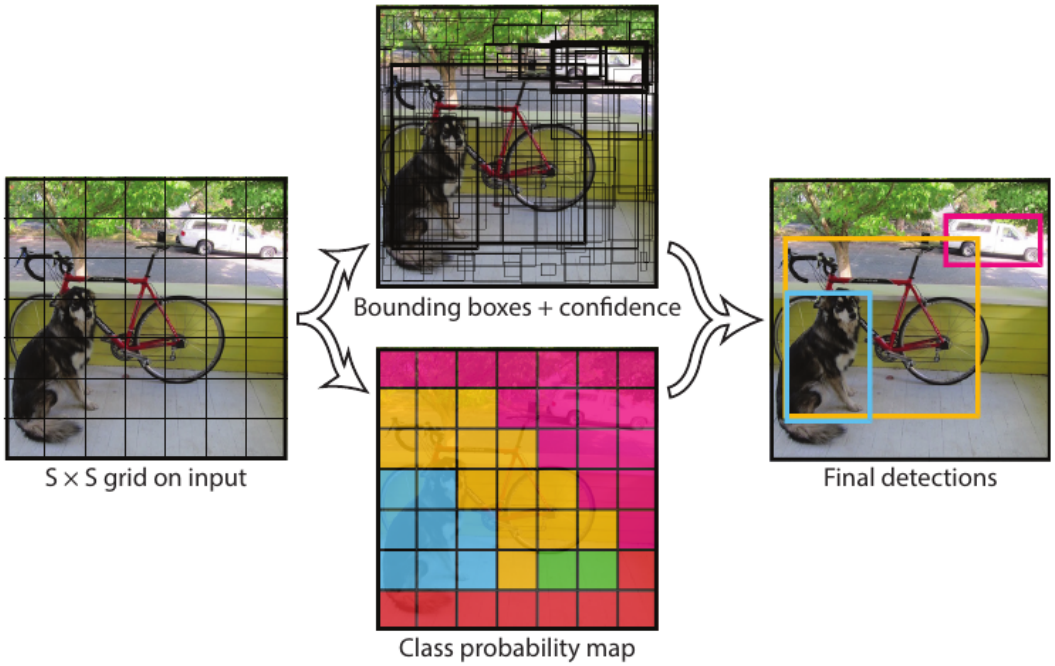
Fonte: Comaschi *et al.* (2013).

Um dos maiores contribuintes para a alta velocidade em tempo real da YOLO é o seu algoritmo que elimina repetições com vários tamanhos de janela, passando uma única vez pela imagem. A ideia, desenvolvida por Redmon *et al.* (2016), ilustrada na Figura 11, está em dividir previamente as imagens de entrada em uma grade de S x S células. Durante o processamento, é previsto para cada uma dessas células a existência de um certo tipo de objeto (dentro das C classes possíveis) com uma certa probabilidade e previsto

B caixas delimitadoras. Onde cada bounding box é agrupada como a composição de 5 valores, sendo eles as coordenadas do centro (x, y), as dimensões de largura e altura (w, h) e a probabilidade da classe.

Essas informações para cada célula, juntamente com a eliminação de predições menos relevantes com o algoritmo *non maximum supression*, fazem que essa rede de detecção tenha como saída um tensor com informações da probabilidade de classe e parâmetros da bounding box.

Figura 11 – Ilustração da Detecção pelo Algoritmo da YOLO.



Fonte: Redmon *et al.* (2016).

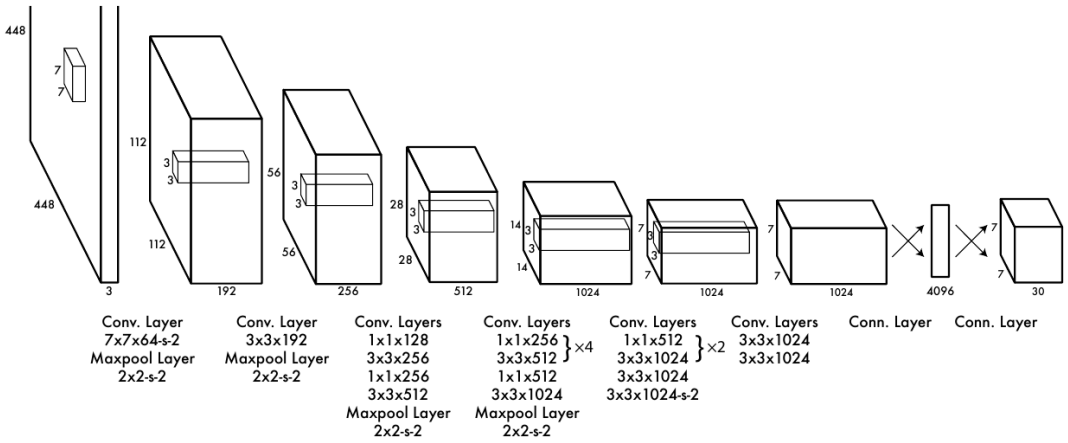
A rede de detecção é implementada com camadas de convolução para extração de atributos espaciais seguida de camadas totalmente conectadas para predição de probabilidades e coordenadas. A estrutura dessa rede pode ser vista na Figura 12.

Modelos mais atuais da YOLO implementam a função perda como uma combinação de perdas relativas a coordenadas das bounding boxes, existência de objetos e classificação dos mesmos.

- $loss_{box}$ — Perda de regressão da bounding box que utiliza erro quadrático médio. Representa quão bem o algoritmo pode encapsular um objeto com uma caixa delimitadora.
- $loss_{obj}$ — Perda de detecção de objetos que utiliza Entropia Cruzada Binária. Refere-se à probabilidade de existência de um objeto em uma região de interesse (janela) proposta.

- $loss_{cls}$ — Perda de classificação que utiliza Entropia Cruzada. Avalia a capacidade do modelo em prever corretamente a classe do objeto.

Figura 12 – Estrutura da Rede YOLO.



Fonte: Redmon *et al.* (2016).

2.5 TECNOLOGIAS WEB PARA PLATAFORMA DE INTERAÇÃO E VISUALIZAÇÃO DOS RESULTADOS

Uma vez que os resultados de detecção são gerados, é interessante que o terapeuta possa acessá-los por meio de um aplicativo ou plataforma que possibilite o armazenamento e extração dos dados de forma eficaz. Dessa forma, se o terapeuta quiser rever resultados anteriores, é necessário apenas uma pequena requisição rápida que contenha por exemplo o nome de paciente e data da sessão.

2.5.1 Bancos de Dados

Formas comuns de armazenamento de dados como planilhas eletrônicas e documentos de texto tendem a se mostrar dificultosos para mantê-los a longo prazo, seja por falta de um padrão de registro, falta de segurança ou lentidão com o aumento do número de dados (GORDON, 1999).

Para se ter a garantia de uma forma de armazenamento confiável, estável e veloz, é necessário o uso de um Sistema Gerenciador de Banco de Dados (SGBD). Este, é um software destinado a armazenamento de informações dos mais diversos tipos, com alta eficiência e diversas camadas de segurança, possibilitando que os usuários criem, alterem e excluam os dados, gerando histórico de atividades e relatórios.

Para a criação de um banco de dados estruturado que atenda os registros das detecções em sessões de terapia, pode-se usar um Sistema Gerenciador de Banco de Dados Relacional (SGBDR) que use a linguagem relacional chamada *SQL* (SILVA; ALMEIDA;

QUEIROZ, 2016), o qual funciona com operações de criação, atualização, remoção e recuperação de dados, conhecidas popularmente como *requisições CRUD*, numa sequência de tabelas relacionadas entre si que compõe o *esquema* do banco de dados.

A grande maioria dos bancos de dados relacionais são processos separados da aplicação principal, porém existe uma versão leve de SGBDRs chamado *SQLite*[4], que roda juntamente com a aplicação e é interessante para projetos de pequeno e médio porte.

2.5.2 Python

Python[5] é uma linguagem de programação de alto nível criada em 1991 por Guido van Rossum. Com uma sintaxe simples e alta abstração, é uma ótima escolha para desenvolvimento de scripts, desenvolvimento web e análise de dados (SRINATH, 2017). Além de ser referência em machine learning e deep learning com os frameworks e bibliotecas *SciPy*[6], *TensorFlow*[7] e *PyTorch*[8] (RASCHKA; PATTERSON; NOLET, 2020).

2.5.3 Streamlit

Existem vários *frameworks* para criação de interfaces locais e web, mas um bastante utilizado é o framework *Streamlit*[9], que de um jeito prático transforma scripts Python em páginas web. Com ele, a lógica do aplicativo está junto com o design da interface, o que impulsiona o desenvolvimento de aplicativos de pequeno e médio porte, e pelo fato de ser em Python, é multiplataforma, facilitando a distribuição do software final. Dessa forma, o uso dessas tecnologias em conjunto se torna viável para rápida prototipação e criação da ferramenta como um todo.

[4] https://www.sqlite.org/index.html
[5] https://www.python.org/
[6] https://scipy.org/
[7] https://www.tensorflow.org/
[8] https://pytorch.org/
[9] https://streamlit.io/

3 SOLUÇÃO PROPOSTA

Propomos um sistema capaz de detectar e classificar cada um dos integrantes presentes em sessões de terapia do Transtorno do Espectro Autista gravadas em vídeo e com a informação de suas posições detectar interações baseadas em toque (eventos).

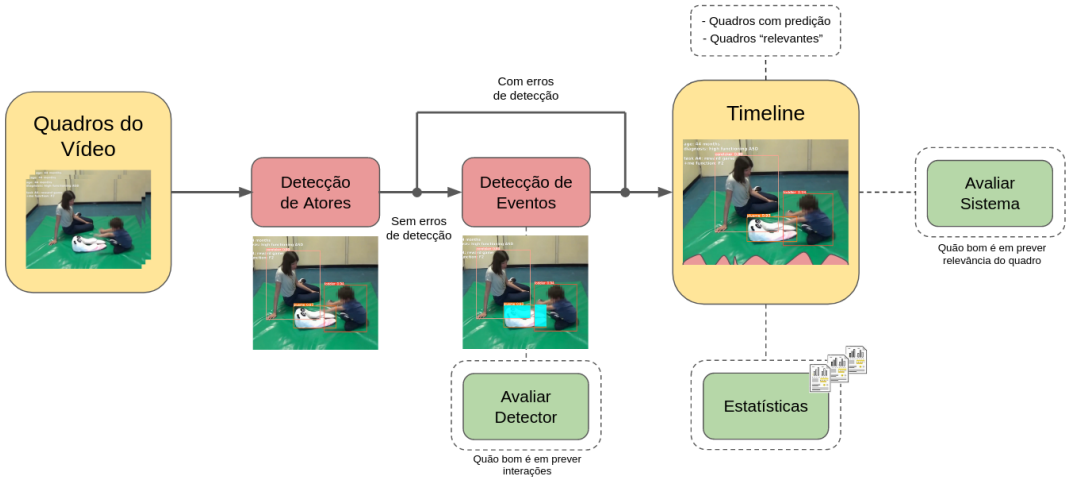
Seguindo o diagrama proposto na Figura 13, o usuário faz *upload* de um vídeo de sessão do TEA e o mesmo é processado pelo algoritmo da YOLO na etapa de **detecção de atores** gerando as caixas delimitadoras ao redor de cada um dos integrantes.

Se não há nenhum erro na detecção de atores, é feito a **detecção de eventos**, onde utilizamos uma heurística de sobreposição de caixas delimitadoras que processa as coordenadas dessas caixas e identifica o nível intersecção entre elas, fazendo com que cada um dos quadros seja predito como contendo ou não algum tipo de interação e enviado para a construção de uma timeline de eventos. E ao **avaliarmos** as predições desse **detector de eventos**, podemos verificar "o quão bom ele é em detectar essas interações".

Se há algum **erro** na etapa de **detecção de atores**, onde não existe alguma caixa delimitadora, seja pela falta da mesma ou classificação errada, **não** é feito a detecção de eventos e os quadros são enviados diretamente para a timeline como sendo quadros "relevantes", onde são identificados automaticamente como **contendo uma interação**, necessitando dessa forma que o médico confirme de forma manual na timeline final se está ou não acontecendo uma interação. E dessa forma podemos **avaliar o sistema** como um todo ao questionarmos "o quão bom ele é em prever se o quadro é ou não relevante" com base em suas predições.

Finalmente, quando toda a timeline é criada, é gerado um conjunto de **estatísticas** sobre as interações entre atores, que podem ser usadas para o acompanhamento do avanço do paciente entre sessões.

Figura 13 – Diagrama Proposto da Ferramenta.



Fonte: Adaptado de Ribeiro, Soares, Carvalho e Grellert (2022).

3.1 CONJUNTO DE DADOS

Como o foco desse trabalho está em auxiliar o profissional do TEA na tomada de decisões em sessões de terapia usando visão computacional, nada mais natural que o conjunto de dados seja composto por imagens ou frames de vídeos dessas sessões.

Um dos nossos parceiros de pesquisa, o ISTC, desenvolve um trabalho muito interessante com um ursinho de pelúcia interativo chamado *PlusMe*[1]. Esse ursinho é classificado como uma companhia "inteligente", não tão complexa quanto um robô e nem tão simples como um ursinho comum, despertando apego emocional com um design vestível que é usado para brincadeiras que emitem luzes e cores estimulando diversas competências sociais das crianças (SPERATI; ÖZCAN *et al.*, 2019a) (GIOCONDO *et al.*, 2022).

Esse time de pesquisa italiano disponibilizou em seu site um vídeo demonstrativo de 8 sessões experimentais com o PlusMe, tendo em torno de oito minutos e meio. O vídeo apresenta trechos dessas 8 sessões, e neles estão presentes o ursinho, uma terapeuta e uma criança diferente em cada trecho.

Figura 14 – Ursinho interativo PlusMe.

