

ESTIMADOR AUTOMÁTICO POR IMAGENS DO NÚMERO DE PASSAGEIROS PARA TRANSPORTE COLETIVO¹

Eduardo Compadre Espadas²

Graduando em Bacharelado de Engenharia Eletrônica
IFSP/Campus São Paulo

Gustavo Domene Saboia³

Graduando em Bacharelado de Engenharia Eletrônica
IFSP/Campus São Paulo

Ricardo Pires⁴

Doutor em Sistemas Automáticos e
Microeletrônicos/Université de Montpellier II
Docente do Departamento de Elétrica
IFSP/Campus São Paulo

RESUMO

O aumento da população urbana evidencia a importância do transporte público, o qual possibilita o deslocamento com baixo custo, com pequenos níveis de emissão de carbono por passageiro, em comparação ao transporte individual. O processamento de imagens possibilita o desenvolvimento de um sistema de transporte inteligente, o qual, a partir de informações coletadas em tempo real, informará aos usuários e aos administradores a sua situação de ocupação a cada momento. Este trabalho propõe a utilização de processamento de imagens para a estimativa, em tempo real, do número de passageiros num vagão de trem urbano ou de metrô. Para isso, ele investiga o uso de um programa, o YOLO, para a contagem automática de pessoas em imagens e o uso de uma técnica de medição da diferença entre imagens de um vagão vazio e imagens do mesmo vagão contendo passageiros. Foi criado um banco de imagens sintéticas para testes, formado por exemplos contendo números variados de pessoas dentro do vagão. O YOLO foi aplicado às imagens e foi registrado o número de pessoas que ele detectou em cada uma delas. Para comparação, foi medida a diferença, por meio da norma L2 dos valores dos pixels, entre cada uma daquelas imagens e imagens do vagão vazio, buscando-se relacioná-la ao número de pessoas. Dos resultados experimentais, conclui-se que, para esta finalidade, a medição da diferença entre imagens é vantajosa em relação à identificação individual de pessoas nas imagens pelo YOLO, devido à obstrução das imagens de uns passageiros pelas imagens de outros, em um vagão muito ocupado. O resultado da medição da diferença apresentou boa correlação com o número real de passageiros no vagão.

Palavras chave: YOLO; número de passageiros; processamento de imagens; transporte público.

¹ Artigo resultante do Trabalho de Conclusão de Curso de Bacharelado de Engenharia Eletrônica do Instituto Federal de Educação e Tecnologia de São Paulo (IFSP), São Paulo

² eduardo.compadre@gmail.com

³ gustavosaboia32@gmail.com

⁴ ricardo_pires@ifsp.edu.br

ABSTRACT

The increase in the urban population highlights the importance of public transport, which enables low-cost travel, with low levels of carbon emissions per passenger, compared to individual transport. Image processing enables the development of an intelligent transport system, which, based on information collected in real time, will inform users and administrators of their occupancy situation at every moment. This work proposes the use of image processing to estimate, in real time, the number of passengers in an urban train or subway car. To do this, it investigates the use of a program, YOLO, for the automatic counting of people in images and the use of a technique to measure the difference between images of an empty wagon and images of the same wagon containing passengers. A bank of synthetic images was created for testing, made up of examples containing varying numbers of people inside the wagon. YOLO was applied to the images and the number of people it detected in each of them was recorded. For comparison, the difference was measured, using the L2 norm of pixel values, between each of those images and images of the empty carriage, seeking to relate it to the number of people. From the experimental results, it is concluded that, for this purpose, the measurement of the difference between images is advantageous in relation to the individual identification of people in the images by YOLO, due to the obstruction of the images of some passengers by the images of others, in a very crowded wagon. The difference measurement results showed good correlation with the actual number of passengers in the wagon.

Keywords: YOLO; number of passengers; image processing; public transportation.