Fonte: Colagem de Sperati, Özcan *et al.* (2019b).

Uma vez que o conjunto de frames desse vídeo não é um *dataset* de treinamento supervisionado por si só, ou seja, não é um conjunto estruturado com anotações úteis pré feitas para tarefas de machine learning, chega o momento de passá-los por uma fase de anotação manual. Para isso, o software escolhido foi o *Labelbox*[2], pois é uma plataforma com várias utilidades para todo o processo de anotação de datasets, treinamento e validação de modelos.

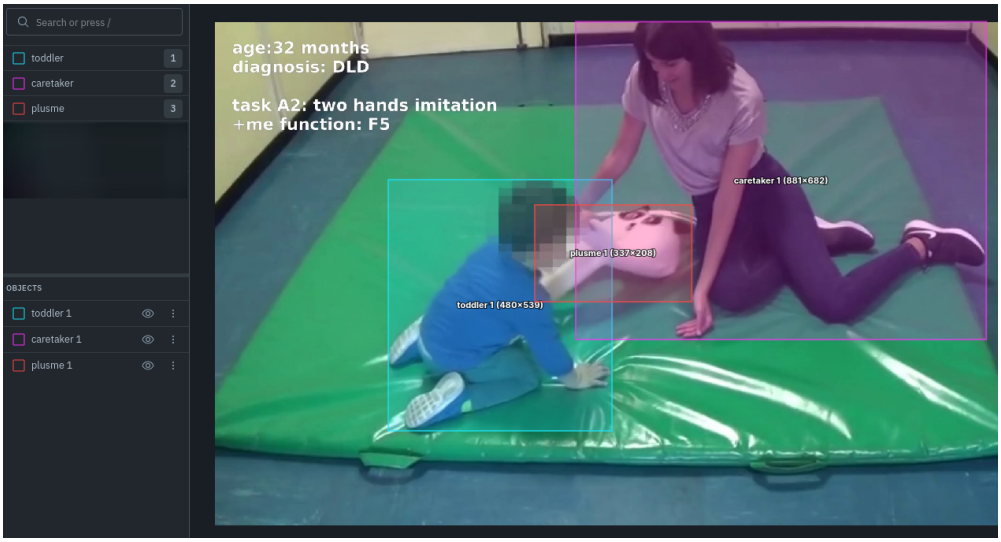
Esse vídeo gerou um total de 9491 frames dentro dos quais escolhemos apenas **819 frames aleatórios** para passarem pela etapa de anotação. Esse valor de 819 vem do fato de escolhermos cerca de 100 frames aleatórios para cada sessão (8 sessões no total),

[1] https://www.plusme-h2020.eu/
[2] https://labelbox.com/

Figura 15 – Frames de cada trecho do vídeo disponibilizado

Fonte: Colagem de Sperati (2020).

Figura 16 – Interface para anotações do Labelbox. Da esquerda para direita temos as bounding boxes para *Criança*, *PlusMe* e *Terapeuta*.



Fonte: O autor.

e não mais do que isso porque esse trabalho manual consome bastante tempo e requer muita atenção, o que inviabilizaria o prosseguimento com outras etapas do projeto. Caso houvesse mais tempo durante o projeto poderíamos retornar a essa etapa.

O Labelbox possibilita que criemos um esquema de anotações, e como a ideia inicial que tínhamos para modelos de visão computacional era a de fazermos detecção de todos os três atores, **Criança**, **Terapeuta** e **PlusMe**, utilizamos um esquema de caixas delimitadoras para os três atores que poderá ser usado no treinamento da rede YOLO. Após a conclusão dessas anotações conseguimos seguir para a próxima etapa.

4 RESULTADOS

Este capítulo apresenta a sequência de passos tomados com base na solução proposta e os resultados obtidos envolvendo treinamento de modelos de aprendizado de máquina com o algoritmo da YOLO, detecção de eventos com heurística de sobreposição de caixas delimitadoras, redução de tempo de análise e métricas avaliativas.

4.1 YOLOV4

Inicialmente, decidimos usar a implementação original da YOLO na sua quarta versão (YOLOv4) em linguagem C utilizando o framework *Darknet*[1]. Esse framework disponibiliza várias redes neurais para fins diversos, incluindo a YOLO para detecção de objetos.

Após a devida instalação do framework e dependências fizemos algumas detecções com os modelos pré treinados em imagens e vídeos pessoais e da internet. A YOLO teve sucesso em detectar os objetos, mas a interface disponível não era amigável o suficiente para alavancar múltiplas detecções e monitoramento de evolução e desempenho dos modelos.

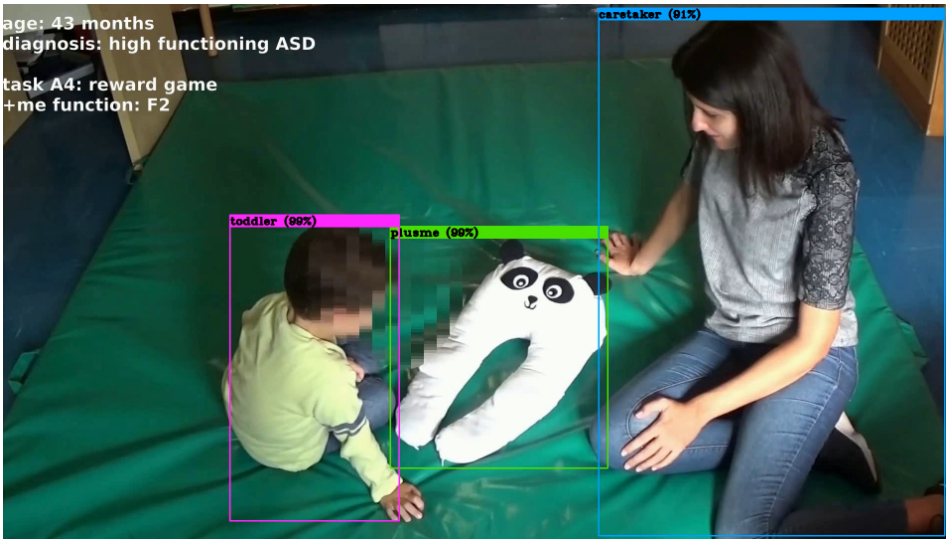
Em um dos treinamentos realizados, utilizamos 600 frames para treino e o restante para validação e teste. Em poucas épocas de treinamento os modelos pré treinados demonstravam um alto desempenho na tarefa de identificar os três atores, como mostra a Figura 17. Porém, apesar da perda no conjunto de treinamento estar baixa (0.86), e a mAP no conjunto de validação estar alta (95%) (Figura 18), temos apenas o valor da perda conjunta de classificação, objetividade e localização, e nenhum valor de perda para validação, o que impede de checarmos a ocorrência de overfitting em um dos 3 tipos de perda, que pode estar sendo mascarado pela alta mAP, uma vez que os frames de todos os 8 trechos do vídeo apresentam muitas características em comum.

Porém, apesar de encontrarmos muitas detecções ótimas da YOLO, uma quantidade considerável de detecções ruins apareciam também, independente de ser no conjunto de treino ou teste, como na figura 19, onde as bounding boxes não delimitam corretamente os três atores.

Com a darknet, para detectarmos uma sequência de frames aleatórios, como é o caso do nosso dataset de 819 frames, foi mais conveniente transformá-los em um vídeo contínuo e usar o *detector de vídeo* da YOLOv4, porém, esse detector de vídeo utiliza um *algoritmo interpolador* entre cada frame para economizar tempo de processamento. Isso é muito bom para um vídeo com frames em sequência, onde um objeto comum se movendo no tempo do ponto A até o ponto B segue uma trajetória previsível entre frames, mas péssimo para o nosso caso, onde usamos frames aleatórios, fazendo com que a interpolação gerasse bounding boxes deslocadas de seus respectivos atores.

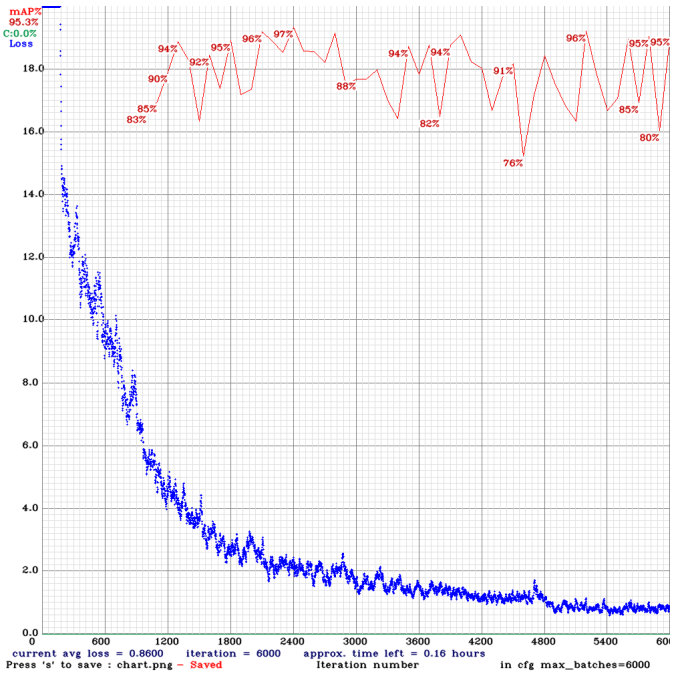
[1] https://github.com/AlexeyAB/darknet

Figura 17 – Detecção experimental pela YOLOv4.



Fonte: O autor.

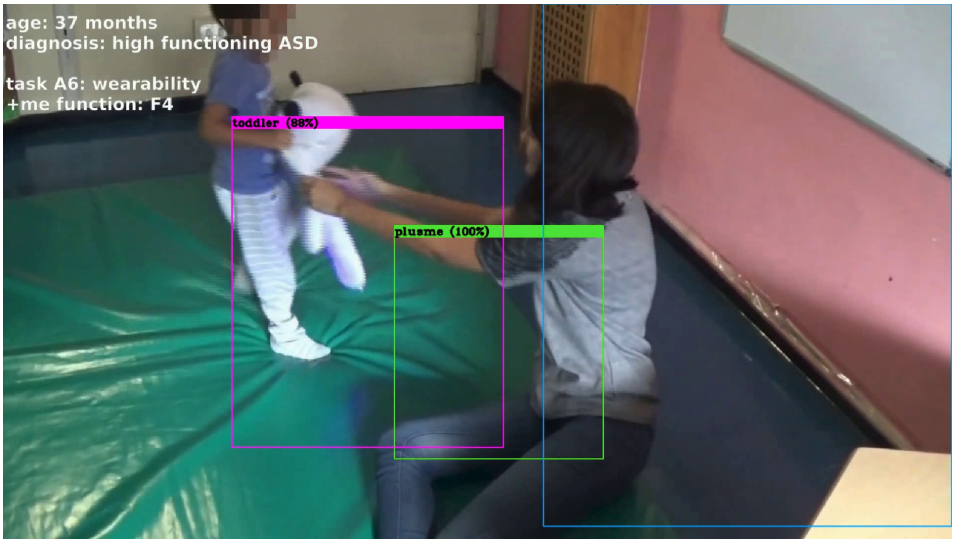
Figura 18 – Treinamento experimental com YOLOv4. Perda (azul) no conjunto de treinamento e métrica mAP (vermelho) no conjunto de validação.



Fonte: O autor.

A Figura 20 mostra um exemplo desse deslocamento de bounding boxes na imagem da esquerda, onde o frame está sendo processado pelo detector de **vídeo** (com auxílio do algoritmo interpolador) e gerando resultados errados. Já na direita temos o mesmo frame sendo processado pelo detector de **imagens** (apenas um único frame é detectado sem interpolação).

Figura 19 – Detecção ruim da YOLOv4.



Fonte: O autor.

Figura 20 – Comparação entre detecção com interpolação (esquerda) e sem (direita).

Fonte: O autor.

4.2 YOLOV5

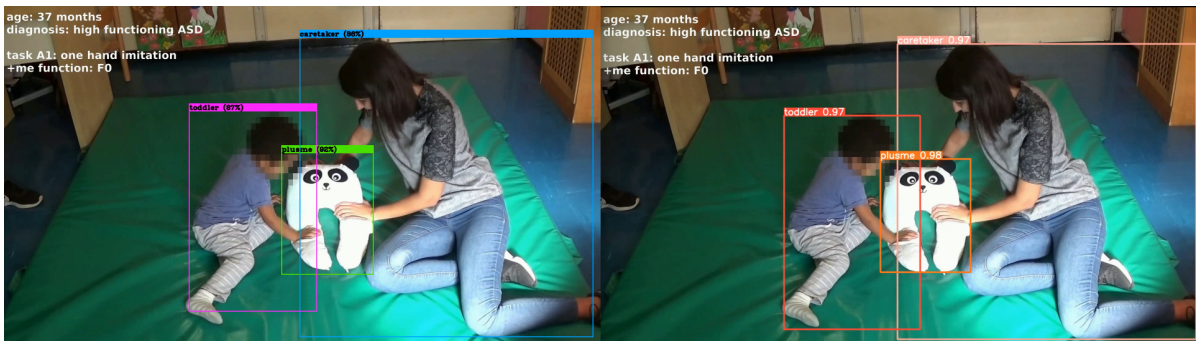
Com a necessidade de automatizar o treinamento e detecção dos modelos, assim como a evolução e desempenho dos mesmos, decidimos parar os experimentos com a YOLOv4 e começar a usar a *YOLOv5*[2] (JOCHER *et al.*, 2022) da empresa *Ultralytics*[3]. Por mais que não seja um *fork* direto do trabalho original da YOLO, é uma excelente implementação em *PyTorch*[4], um framework bem estabelecido em aprendizado de máquina, ganhando bastante destaque. Desde os primeiros momentos de uso, tivemos um ganho em usabilidade, interface e conectividade com outros serviços que não tínhamos com a YOLOv4.

Fizemos uma comparação rápida entre os modelos da YOLOv4 e YOLOv5 com poucas imagens até atingir overfitting e percebecemos que os resultados da YOLOv5 são

[2] https://github.com/ultralytics/yolov5
[3] https://ultralytics.com/
[4] https://pytorch.org/

muito parecidos com da YOLOv4 como mostra a Figura 21, fazendo com que a YOLOv5 fosse nossa escolha definitiva.

Figura 21 – Comparação entre detecções, YOLOv4 (esquerda) e YOLOv5 (direita).



Fonte: O autor.

4.2.1 Treinamento de Modelos

Em nosso laboratório usamos uma placa de vídeo *NVIDIA GeForce GTX 1650* com 4 Gb de *VRAM* o que acabou nos limitando no que se refere ao tamanho da rede neural, imagem e batch durante o treinamento, onde muitas vezes o uso de memória excedia ao disponível.

Para evitar erros inesperados pela sobrecarga da GPU, experimentamos vários modelos com modificações em cada um desses três parâmetros. O máximo batch possível foi de valor unitário para todos os modelos, e o tamanho máximo de imagem foi de 1280x1280 pixels com o modelo nano. A tabela 2 mostra alguns dos modelos treinados.

Para um comparação justa, todos os modelos foram treinados nos 4 primeiros vídeos (419 frames), validados no quinto e no sexto (200 frames) e testados nos dois últimos (200 frames). Para fins de referência, os frames na Figura 15 estão enumerados de 1 a 8. Além disso, os modelos foram programados para treinar durante 200 épocas, com um early stopping de paciência igual a 100 épocas (padrão do framework) demonstrando uma alta convergência ao verificarmos perdas no conjunto de treinamento em torno de 0.01 e overfitting quase nulo devido a perdas de validação próximas das de treinamento (em torno de 0.03).

Tabela 1 – Conjunto de Dados

Conjunto	Trechos do Vídeo	Frames
Treinamento	$1 \rightarrow 4$	419
Validação	5 e 6	200
Teste	7 e 8	200

Fonte: O autor.

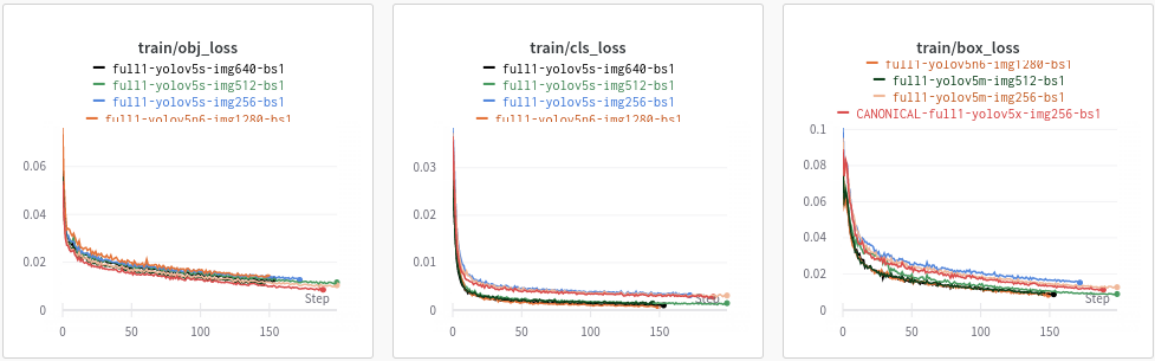
Tabela 2 – Modelos treinados da YOLOv5

Modelo	Tamanho da imagem (pixels)	Batch
Nano (n6)	1280x1280	1
Pequeno (s)	256x256 | 512x512 | 640x640	1
Médio (m)	256x256 | 512x512	1
Extra Grande (x)	256x256	1

Fonte: O autor.

O framework da YOLOv5 possibilita a integração com uma plataforma de histórico de modelos e métricas chamado *Weights and Bias*[5](WandB) que facilita a escolha e versionamento de modelos. Para todos os modelos da Tabela 2 foi gerado automaticamente resultados para perdas, métricas mAP e uso do sistema. Esses resultados formam um conjunto de informações qualitativas e quantitativas que nos permitem ponderar por diversos ângulos e formas de aplicações quais devem ser os modelos escolhidos.

Figura 22 – Perdas no treinamento dos modelos da YOLOv5.



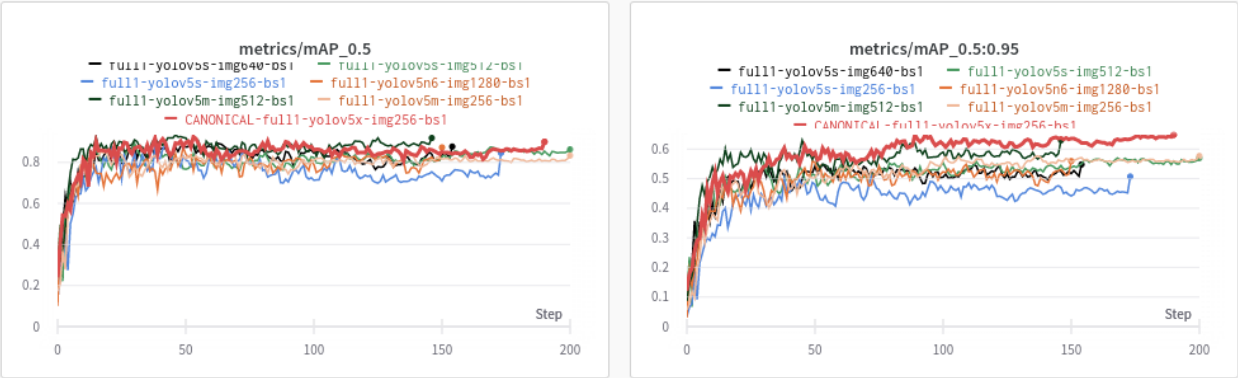
Fonte: O autor.

Figura 23 – Perdas na validação dos modelos da YOLOv5.

Fonte: O autor.

[5] https://wandb.ai/site

Figura 24 – mAP dos modelos da YOLOv5.



Fonte: O autor.

A Figura 22 mostra que as 3 perdas no treino chegaram numa convergência similar para todos os modelos, e como a escala da Figura 23 é bem pequena também, podemos dizer o mesmo para as perdas de validação. É muito importante ter boas perdas, mas uma das análises final para tarefas de detecção de objetos está em maximizar o valor da métrica mAP. Para isso, o melhor modelo nesse primeiro momento foi então o extra grande (linha vermelha na Figura 24). Esse modelo levou aproximadamente 16 horas para ser treinado durante 189 épocas, onde parou por causa do *early stopping*, indicando a 89ª época como a melhor de todas.

As figuras 25 e 26 mostram em detalhes as perdas e mAPs desse modelo durante todo o treinamento e a Tabela 3 sumariza os resultados para a época 89, a melhor.

Tabela 3 – Resultados do modelo extra grande na melhor época (89).

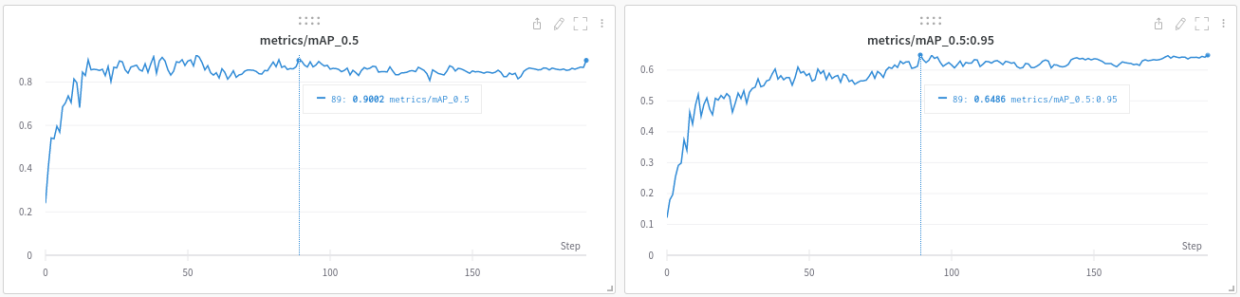
Conjunto	Class Loss	Box Loss	Obj. Loss	mAP@0.5	mAP@0.5:0.95
Treinamento	0.0036	0.0175	0.0124	—	—
Validação	0.0105	0.0324	0.0193	0.9002	0.6486

Fonte: O autor.

Figura 25 – Perdas no treino (azul) e validação (laranja) do modelo extra grande.

Fonte: Adaptado de Ribeiro, Soares, Carvalho e Grellert (2022).

Figura 26 – mAP@0.5 à esquerda e mAP@0.5:0.95 à direita do modelo extra grande.



Fonte: Adaptado de Ribeiro, Soares, Carvalho e Grellert (2022).

4.3 DETECÇÃO (PREDIÇÃO) DE EVENTOS

Interações por meio de toques compõem parte do trabalho realizado por Sperati, Özcan *et al.* (2019a) e Giocondo *et al.* (2022), existindo várias maneiras de detectarmos toques entre objetos e pessoas. Uma maneira é fazer a detecção detecção direta do evento, gerando bounding boxes para cada tipo de toque, como por exemplo, ao redor de regiões onde um a Criança está tocando no PlusMe ($Toque_{cri-pm}$).

Porém, como o conjunto de dados foi inicialmente anotado com classes referentes apenas aos 3 atores, estendê-lo para considerar novas classes de toque atrasaria o projeto pelo processo manual e lento de anotação.

Dessa forma, podemos explorar o toque entre **bounding boxes** na detecção de objetos (realizada pela YOLOv5) usufruindo de uma heurística simples: **"Sempre que houver algum toque ou sobreposição entre um par de bounding boxes, isso será considerado como uma interação entre aquele par de objetos"**. Dessa forma, a detecção de uma interação é na verdade uma **predição** com base na sobreposição das bounding boxes.

E para isso, foram definidos 3 tipos principais de interações (eventos), advindos dos 3 tipos de interações possíveis dentro do nosso dataset, são eles os toques bidirecionais entre *Criança-Terapeuta*, *Criança-PlusMe* e *Terapeuta-PlusMe*[6].

Cabe destacar que, como o uso dessa heurística evita o uso explícito de novas classes, a mesma pode ser usada (com as devidas adaptações) para detecção de interações em conjuntos de dados com outros esquemas de anotação, como por exemplo ao explorarmos o toque entre máscaras nas tarefas de segmentação ou entre esqueletos artificias na tarefa de *tracking* de membros.

Essa predição faz sentido pois a probabilidade de dois objetos num plano 2D se tocaram aumenta proporcionalmente ao quão perto eles e suas bounding boxes estão, porém não garante, como em casos de oclusão num plano 3D (CHENG *et al.*, 2019). A figura 27 mostra a existência de dois pares de sobreposição de bounding boxes, um entre Criança-PlusMe e um entre Terapeuta-PlusMe, porém só está acontecendo uma interação de fato, que é no primeiro caso, onde a Criança está tocando o ursinho PlusMe, mas não no segundo, onde não há nenhum toque entre Terapeuta e PlusMe. Isso mostra a existência de um verdadeiro positivo (VP) e um falso positivo (FP) para dois pares de sobreposição, respectivamente. Isso, o quanto de afirmações verídicas e não verídicas que nosso preditor está fazendo, deve ser estudado.

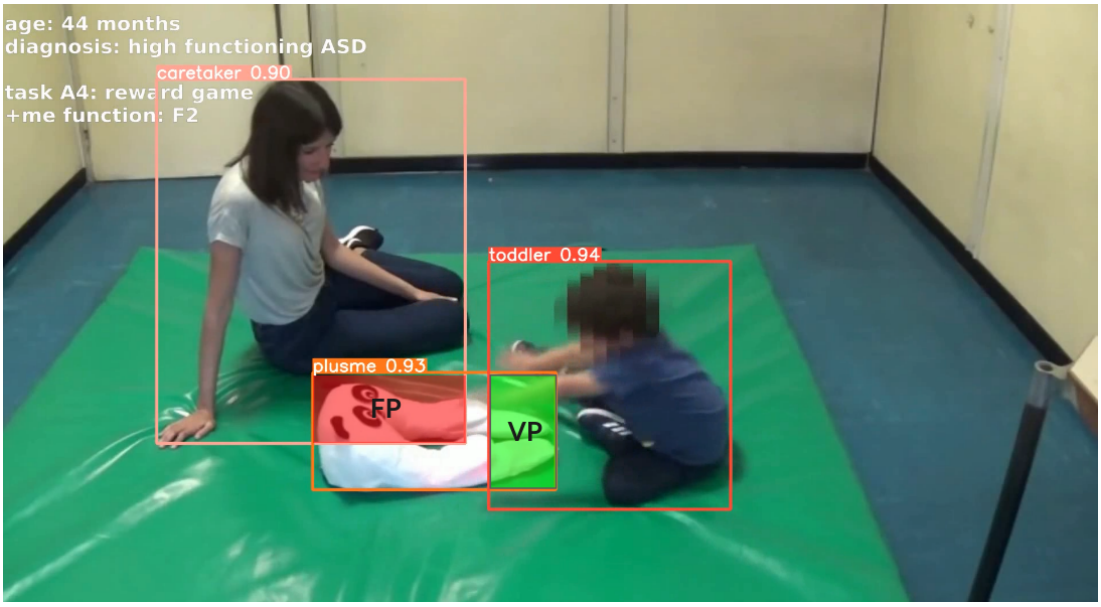
Para uma versão inicial, desenvolvemos um script que retornava no vídeo detectado, informações desenhadas em cada frame, se estava ocorrendo uma interação ou não de acordo com essa heurística. E como propomos no diagrama da ferramenta na Figura 13, temos que contabilizar em nossa análise, erros que o modelo da YOLO possa vir a cometer

[6] Essas classes de interações não compõem o esquema de anotação do conjunto de dados. Elas se originam da heurística de sobreposição de bounding boxes.

durante a tarefa de classificação e localização dos atores.