Introdução

O crescimento da população urbana no Brasil é evidente. Conforme apresentado pelos censos demográficos do IBGE, em 30 anos (1970-2000) a relação entre o número absoluto da população urbana e a população brasileira total passou de aproximadamente 55,94% em 1970 para 81,25% em 2000 (Brito, 2006). Este contexto de aumento expressivo populacional urbano e adensamento das cidades evidencia a importância da mobilidade urbana. Este conceito define basicamente o quão fácil é o deslocamento populacional em uma metrópole. Um plano de mobilidade ineficiente impacta negativamente na qualidade de vida dos cidadãos, pelo aumento nos níveis de congestionamentos e poluição (Costa, 2016). Legalmente, tem-se que a gestão do transporte nas cidades é de responsabilidade municipal, conforme a Constituição de 1988. Em 2012, estabeleceu-se a Lei Nº 12.587, de 3 de janeiro de 2012, que trata sobre mobilidade urbana e estabelece que é papel do estado promover a mobilidade urbana, buscando o desenvolvimento sustentável das cidades nas dimensões socioeconômicas e ambientais.

O transporte coletivo, como o metrô, apresenta demanda crescente, como pode ser observado comparando os indicadores do metrô de São Paulo em 2021 (Metrô de São Paulo, 2021) e 2022 (Metrô de São Paulo, 2023), que indicam uma ampliação de, em média, 700 mil passageiros/dia em dias úteis. O transporte coletivo

se destaca por possibilitar o deslocamento de baixo custo associado a pequenos níveis de emissão de carbono por passageiro (Lerman et al., 2020).

Aliado a isso, tem-se o grande avanço da tecnologia em áreas como as telecomunicações, inteligência artificial e processamento de imagens. A integração entre elas e os meios de transporte pode gerar uma malha de transporte coletivo mais eficiente, rápida e confortável ao usuário.

Serviços como o “Google Maps – Transporte público” permitem a visualização de quão ocupado um ônibus está, permitindo que um potencial passageiro tome a decisão de embarcar ou não. Toda esta integração, porém, é feita com base na utilização de dados anônimos de usuários da própria plataforma (Google, 2023).

Com o gerenciamento assertivo das frotas em serviço, isto é, com base nas informações colhidas em tempo real, um gerenciador central pode, por exemplo, diminuir o intervalo de partida entre os carros de metrô de uma linha.

O processamento de imagens é um instrumento promissor para possibilitar o desenvolvimento de um sistema de transporte inteligente, o qual, a partir de informações coletadas em tempo real, poderá informar aos usuários do sistema e aos seus administradores a sua situação de ocupação a cada momento.

Este trabalho propõe a utilização de processamento de imagens para a estimativa, em tempo real, do número de passageiros num vagão de trem urbano ou de metrô.

Objetivos

Desenvolver um sistema para estimativa automática do número de passageiros no vagão de um trem, por meio de análise de imagens.

Objetivos Específicos

- Identificar automaticamente pessoas numa imagem, com boa acurácia.
- Contar automaticamente as pessoas identificadas.
- Estimar automaticamente o número de pessoas num vagão, a partir das contagens feitas em poucas imagens parciais.

Estudo Bibliográfico

Para o estudo bibliográfico do tema, foi utilizado o Portal de Periódicos da CAPES. Foram feitas três buscas por assunto, com os conjuntos de palavras: “*estimation passenger image*”, “*image passenger counting*” e “*passenger counting deep learning*” buscando artigos e anais de congressos escritos nos últimos 20 anos em português, inglês ou espanhol que estivessem integralmente disponíveis. As buscas chegaram a 157 documentos que passaram por uma primeira análise, pelos autores, de títulos, resumos e datas de publicação. Os resultados que apresentavam em seu escopo a contagem de passageiros em transporte público partindo do processamento de imagens foram selecionados como relevantes para esta análise bibliográfica, estes trabalhos são citados abaixo.