Percebemos que a maioria dos erros acontecia quando alguma bounding box faltava ou era classificada erroneamente como outra classe, como por exemplo na Figura 28, onde a Terapeuta é classifica erroneamente pela YOLO como sendo uma Criança, e pelo fato de estar sobrepondo com a bounding box do PlusMe é predito como uma interação Criança-PlusMe quando de fato as bounding boxes da Criança e do PlusMe não estão se tocando. Mas isso é corrigido logo em seguida eliminando bounding boxes repetidas da mesma classe com *score* de confiança menor, como na Figura 29. Outro exemplo de um erro da YOLO é quando uma das bounding boxes está faltando, seja como no exemplo anterior quando uma classe foi trocada, ou simplesmente nenhuma bounding box foi associada à ela **e está acontecendo uma interação**. Com isso o preditor acaba prevendo nenhuma interação quando de fato está acontecendo, como mostra a Figura 30, gerando um Falso Negativo (FN).

Figura 27 – Verdadeiro Positivo (VP) e Falso Positivo (FP) nas predições de interação.



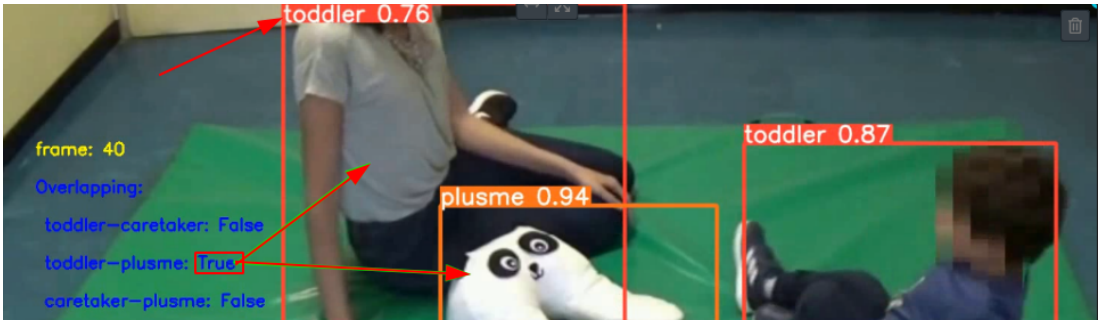
Fonte: O autor.

Dessa forma, para analisarmos o desempenho da heurística de sobreposição de bounding boxes (detector/preditor de eventos) devemos ignorar as situações onde há alguma bounding box faltante, e para analisarmos o desempenho da ferramenta como um todo (sistema) devemos considerá-las.

Dos 200 frames utilizados para essa análise, 19 continham erros de detecção e os 181 restantes não. Assim, como todo frame é parte da saída do sistema, a análise do mesmo conta com 200 frames, mas a análise do detector conta apenas com os 181 que não continham erros.

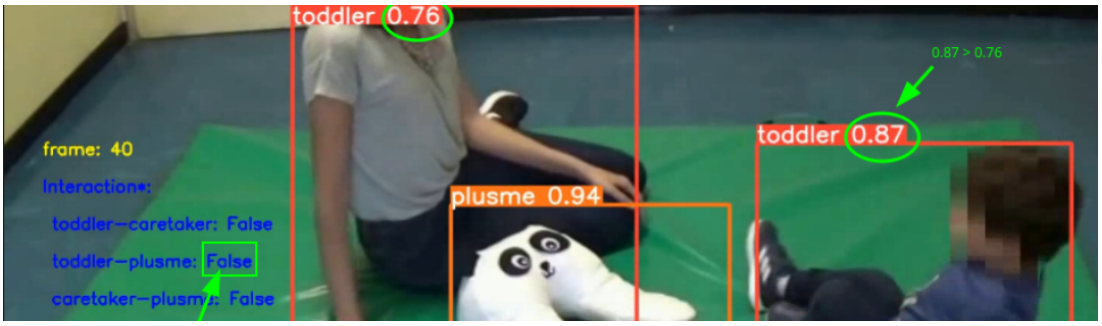
Um quadro com ausência de alguma bounding box não é considerado válido para a avaliação do preditor, e automaticamente é considerado como relevante para o usuário.

Figura 28 – Predição incorreta de interação entre *Criança-PlusMe*.



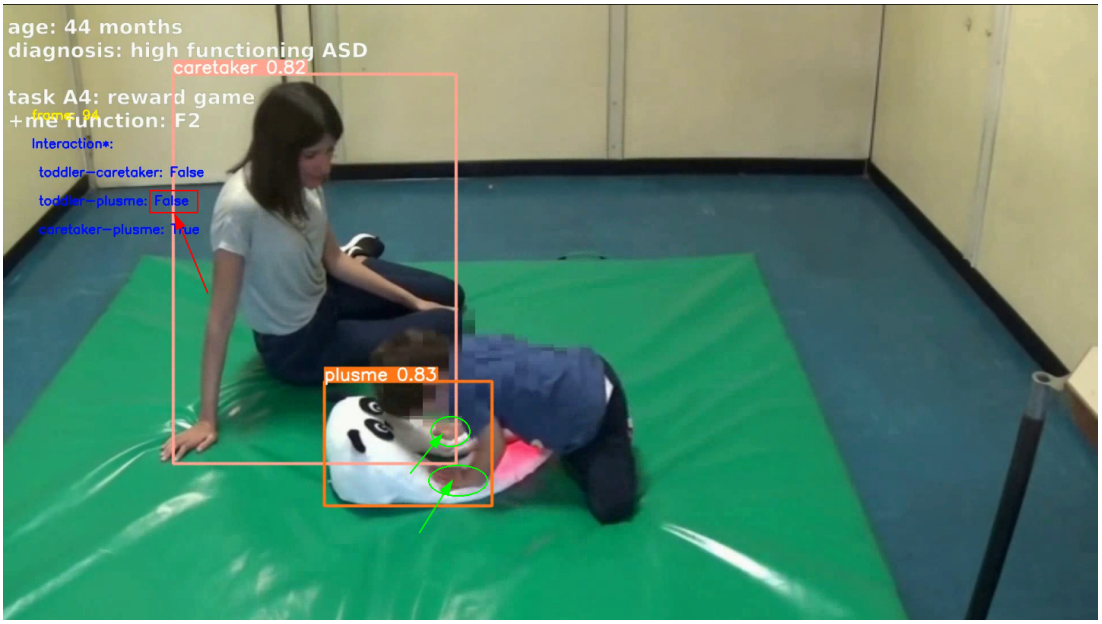
Fonte: O autor.

Figura 29 – Predição correta de interação entre *Criança-PlusMe*.



Fonte: O autor.

Figura 30 – Falso Negativo (FN) na predição entre *Criança-PlusMe*.



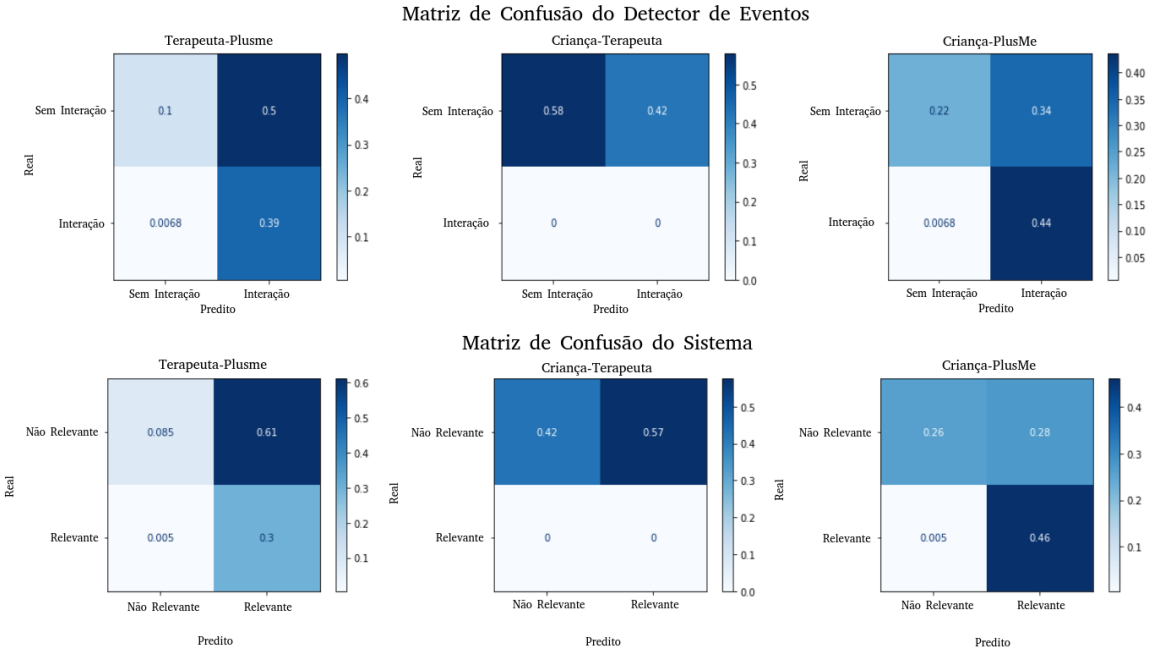
Fonte: O autor.

Onde um frame relevante significa a possibilidade de uma interação onde há a falta de uma bounding box, que por sua vez também é considerado como uma **predição**. Dessa

forma, para verificar se o frame é realmente relevante, se de fato está acontecendo uma interação sem a informação das bounding boxes, o usuário deve verificar manualmente esse frame.

Tanto o preditor de eventos quanto o sistema trabalham com predições e com os valores reais (*ground-truth* - verificados manualmente). Com isso, podemos montar matrizes de confusão para cada par de interação no que se refere a "considerar um frame com interação" e "considerar um frame como relevante". Dessa forma estamos avaliando quão bom o preditor é em usar a heurística de sobreposição entre um par de bounding boxes, e quão bom o sistema é em indicar para o terapeuta onde ele deve fazer sua análise médica (que deve ser apenas onde realmente está acontecendo uma interação).

Figura 31 – Matriz de Confusão para o Detector (Preditor) de Eventos e o Sistema como um todo.



Fonte: Adaptado de Ribeiro, Soares, Carvalho e Grellert (2022).

Tabela 4 – Redução da análise manual do sistema em vídeos.

Vídeo	Frames	Ter-PM	Cri-Ter	Cri-PM	Todas interações
Vídeo 1	100	0.00%	34.0%	35.0%	0.0%
Vídeo 2	100	17.8%	51.4%	18.8%	17.8%
Vídeos 1 e 2	200	9.0%	42.5%	11.39%	9.0%

Fonte: Adaptado de Ribeiro, Soares, Carvalho e Grellert (2022).

A Figura 31 mostra as matrizes de confusão para o preditor e o sistema. Como queremos indicar para o terapeuta quais frames são relevantes (que precisam de análise

manual), é muito importante que o valor de Falsos Negativos seja baixo, pois é necessário que qualquer interação seja recomendada para ele analisar e tomar uma decisão, o que é verdade para cada uma das 6 matrizes. Além disso, essa indicação de onde analisar traz uma redução no tempo de análise para o usuário, que não precisa procurar no vídeo inteiro onde está acontecendo determinado evento de interesse.

Utilizando os frames dos 2 vídeos do conjunto de teste para a criação da Tabela 4, mostramos que é possível uma redução de até 51% de análise se considerarmos apenas uma interação como foco. As siglas da tabela representam o seguinte: Cri-PM para Criança-PlusMe, Ter-PM para Terapeuta-PlusMe e Cri-Ter para Criança-Terapeuta.

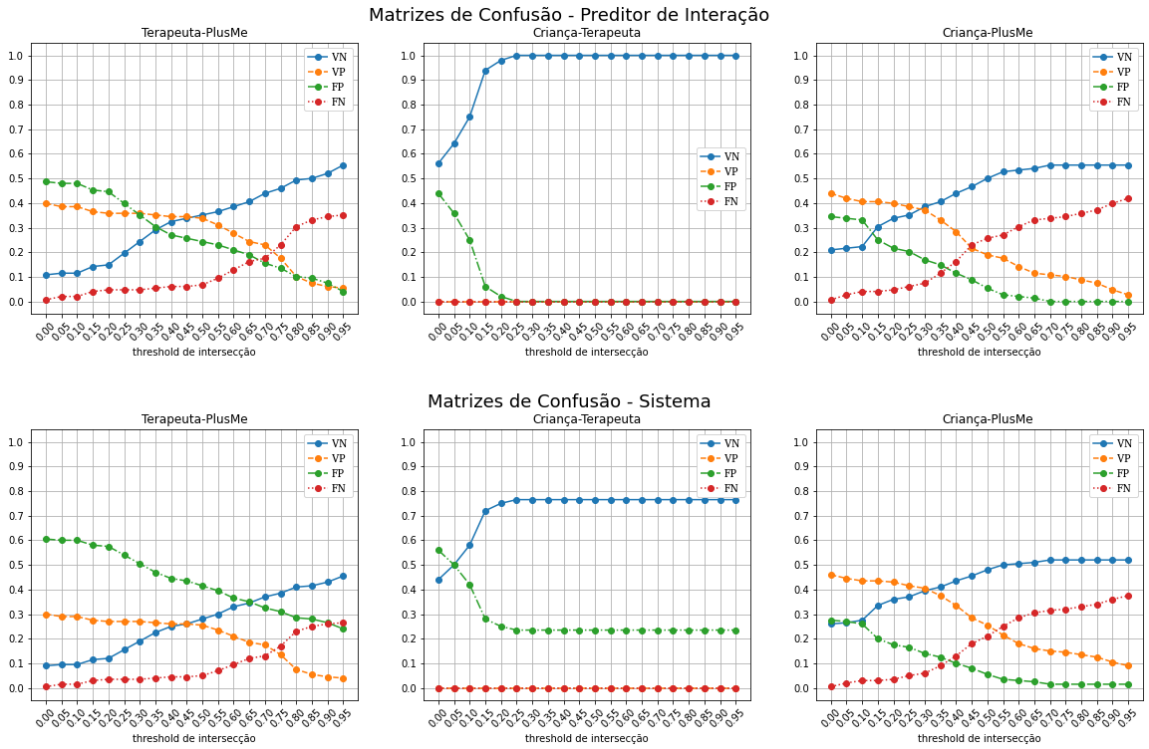
4.3.1 Detecção de Proximidade e Predição de Interação

No próximo capítulo exploramos uma interface para o sistema que facilita a tarefa de indicar para o usuário um momento relevante no vídeo, utilizando uma *timeline* de eventos como forma de retorno visual. Para essa timeline, percebemos que indicar o quão perto os atores estão um do outro, mesmo que não esteja acontecendo nenhuma interação, fornece informações temporais entre toques que o usuário pode querer analisar também. E para isso, além de extrair informações sobre a sobreposição de um par de bounding boxes, podemos extrair o **nível de sobreposição**, que nada mais é que a intersecção dessas bounding boxes. Para termos uma noção de *proximidade* entre atores, essa intersecção é normalizada entre 0 e 1, o que conseguimos ao dividir o valor da intersecção pela área da menor bounding box da sobreposição. Sem isso teríamos um valor absoluto que foge ao que queríamos analisar.