As primeiras abordagens encontradas expõem vários procedimentos para tratamento do sinal da imagem, buscando a identificação e contagem das pessoas. Podem-se citar algoritmos baseados na transformada Hough (Yu; He; Liu; 2007; Mukherjee et al., 2011), fluxo óptico (Bozzoli; Cinque; Sangineto, 2007, Hejin, 2010), identificação de movimento humano com a segmentação da imagem tratada (Ming; Wei, 2009) e no método de Haar-like (Yan-Yan et al., 2014) . Em Lengvenis et al. (2013), os autores propõem a identificação e contagem partindo da medição da variação na intensidade de pixels das imagens internas de um ônibus. Os autores Nakashima, Arai e Fujikawa (2018) cruzam as informações do GPS interno do veículo com a identificação dos passageiros. Nele, a detecção ocorre com base no contorno das imagens tratadas, para a estimativa do número total de ocupantes.

Todos os demais trabalhos encontrados utilizam alguma configuração de rede neural convolucional para a estimativa. Atualmente, esta é a solução mais explorada, visto que foi abordada por todos trabalhos selecionados publicados nos últimos cinco anos. Nos trabalhos de Yang et al. (2018), Mazzeo et al. (2020), Yang et al. (2020), Zhu et al. (2021), Zhang, Zhu e Wang (2020), Zhang, Liu e Wang (2021) e Rendon et al. (2023), há a utilização das imagens coletadas para a geração de um mapa de densidade, o qual, posteriormente, é utilizado na estimativa final do número de passageiros. Outra abordagem é a identificação de objetos com a utilização da arquitetura YOLO. Esta arquitetura é usada nos trabalhos de Nakashima, Arai e Fujikawa (2019), Sipetas, Keklikoglou e Gonzales (2020), Hsu, Chen e Perng (2020), Huang et al. (2022), Sutopo et al. (2021). Os melhores resultados observados são

obtidos quando a cabeça e não o corpo dos passageiros é o objeto da identificação, como demonstrado por Huang et al. (2022). Ainda em menor proporção, pode-se citar a utilização de arquiteturas SSD (*Single-Shot Detector*) para a identificação, como em McCarthy et al. (2021) e Hsu, Wang e Perng (2020) e de SVM (*Support Vector Machine*), como em Haq et al. (2019).

Fundamentação Teórica

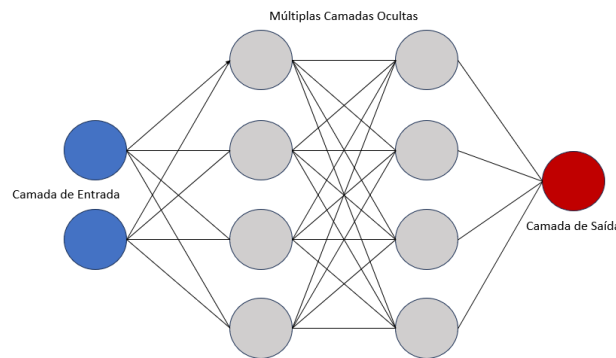
Buscando elucidar o método de solução proposto por este trabalho, são introduzidos, nesta seção, os principais conceitos teóricos utilizados, a serem retomados nas próximas seções. São eles: representação digital de imagens, redes neurais, redes convolucionais e a arquitetura YOLO.

Uma vez que os sistemas de visão computacional atuam sobre as imagens digitais para a constituição de suas análises, parte-se, aqui, de como estas representações digitais são realizadas. A formação de uma imagem digital é a base para o processamento digital de imagens. Pode-se descrever o processo de representação digital como sendo a representação da imagem por um número finito de elementos distribuídos num plano X e Y, cada qual com uma intensidade específica. A estes elementos finitos se estabelece o nome de pixel. Uma das formas de representação digital de imagens coloridas é o modelo RGB (*red, green, blue*). Neste modelo, cada pixel tem três valores de informação que definem a sua representação nas componentes espectrais azul, verde e vermelho. A interação entre as intensidades das componentes gera a cor específica do pixel enxergada pelo usuário do sistema (Gonzalez; Woods, 2007).

Uma das formas da análise de imagens digitais pelos sistemas de visão computacional é utilizando redes neurais. As redes neurais artificiais (ANNs) ou redes neurais simuladas são modelos computacionais inspirados pelo sistema nervoso biológico, compostas por unidades interconectadas chamadas neurônios artificiais. Cada neurônio artificial recebe um conjunto de valores de entrada, realiza um cálculo a partir delas, na forma de uma soma ponderada seguida pela aplicação de uma função não linear, e produz um valor de saída. Essas saídas são transmitidas para outros neurônios por meio de conexões ponderadas, formando uma rede interconectada (Woods, 2007).

Uma rede neural é organizada em camadas, geralmente divididas em três tipos: camada de entrada, camadas ocultas e camada de saída. A camada de entrada recebe os dados a serem classificados e os transmite para as camadas ocultas, que realizam cálculos e extraem características dos dados. A camada de saída produz as previsões ou resultados finais da rede. Um exemplo de estrutura da rede neural se encontra na Figura 1.

Figura 1: Exemplo de Rede Neural. Cada círculo representa um neurônio.



Fonte: Autores

Cada conexão entre os neurônios possui um peso associado, que é ajustado durante o treinamento da rede para capturar os padrões relevantes nos dados. O processo de treinamento envolve a apresentação dos dados de treinamento à rede, o cálculo do erro entre as previsões da rede e os valores esperados e, em seguida, o ajuste dos pesos das conexões para minimizar o erro. Esse ciclo se repete, até que a rede esteja bem ajustada aos dados de treinamento. Cada ciclo de apresentação à rede de todos os dados de treinamento é chamada de época (Haykin, 2008).

As redes neurais podem ter diferentes arquiteturas, como redes neurais *feedforward*, redes neurais convolucionais (CNNs) e redes neurais recorrentes (RNNs), cada uma adequada para diferentes tipos de problemas. Elas são capazes de aprender automaticamente a partir dos dados, descobrindo padrões complexos e realizando tarefas como classificação, regressão, reconhecimento de padrões e muito mais (Haykin, 2008; Li et al., 2022).

As redes neurais convolucionais (CNN) são um tipo especializado de arquitetura de rede neural amplamente utilizada em tarefas de processamento de imagens e visão computacional. Elas foram inspiradas pela organização do córtex visual do cérebro humano e se mostraram altamente eficazes na extração de

características e no reconhecimento de padrões em imagens (Yamashita, 2018; Li et al, 2022).

A principal característica das CNNs é a presença de camadas convolucionais, que consistem em um conjunto de filtros convolucionais. Esses filtros são pequenas matrizes de pesos que deslizam sobre a imagem de entrada, realizando operações de convolução para extrair características locais em cada posição. Essas características incluem bordas, texturas e outras informações relevantes presentes nas imagens (Li et al, 2022).

Durante a convolução, os filtros multiplicam seus pesos pelos valores dos pixels da imagem em uma janela de convolução, e o resultado é somado para produzir um mapa de características. A operação de convolução é repetida em diferentes posições da imagem, gerando vários mapas de características que capturam diferentes aspectos das imagens (Li et al, 2022). Após as camadas convolucionais, as CNNs geralmente têm camadas de subamostragem, que reduzem a dimensionalidade dos mapas de características, mantendo as informações mais relevantes. Isso ajuda a tornar as CNNs invariantes a pequenas variações na posição dos objetos nas imagens. As camadas convolucionais e de subamostragem são seguidas por uma ou mais camadas totalmente conectadas, que atuam como classificadores finais ou realizam outras tarefas específicas, como detecção de objetos ou segmentação de imagens. Essas camadas combinam as informações das características extraídas anteriormente e produzem as previsões ou resultados finais.