Anteriormente, qualquer sobreposição de bounding boxes era predito como uma interação, mas agora temos a liberdade de escolher um limiar, ou *threshold* em inglês, que define quando fazer essa predição já que agora temos um valor numérico, e não mais Booleano (com sobreposição ou sem), é claro, à custa de modificar as matrizes de confusão que vimos na Figura 31, que agora percebemos, apresentava limiar igual a 0.

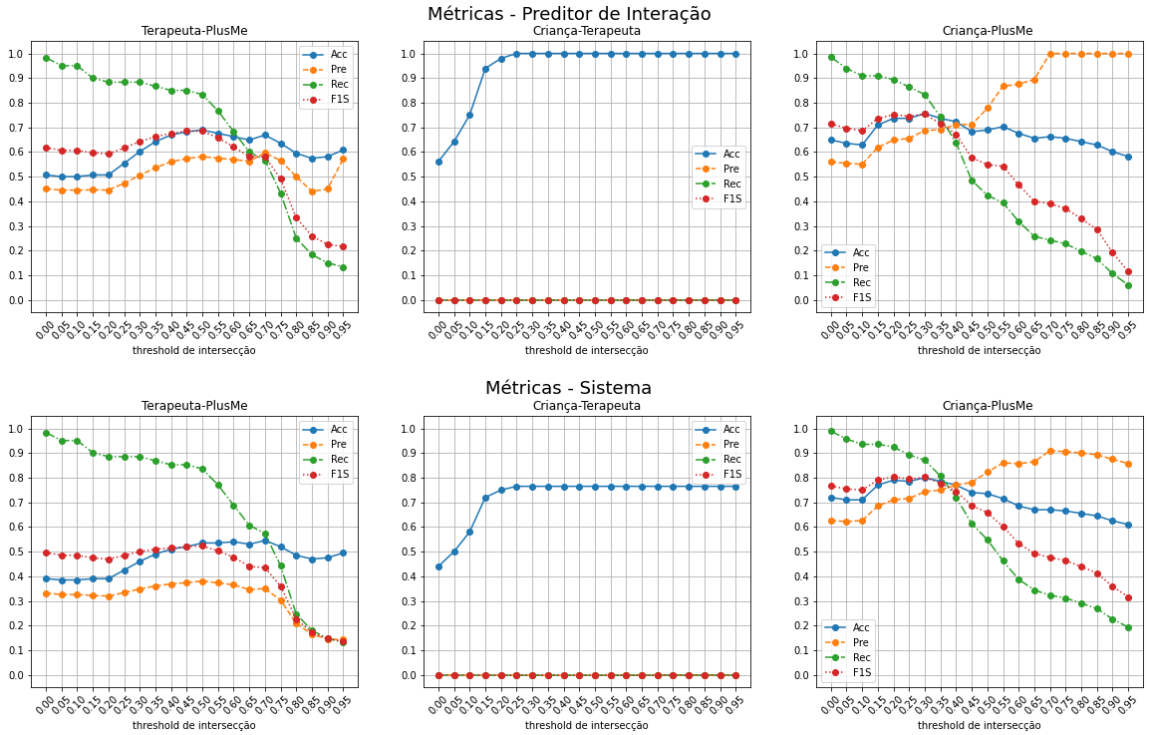
Para entendermos o efeito do threshold de intersecção foi realizado um estudo similar do apresentado nas matrizes de confusão da Figura 31 e da redução no tempo de análise na Tabela 4 apenas com a junção dos vídeos 1 e 2. Com um threshold variando de 0 até 0.95 em passos de 0.05 é possível visualizar os componentes das matrizes de confusão VN, FP, FN e VP e as métricas Acurácia, Precisão, Recall e F1Score tanto para o preditor de interação quanto para o sistema. E além disso podemos visualizar também a redução de análise temporal do sistema. As Figuras 32, 33 e 34 mostram os resultados, e os Quadros 1, 2 e 3 sumarizam todas as conclusões desse estudo.

Figura 32 – Evolução das Medidas das Matrizes de Confusão.



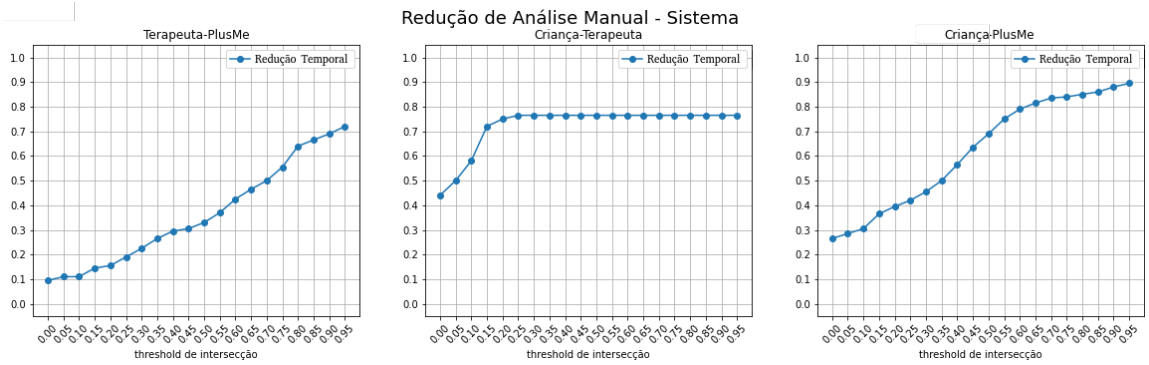
Fonte: O autor.

Figura 33 – Evolução das Métricas.



Fonte: O autor.

Figura 34 – Evolução da Redução de Análise Manual do Sistema.



Fonte: O autor.

Quadro 1 – Conclusões: Evolução das Medidas das Matrizes de Confusão.

Medida	**Tendência**	**Discussão**
FN	Crescente ↑	O aumento dessa medida no gráfico faz sentido, pois quanto maior o threshold, mais estamos deixando passar resultados verdadeiros com baixa sobreposição. Causando a predição de "não ocorrência de interação" quando na verdade está ocorrendo.
FP	Decrescente ↓	Aqui, o decaimento é explicado pela oclusão que ocorre entre os atores no vídeo, que está muito relacionado com o nível de sobreposição das bounding boxes. Quando o threshold aumenta, maior é o número de oclusões com nível de sobreposição abaixo desse limiar sendo consideradas como "não interação", o que diminui a falsa predição de "interação" quando há somente oclusão.
VN	Crescente ↑	Pela mesma explicação do FP no quesito de oclusão, conforme o threshold for aumentando, maior o número de oclusões que ficará abaixo dele. Assim, será feito uma predição de "não ocorrência" e de fato não estará ocorrendo nada.
VP	Decrescente ↓	No mesmo sentido da explicação em FN, esses vídeos apresentam mais interações acorrendo quando o nível de sobreposição é mais baixo, os níveis mais altos tendem a estarem envolvidos com oclusão. Dessa forma, com o aumento do threshold, fica mais difícil encontrar uma sobreposição que signifique de fato uma interação.
Obs.		**1)** Perceba que, por mais que essas medidas sejam diferentes entre cada um dos pares de interação, e entre o preditor e o sistema, elas apresentam a mesma tendência e por isso as conclusões desse quadro se aplicam a todas as situações (menos quando as medidas são iguais a 0 durante toda a evolução). **2)** Podemos perceber também que as medidas de VP e FN são sempre nulas para a interação Criança-Terapeuta. Isso se deve ao fato de que os vídeos 1 e 2 não apresentavam interação entre esses dois atores.

Fonte: O autor.

Quadro 2 – Conclusões: Evolução das Métricas.

Métrica	**Tendência**	**Discussão**
Acurácia	Crescente ↑	Teve um comportamento majoritariamente crescente em 4 situações das 6.
Precisão	Crescente ↑	Dos 6 plots de métricas, apenas a metade teve um comportamento crescente indicando que quanto maior o threshold, maior é a chance do preditor e do sistema acertarem. A outra metade representou uma Precisão nivelada ou nula.
Recall	Decrescente ↓	Na Recall, seu denominador é fixo com o número de ground truth existente, o que faz ela decair é o decaimento do VP que se encontra sozinho no numerador.
F1Score	Decrescente ↓	A F1Score se demonstrou decrescente após o pico em 4 das 6 situações. Com o picos acontecendo entre os valores de threshold 0.3 e 0.5.
Obs.		**1)** Dentro da proposta do projeto, uma Recall alta é bem importante, pois entregamos para o terapeuta tudo que ele precisa verificar manualmente. Dessa forma, é interessante o uso do threshold nulo (0) para a análise. **2)** Porém, uma Recall alta representa uma Precisão baixa, então é uma boa ideia questionar se realmente é viável mostrar todo e qualquer tipo de interação para o terapeuta, pois pode acontecer casos onde o médico não vai querer olhar para pequenas interações como um simples esbarrar de dedo, mas sim para um aperto de mão, um abraço, etc. Com isso, seria interessante aumentar o valor do threshold para filtrar apenas interações expressivas que apresentam alta Precisão.

Fonte: O autor.

Quadro 3 – Conclusões: Evolução da Redução de Análise Manual do Sistema.

Medida	**Tendência**	**Discussão**
Redução da Análise Manual	Crescente ↑	A evolução crescente dessa medida indica uma relação direta com o threshold escolhido, mostrando que nesses vídeos o mais comum é acontecer toques com baixo nível de sobreposição entre os atores. No caso, com um threshold baixo igual a 0, o programa mostrará todo e qualquer toque, acidental ou intencional, ao custo de reduzir no máximo 9% da análise como mostra a Figura 34 para Terapeuta-PlusMe. Já thresholds mais altos, tendem a mostrar toques mais significativos, reduzindo em muito a análise manual, como a redução em 90% no último threshold entre Criança-PlusMe, também na Figura 34.

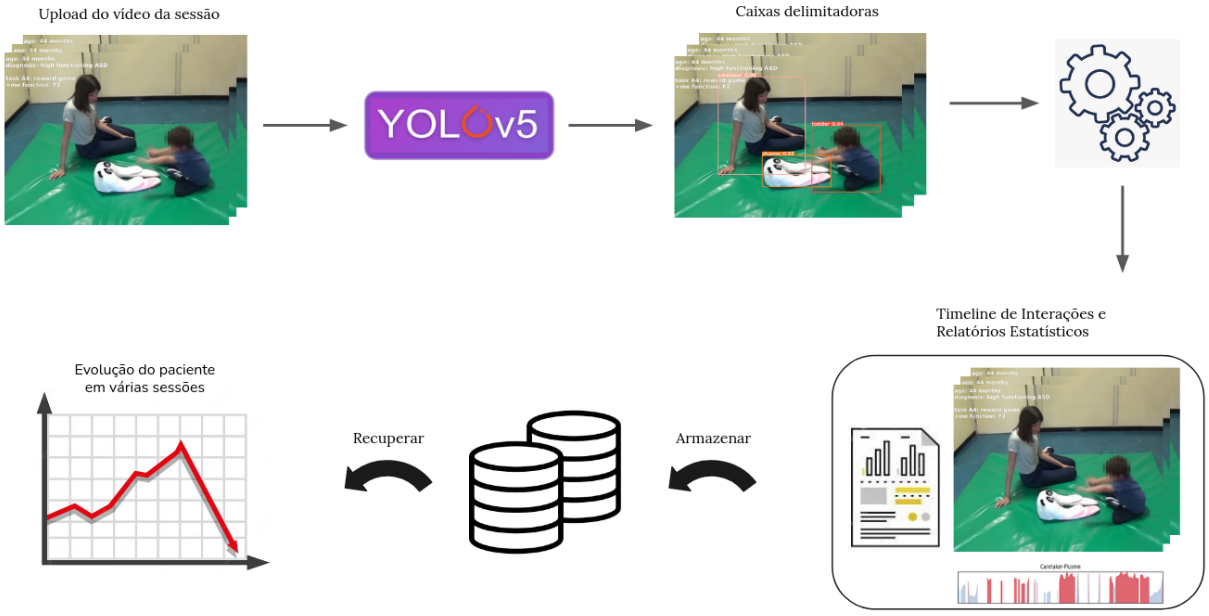
Fonte: O autor.

5 CONSTRUÇÃO DA FERRAMENTA

O objetivo final deste trabalho foi a construção e manutenção de um software web gratuito que auxiliasse na tomada de decisões na terapia de pacientes com TEA. Para isso utilizamos o framework chamado *Streamlit*, que de um jeito prático transforma scripts Python em páginas web.

A Figura 35 mostra um diagrama de como o aplicativo funciona. Inicialmente o usuário faz upload de vídeos de sessão de terapia e em seguida a YOLO realiza a detecção dos atores retornando bounding boxes ao redor de cada ator para cada um dos frames, depois é realizado a predição de interações e detecção de proximidade por meio do detector de eventos, indicando onde estão acontecendo as principais interações no vídeo por meio da construção de uma "timeline de eventos", juntamente com estatísticas de interação. Uma vez que esse vídeo foi processado é possível adicioná-lo numa base de dados e recuperá-los depois, juntamente com dashboards evolutivos ao longo das sessões de um paciente específico.

Figura 35 – Diagrama de funcionamento da ferramenta.



Fonte: O autor.

Dentro do nosso aplicativo, a entidade que contém informações de bounding boxes e proximidade para cada frame é o *Vídeo*. Uma vez que o Vídeo é processado pela YOLO, ele passa a conter um dicionário[1] com bounding boxes dos três atores para cada frame. Seguindo a implementação, é criado um outro dicionário com os níveis de sobreposição normalizados para cada par de bounding boxes. O que difere do capítulo anterior é que o

[1] Em Python, um dicionário é uma estrutura de dados baseada em chaves e valores. É usado dentro deste trabalho para conter informações de detecções em cada frame.

conteúdo do dicionário de interações é criado agora a partir de um limiar das sobreposições que não precisa ser 0 (como fazíamos ao considerar uma interação quando havia qualquer sobreposição). Além desses dicionários com valores para cada frame, também é criado um outro com estatísticas de interações, contendo o número de interações e tempo mínimo, máximo e médio das interações. O algoritmo 1 sumariza esses passos. E para possibilitar a serialização dessas informações foi implementado um sistema CRUD (adição, remoção, consulta e atualização de dados) em SQLite.

Algorithm 1 Criação dos atributos do Vídeo

```
capture ← video capture in memory
model ← YOLOv5 model
threshold ← K ∈ [0, 1]

for frame in capture do
    bbs_i ← model.detect(frame)          ▷ Bounding boxes para cada frame
    overl_i ← overlaping(bbs_i)          ▷ Nível de sobreposição para cada frame
    inter_i ← (overl_i > threshold)      ▷ Interação para cada frame (True/False)
end for

stat ← statistics(inter)                 ▷ Estísticas das interações
video ← Video(bbs, overl, inter, stat)   ▷ Constrói o Video
```

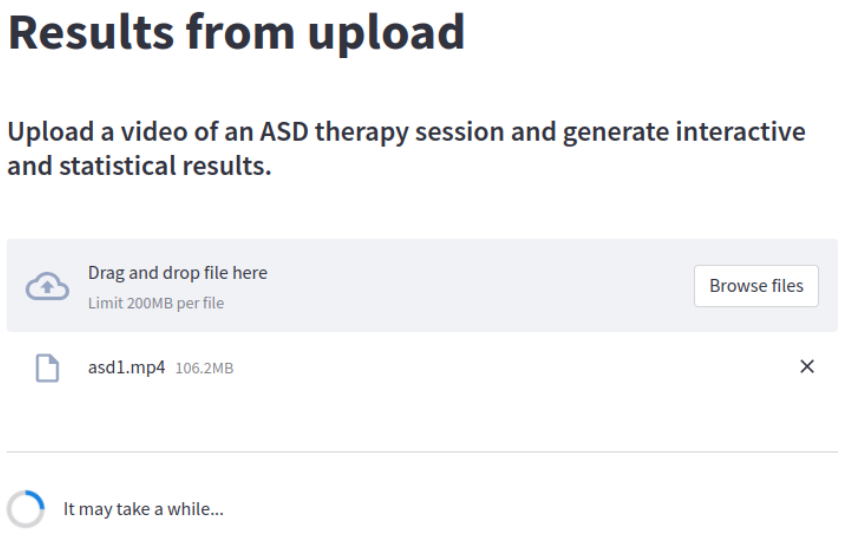
A ferramenta se encontra disponível no GitHub[2] juntamente com documentação explicativa. Com ela o usuário pode rodar um servidor local do Streamlit, ou modificar caso necessário para rodar em outra infraestrutura. É possível também o ajuste de qual modelo pré-treinado usar, por meio de arquivos de configuração.

As Figuras abaixo mostram um pouco da interface do aplicativo. A Figura 36 mostra a área de upload de um vídeo de sessão do TEA, onde é indicado o início do processamento após a escolha do arquivo. Por padrão, o threshold de intersecção para se considerar uma interação é de **0.6**, e as Figuras 37, 38, 39 e 40 mostram as timelines de interações para Criança-PlusMe[3] de dois vídeos diferentes, onde a parte acinzentada indica a proximidade e a vermelha as interações. Ainda nessas figuras, é indicado manualmente pelo autor os casos de verdadeiro positivo, falso positivo, verdadeiro negativo e falso negativo durante a análise da timeline. Após a análise da timeline, o usuário pode escolher adicionar esses resultados na base de dados, com nome do paciente e data da sessão, conforme mostrado na Figura 41. Depois do cadastro da sessão, cada uma das métricas recolhidas pode ser comparada num gráfico de evolução do paciente conforme novas sessões são realizadas, como por exemplo nas Figuras 42, 43 e 44, que mostram o aumento do tempo de interação e número de interações feitas em cada sessão entre os atores.

[2] https://github.com/solisoares/therapy-aid-tool

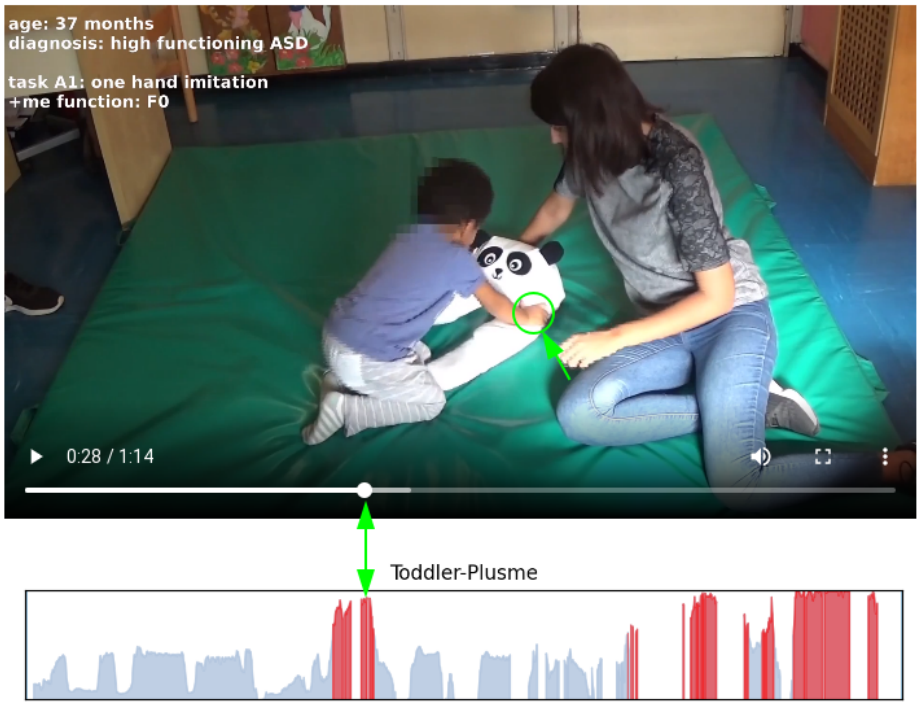
[3] Na ferramenta também é incluso a timeline de interações entre Terapeuta-PlusMe e Criança-Terapeuta.

Figura 36 – Área de upload da ferramenta.



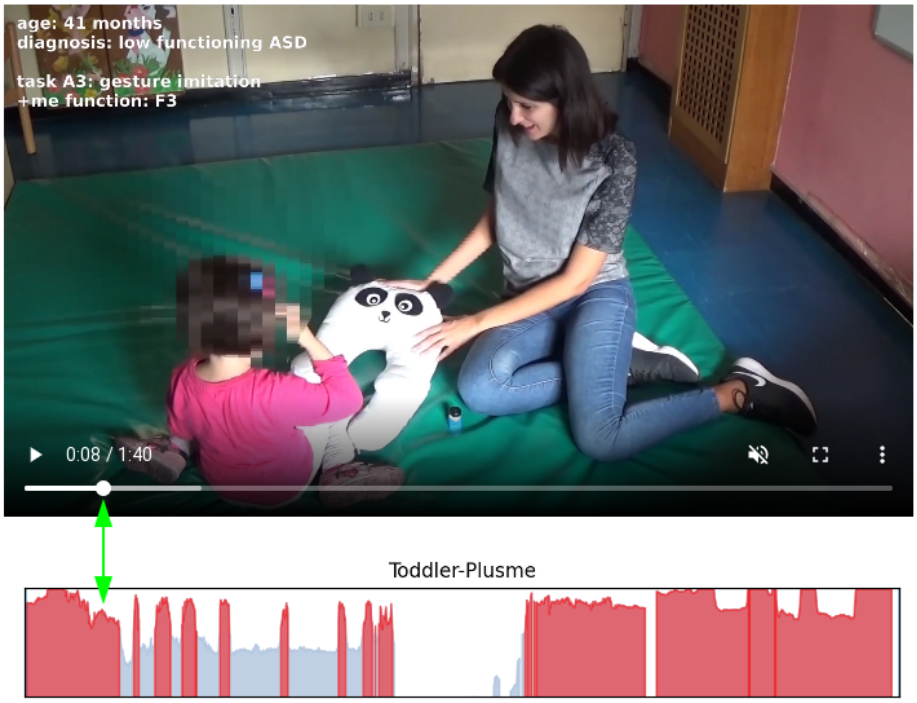
Fonte: O autor.

Figura 37 – Verdadeiro Positivo na timeline (Criança-PlusMe).



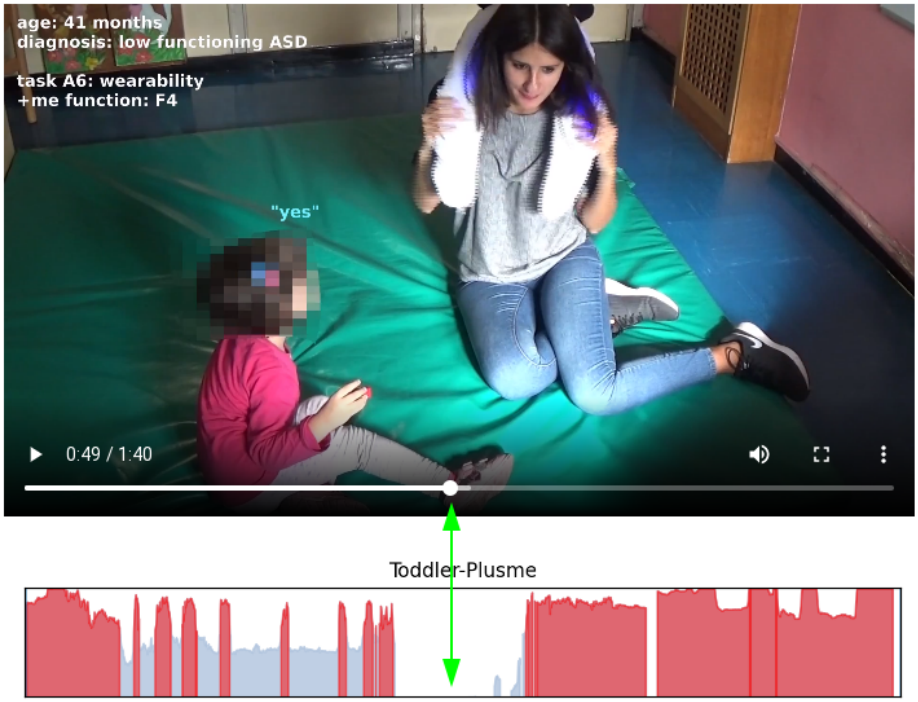
Fonte: O autor.

Figura 38 – Falso Positivo na timeline (Criança-PlusMe).



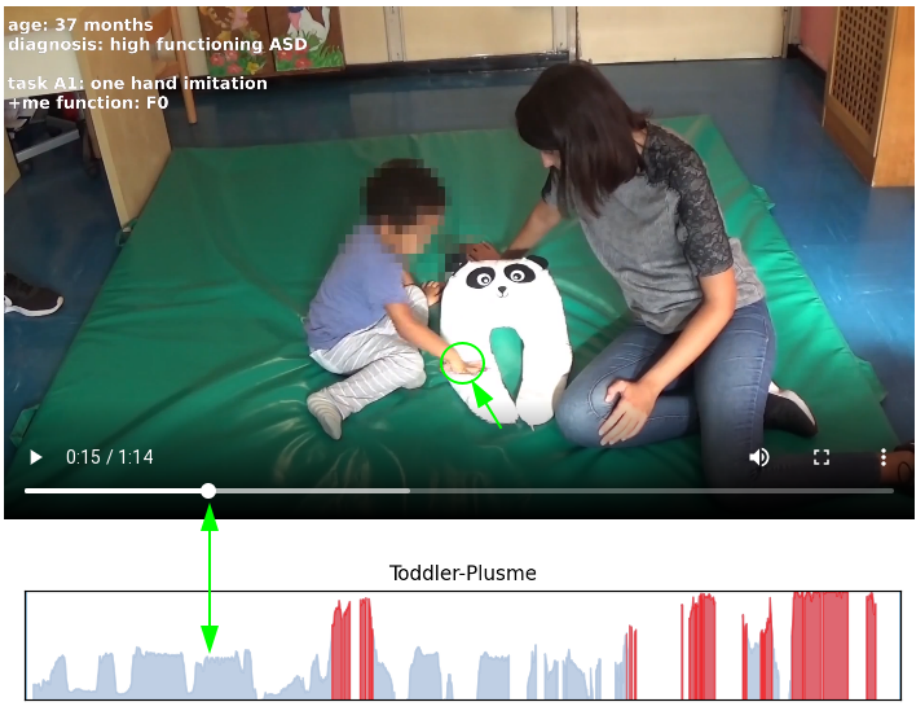
Fonte: O autor.

Figura 39 – Verdadeiro Negativo na timeline (Criança-PlusMe).



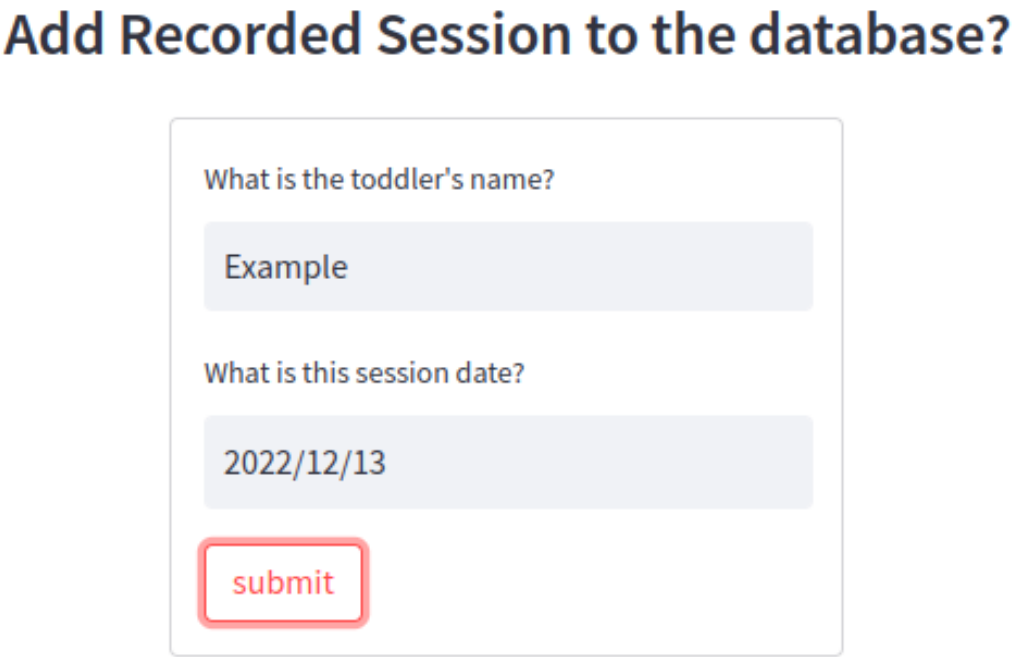
Fonte: O autor.

Figura 40 – Falso Negativo na timeline (Criança-PlusMe).



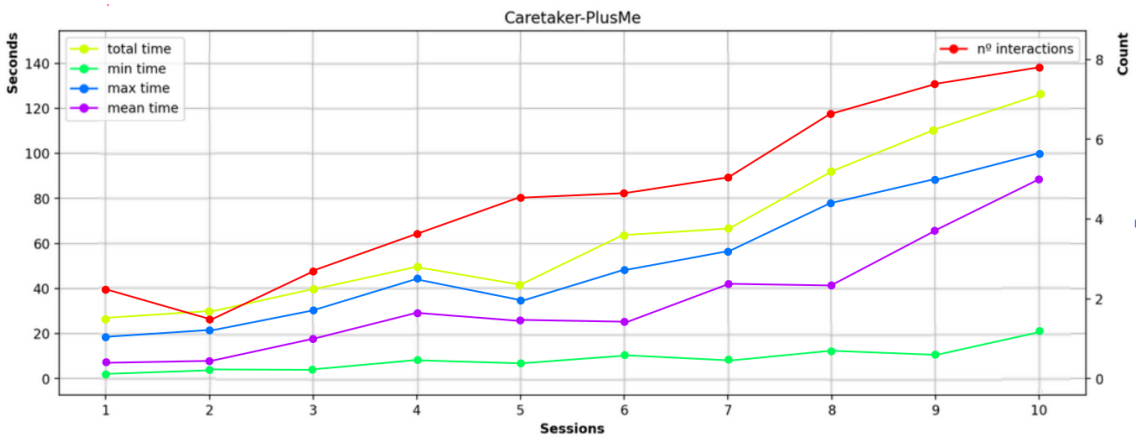
Fonte: O autor.

Figura 41 – Adição do vídeo processado na base de dados.



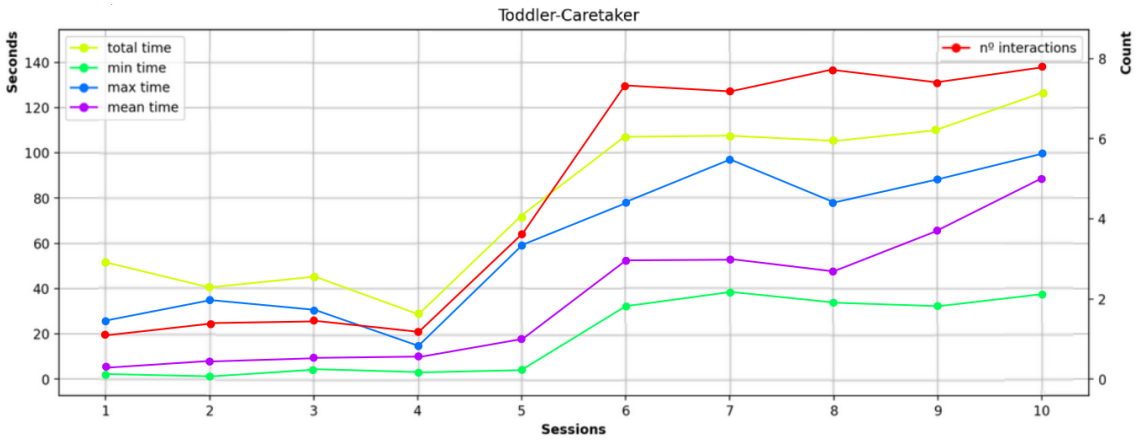
Fonte: O autor.

Figura 42 – Evolução da interação entre Terapeuta-PlusMe entre sessões.



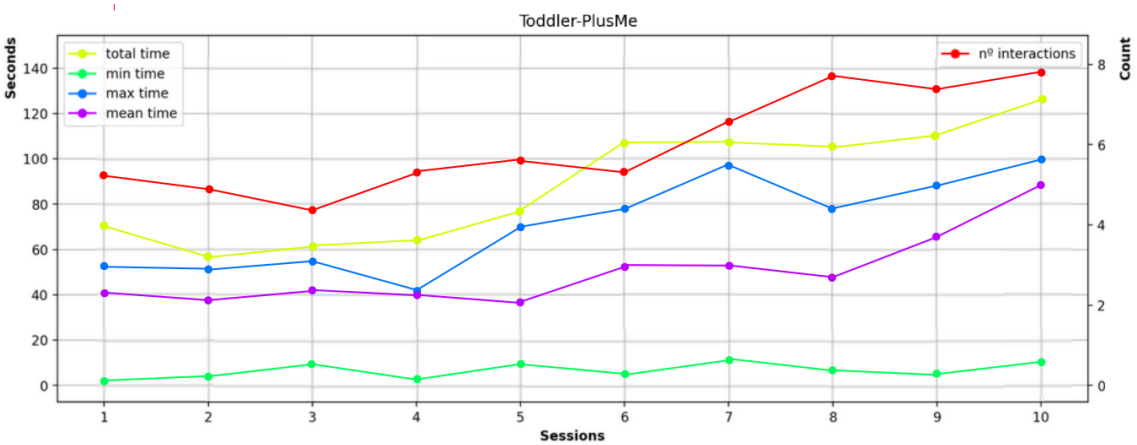
Fonte: O autor.

Figura 43 – Evolução da interação entre Criança-Terapeuta entre sessões.



Fonte: O autor.

Figura 44 – Evolução da interação entre Criança-PlusMe entre sessões.



Fonte: O autor.

6 CONCLUSÃO

O acompanhamento terapêutico é fundamental no tratamento de indivíduos dentro do espectro autista, pois trabalha a melhora da comunicação e interação social, além de promover a independência e autoestima do indivíduo.

A presença do terapeuta nas sessões é muito importante para coleta de informações e análise de progresso do paciente. Porém, essa forma manual de se extrair informações comportamentais atrasa a evolução do tratamento pois algumas informações são sutis demais para serem percebidas pelo olho humano, não sendo incluídas na análise geral do paciente. Sendo outro fator limitante a falta de padronização desses registros num banco de dados interligado.

Pensando nesse problema, esse trabalho propôs a construção de uma ferramenta que auxilie terapeutas na análise de sessões de terapia do TEA. Por meio do uso de aprendizado de máquina e visão computacional na detecção de eventos de interesse, o objetivo é promover a redução do tempo necessário para o estudo de interações importantes nessas seções e proporcionar aos pacientes com TEA intervenções terapêuticas mais focadas e um melhor tratamento.

Para isso, anotamos 819 quadros de vídeos de sessões de terapia do TEA com 8 crianças diferentes, onde esses vídeos sempre apresentam uma Criança, um Terapeuta e o ursinho PlusMe. Utilizamos caixas delimitadoras como anotação para os atores e depois treinamos vários modelos de redes neurais de detecção de objetos com o framework da YOLOv5.

Depois que o melhor modelo foi escolhido, processamos cada um dos quadros detectados do conjunto de teste com uma heurística simples de sobreposição de caixas delimitadoras para predição de interação entre os três atores, sendo elas as interações Criança-Terapeuta, Criança-PlusMe e Terapeuta-Plusme.

No conjunto de teste mostramos que a detecção de eventos de interesse é capaz de reduzir o tempo de análises manuais entre 9 a 90% do tempo total do vídeo, considerando a relação entre redução de tempo de análise, nível de interação e desempenho de detecção, que pode trazer uma redução significativa da carga de trabalho para os especialistas em saúde.

Durante a predição de eventos nesse conjunto, o que mais causou déficites de desempenho foi a oclusão entre atores, se refletindo em falsos positivos durante a avaliação do sistema. Como a heurística é baseada em sobreposição de bounding boxes em plano 2D, um ator em frente ao outro não se adéqua à proposta.

Ao variarmos o limiar de decisão da heurística de sobreposição de caixas delimitadoras, foi possível a realização de um estudo de tendência das métricas, que mostrou que o melhor threshold para as sobreposições deve ser escolhido após um estudo de equilíbrio entre Precisão e Recall, o que vai depender do terapeuta e da análise que estiver fazendo.

A fim de promover valores altos de métricas nos resultados, o uso do modelo extra grande é o mais indicado, pois como vimos, o grande fator que diminui o desempenho do preditor são os erros da YOLO. Porém, modelos menores podem ser usados para aumentar a velocidade de processamento.

O limitador principal na qualidade dos modelos da YOLO foi o conjunto de dados. A anotação manual dos frames impediu um maior avanço nessa parte, gerando apenas 819 frames anotados. Uma grande quantidade de frames beneficiaria o treinamento gerando uma variação natural nos dados que aumentaria a convergência e diminuiria overfitting.

Ao criarmos uma timeline de eventos em uma plataforma web, como uma composição entre interações e proximidade, entregamos ao médico uma visão geral do que está acontecendo em todos os momentos do vídeo. Permitindo dessa forma, a investigação de momentos anteriores e posteriores de uma interação que podem ser tão importantes quanto o principal evento de interesse.

6.1 TRABALHOS FUTUROS

Para trabalhos futuros, podemos estender o número de frames anotados e a diversidade dos mesmos para termos um melhor desempenho no conjunto de validação e teste. Podemos também utilizar técnicas de validação cruzada como *K-Fold* e afins. Além disso podemos treinar outras arquiteturas de rede neural de detecção, como a *R-CNN*, *SqueezeDet* e *MobileNet*.

Escolher um pequeno conjunto de métricas facilitaria o estudo do desempenho do sistema. Vimos que todos os *plots* das medidas de matriz de confusão e métricas nos mostram o problema por vários ângulos, por isso que, ao definirmos a melhor métrica ou conjunto de métricas ganharíamos muito em questão de estudo e planejamento.

Ainda utilizando a YOLO, podemos treinar os modelos utilizando anotações das interações em vez de detectar os atores e indiretamente gerar as interações com threshold.

Além de modelos de detecção de bounding boxes, podemos expandir a pesquisa com modelos de esqueletização e *face tracking*. Com essses modelos, podemos analisar a movimentação corporal e de membros, *tracking* do olhar e repetição de movimentos, aumentando as possibilidades de eventos de interesse, não ficando limitados apenas a toques.

Escolher um conjunto de dados já anotado, que envolva sessões de terapia e pacientes, ajudaria bastante na etapa de treinamento dos modelos, bastando apenas um *fine-tunning* para nossos propósitos.

E por fim, como um dos maiores problemas enfrentados foi o de oclusão, em trabalhos futuros podemos gerar um conjunto de dados próprio, com o apoio do grupo de pesquisa italiano, e adicionar mais ângulos de câmeras e realizando leituras do retorno tátil disponibilizado em futuras versões do ursinho PlusMe.

7 CONSIDERAÇÕES FINAIS

Esse projeto se mostrou muito interessante, e desde o começo apresentou um desafio bem claro à frente, auxiliar médicos e terapeutas na tomada de decisões em sessões de terapia do Transtorno do Espectro Autista. E com sucesso foi possível entregar uma versão estável da ferramenta que pode ser usada com facilidade.

Contruído com o framework Streamlit em cima da linguagem Python, a ferramenta web está disponível em repositório público e pode ser acessada e modificada (em cópia pessoal) dentro da filosofia open-source. Apresentando um sistema CRUD, o usuário pode processar seus vídeos e acessá-los sem a necessidade de rodar o modelo de deep learning novamente. E agrupando por nome de paciente, o usuário pode gerar dashboards evolutivos em relações a todas as sessões em que esse paciente está presente.

Foi uma experiência muito desafiadora e enriquecedora ao lado de um time incrível de orientadores e colegas, e estamos muito felizes de finalizar essa primeira etapa do projeto.

⋆ Parte dos resultados desse trabalho foram publicados em artigo (RIBEIRO; SOARES; CARVALHO; GRELLERT, 2022) no evento *Brazilian Conference on Intelligent Systems 2022* (BRACIS), onde o autor deste projeto de TCC é um dos autores, e também nos anais do *Computer on the Beach 2023* como primeiro autor (SOARES; CARVALHO; RIBEIRO; GRELLERT, 2023).

REFERÊNCIAS

ABDULLAH, Sharmeen M Saleem; ABDULAZEEZ, Adnan Mohsin. Facial expression recognition based on deep learning convolution neural network: A review. **Journal of Soft Computing and Data Mining**, v. 2, n. 1, p. 53–65, 2021.

AYODELE, Taiwo Oladipupo. Types of machine learning algorithms. **New advances in machine learning**, University of Portsmouth UK, v. 3, p. 19–48, 2010.

BERTAMINI, Giulio *et al.* Quantifying the Child–Therapist Interaction in ASD Intervention: An Observational Coding System. **Brain Sciences**, MDPI AG, v. 11, n. 3, p. 366, mar. 2021. DOI: 10.3390/brainsci11030366. Disponível em: https://doi.org/10.3390/brainsci11030366.

BI, Qifang *et al.* What is Machine Learning? A Primer for the Epidemiologist. en. **Am. J. Epidemiol.**, Oxford University Press (OUP), v. 188, n. 12, p. 2222–2239, dez. 2019.

_____. What is machine learning? A primer for the epidemiologist. **American journal of epidemiology**, Oxford University Press, v. 188, n. 12, p. 2222–2239, 2019.

CABIBIHAN, John-John *et al.* Why robots? A survey on the roles and benefits of social robots in the therapy of children with autism. **International journal of social robotics**, Springer, v. 5, n. 4, p. 593–618, 2013.

CARREIRA-PERPINÁN, Miguel A. A review of dimension reduction techniques. **Department of Computer Science. University of Sheffield. Tech. Rep. CS-96-09**, v. 9, p. 1–69, 1997.

CHATTERJEE, Poulomi. **How to detect and prevent overfitting in a model?** 2022. Disponível em: https://analyticsindiamag.com/how-to-detect-and-prevent-overfitting-in-a-model/. Acesso em: 11 jan. 2023.

CHENG, Yu *et al.* Occlusion-aware networks for 3d human pose estimation in video. *In:* PROCEEDINGS of the IEEE/CVF international conference on computer vision. [*S.l.*: *s.n.*], 2019. P. 723–732.

COMASCHI, Francesco *et al.* RASW: a run-time adaptive sliding window to improve viola-jones object detection. *In:* IEEE. 2013 Seventh International Conference on Distributed Smart Cameras (ICDSC). [*S.l.*: *s.n.*], 2013. P. 1–6.

DALIANIS, Hercules. Evaluation Metrics and Evaluation. *In:* CLINICAL Text Mining: Secondary Use of Electronic Patient Records. Cham: Springer International Publishing, 2018. P. 45–53. ISBN 978-3-319-78503-5. DOI: 10.1007/978-3-319-78503-5_6. Disponível em: https://doi.org/10.1007/978-3-319-78503-5_6.

DERTAT, Arden. **Applied Deep Learning - Part 4: Convolutional Neural Networks**. 2017. Disponível em: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2. Acesso em: 11 jan. 2023.

E, Swapna K. **Convolutional Neural Network | Deep Learning**. 2020. Disponível em: https://developersbreach.com/convolution-neural-network-deep-learning/. Acesso em: 11 jan. 2023.

EDITION, Fifth *et al.* Diagnostic and statistical manual of mental disorders. **Am Psychiatric Assoc**, v. 21, p. 591–643, 2013.

GIOCONDO, Flora *et al.* Leveraging Curiosity to Encourage Social Interactions in Children with Autism Spectrum Disorder: Preliminary Results Using the Interactive Toy PlusMe. *In:* EXTENDED Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems. New Orleans, LA, USA: Association for Computing Machinery, 2022. (CHI EA '22). DOI: 10.1145/3491101.3519716. Disponível em: https://doi.org/10.1145/3491101.3519716.

GM, Harshvardhan *et al.* A comprehensive survey and analysis of generative models in machine learning. **Computer Science Review**, v. 38, p. 100285, 2020. ISSN 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2020.100285. Disponível em: https://www.sciencedirect.com/science/article/pii/S1574013720303853.

GORDON, Kurtiss J. Spreadsheet or database: Which makes more sense? **Journal of Computing in Higher Education**, Springer, v. 10, p. 111–116, 1999.

GOULD, Stephen; GAO, Tianshi; KOLLER, Daphne. Region-based segmentation and object detection. **Advances in neural information processing systems**, v. 22, 2009.

HAILPERN, Joshua *et al.* A3: HCI Coding Guideline for Research Using Video Annotation to Assess Behavior of Nonverbal Subjects with Computer-Based Intervention. **ACM Trans. Access. Comput.**, Association for Computing Machinery, New York, NY, USA, v. 2, n. 2, jun. 2009. ISSN 1936-7228. DOI: 10.1145/1530064.1530066. Disponível em: https://doi.org/10.1145/1530064.1530066.

HOSSIN, Mohammad; SULAIMAN, Md Nasir. A review on evaluation metrics for data classification evaluations. **International journal of data mining & knowledge management process**, Academy & Industry Research Collaboration Center (AIRCC), v. 5, n. 2, p. 1, 2015.

JOCHER, Glenn *et al.* **ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation**. [*S.l.*]: Zenodo, 2022.

JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. **Science**, v. 349, n. 6245, p. 255–260, 2015. DOI: 10.1126/science.aaa8415. eprint: https://www.science.org/doi/pdf/10.1126/science.aaa8415. Disponível em: https://www.science.org/doi/abs/10.1126/science.aaa8415.

KOŁAKOWSKA, Agata *et al.* Automatic recognition of therapy progress among children with autism. **Scientific Reports**, Springer Science e Business Media LLC, v. 7, n. 1, out. 2017. DOI: 10.1038/s41598-017-14209-y. Disponível em: https://doi.org/10.1038/s41598-017-14209-y.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.

LIN, David Chuan-En. **8 Simple Techniques to Prevent Overfitting**. 2020. Disponível em: https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d. Acesso em: 11 jan. 2023.

LU, Jia; NGUYEN, Minh; YAN, Wei Qi. Deep learning methods for human behavior recognition. *In:* IEEE. 2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ). [*S.l.*: *s.n.*], 2020. P. 1–6.

LUKETINA, Jelena *et al.* A survey of reinforcement learning informed by natural language. **arXiv preprint arXiv:1906.03926**, 2019.

MAENNER, Matthew J. *et al.* Prevalence and Characteristics of Autism Spectrum Disorder Among Children Aged 8 Years — Autism and Developmental Disabilities Monitoring Network, 11 Sites, United States, 2018. **MMWR. Surveillance Summaries**, Centers for Disease Control MMWR Office, v. 70, n. 11, p. 1–16, dez. 2021. DOI: 10.15585/mmwr.ss7011a1. Disponível em: https://doi.org/10.15585/mmwr.ss7011a1.

NIGAM, Swati; SINGH, Rajiv; MISRA, A. K. A Review of Computational Approaches for Human Behavior Detection. **Archives of Computational Methods in Engineering**, Springer Science e Business Media LLC, mai. 2018. DOI: 10.1007/s11831-018-9270-7. Disponível em: https://doi.org/10.1007/s11831-018-9270-7.

NOGAY, Hidir Selcuk; ADELI, Hojjat. Machine learning (ML) for the diagnosis of autism spectrum disorder (ASD) using brain imaging. **Reviews in the Neurosciences**, De Gruyter, v. 31, n. 8, p. 825–841, 2020.

OPREA, Sergiu *et al.* A review on deep learning techniques for video prediction. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, 2020.

PADILLA, Rafael; NETTO, Sergio L; DA SILVA, Eduardo AB. A survey on performance metrics for object-detection algorithms. *In:* IEEE. 2020 international conference on systems, signals and image processing (IWSSIP). [*S.l.*: *s.n.*], 2020. P. 237–242.

PARIKH, Milan N; LI, Hailong; HE, Lili. Enhancing diagnosis of autism with optimized machine learning models and personal characteristic data. **Frontiers in computational neuroscience**, Frontiers, v. 13, p. 9, 2019.

PATEL, Ashish. **What is Object Detection?** 2020. Disponível em: `https://medium.com/ml-research-lab/what-is-object-detection-51f9d872ece7`. Acesso em: 11 jan. 2023.

POLYDOROS, Athanasios S; NALPANTIDIS, Lazaros. Survey of model-based reinforcement learning: Applications on robotics. **Journal of Intelligent & Robotic Systems**, Springer, v. 86, n. 2, p. 153–173, 2017.

RAMIREZ-DUQUE, Andrés A.; FRIZERA-NETO, Anselmo; BASTOS, Teodiano Freire. Robot-Assisted Diagnosis for Children with Autism Spectrum Disorder Based on Automated Analysis of Nonverbal Cues. *In:* 2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob). [*S.l.*: *s.n.*], 2018. P. 456–461. DOI: `10.1109/BIOROB.2018.8487909`.

RASCHKA, Sebastian; PATTERSON, Joshua; NOLET, Corey. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. **Information**, Multidisciplinary Digital Publishing Institute, v. 11, n. 4, p. 193, 2020.

REDMON, J. *et al.* You Only Look Once: Unified, Real-Time Object Detection. *In:* 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA, USA: IEEE Computer Society, jun. 2016. P. 779–788. DOI: `10.1109/CVPR.2016.91`. Disponível em: `https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.91`.

RIBEIRO, Guilherme Ocker *et al.* Event Detection in Therapy Sessions for Children with Autism. *In:*_____. **Intelligent Systems**. Cham: Springer International Publishing, 2022. P. 221–235.

ROGGE, Nicky; JANSSEN, Juliette. The Economic Costs of Autism Spectrum Disorder: A Literature Review. **Journal of Autism and Developmental Disorders**, Springer Science e Business Media LLC, v. 49, n. 7, p. 2873–2900, abr. 2019. DOI: `10.1007/s10803-019-04014-z`. Disponível em: `https://doi.org/10.1007/s10803-019-04014-z`.

SHARMA, Samata R; GONDA, Xenia; TARAZI, Frank I. Autism spectrum disorder: classification, diagnosis and therapy. **Pharmacology & therapeutics**, Elsevier, v. 190, p. 91–104, 2018.

SILVA, Yasin N; ALMEIDA, Isadora; QUEIROZ, Michell. SQL: From traditional databases to big data. *In:* PROCEEDINGS of the 47th ACM Technical Symposium on Computing Science Education. [*S.l.*: *s.n.*], 2016. P. 413–418.

SINAGA, Kristina P; YANG, Miin-Shen. Unsupervised K-means clustering algorithm. **IEEE access**, IEEE, v. 8, p. 80716–80727, 2020.

SOARES, Alexandre S Soli *et al.* Event detection in therapy sessions for children with Autism. **Anais do Computer on the Beach**, v. 14, p. 479–480, 2023.

SPERATI, Valerio. **PlusMe functional features**. 2020. Disponível em: `https://www.plusme-h2020.eu/video/`. Acesso em: 11 jan. 2023.

SPERATI, Valerio; ÖZCAN, Beste *et al.* Acceptability of the Transitional Wearable Companion "+me" in Typical Children: A Pilot Study. **Frontiers in Psychology**, v. 10, 2019. ISSN 1664-1078. DOI: `10.3389/fpsyg.2019.00125`. Disponível em: `https://www.frontiersin.org/articles/10.3389/fpsyg.2019.00125`.

_____._____. **Frontiers in Psychology**, v. 10, 2019. ISSN 1664-1078. DOI: `10.3389/fpsyg.2019.00125`. Disponível em: `https://www.frontiersin.org/articles/10.3389/fpsyg.2019.00125`.

SRINATH, KR. Python–the fastest growing programming language. **International Research Journal of Engineering and Technology**, v. 4, n. 12, p. 354–357, 2017.

THEVENOT, Jérôme; LÓPEZ, Miguel Bordallo; HADID, Abdenour. A Survey on Computer Vision for Assistive Medical Diagnosis From Faces. **IEEE Journal of Biomedical and Health Informatics**, v. 22, n. 5, p. 1497–1511, 2018. DOI: `10.1109/JBHI.2017.2754861`.

TORREY, Lisa; SHAVLIK, Jude. Transfer learning. *In:* HANDBOOK of research on machine learning applications and trends: algorithms, methods, and techniques. [*S.l.*]: IGI global, 2010. P. 242–264.

WEISS, Karl; KHOSHGOFTAAR, Taghi M; WANG, Dingding. A survey of transfer learning. en. **J. Big Data**, Springer Science e Business Media LLC, v. 3, n. 1, dez. 2016.

WU, Hao; LIU, Qi; LIU, Xiaodong. A review on deep learning approaches to image classification and object segmentation. **Comput. Mater. Continua**, v. 60, n. 2, p. 575–597, 2019.

XU, Yun; GOODACRE, Royston. On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. en. **J. Anal. Test.**, Springer Science e Business Media LLC, v. 2, n. 3, p. 249–262, out. 2018.

ZHAO, Zhong-Qiu *et al.* Object detection with deep learning: A review. **IEEE transactions on neural networks and learning systems**, IEEE, v. 30, n. 11, p. 3212–3232, 2019.