Uma das principais vantagens das CNNs em relação às redes neurais convencionais é a capacidade de aprender automaticamente a extração das características relevantes para uma determinada tarefa. Durante o treinamento, as CNNs ajustam os pesos dos filtros convolucionais por meio de algoritmos de otimização, como descida de gradiente, para minimizar uma função de perda que mede a diferença entre as previsões da rede e os valores esperados.

As CNNs têm sido amplamente utilizadas em várias aplicações, incluindo reconhecimento facial, classificação de imagens, detecção de objetos, segmentação semântica, entre outros. Elas são especialmente eficazes em tarefas de visão computacional, pois são capazes de aprender e extrair automaticamente as características mais discriminativas das imagens, permitindo uma representação mais rica e poderosa dos dados visuais (Li et al, 2022).

Deep learning é uma abordagem de aprendizado de máquina que se baseia em redes neurais artificiais profundas para extrair e aprender representações complexas dos dados. O termo *deep* refere-se ao fato de que essas redes neurais possuem várias camadas ocultas, permitindo que elas aprendam gradualmente características cada vez mais abstratas, à medida que dados avançam nas camadas. Essas redes são capazes de modelar relacionamentos complexos nos dados, tornando-se especialmente eficazes em problemas de grande escala e alta dimensionalidade, como processamento de imagens e reconhecimento de fala (Chagas, 2019).

Um algoritmo de detecção de objetos amplamente utilizado é o YOLO (You Only Look Once). Ao contrário dos métodos tradicionais que exigem a aplicação de um classificador em várias regiões de uma imagem, o YOLO adota uma abordagem única, que possibilita a detecção de objetos em tempo real com alta eficiência computacional (Bochkovskiy; Wang; Liao, 2020).

O YOLO utiliza uma arquitetura baseada em redes neurais convolucionais para realizar a tarefa de detecção de objetos. Ao passar pela imagem de entrada, a rede neural é capaz de detectar objetos em uma única iteração. Essa abordagem eficiente é possível devido ao uso de uma grade que divide a imagem em várias células e atribui a cada célula a responsabilidade de detectar objetos. Cada célula da grade prevê um conjunto de caixas delimitadoras e as probabilidades associadas a essas caixas conterem certos objetos. Além disso, são previstas as classes dos objetos presentes nas caixas delimitadoras. Com essa abordagem, o YOLO pode detectar múltiplos objetos simultaneamente, permitindo o reconhecimento e localização de objetos em uma única passagem pela rede (Liao, 2020).

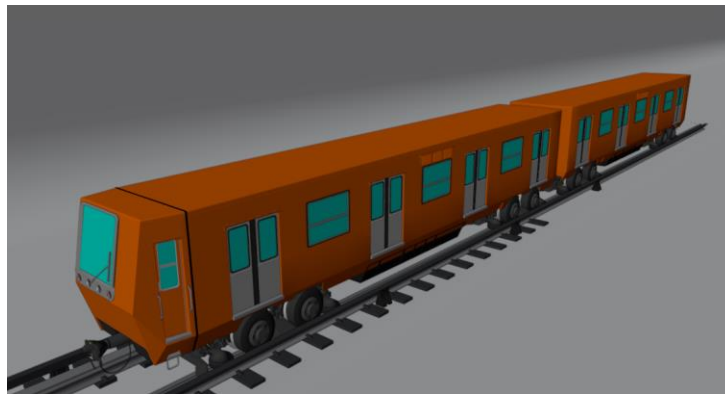
A eficiência do YOLO se deve em parte à sua capacidade de aprender representações ricas de características visuais. As camadas convolucionais da rede neural extraem características relevantes da imagem, como formas, texturas e padrões, fornecendo informações essenciais para a detecção de objetos (Bochkovskiy; Wang; Liao, 2020). O algoritmo YOLO foi desenvolvido com foco em aplicações em tempo real, como sistemas de vigilância, veículos autônomos e análise de vídeo. Sua velocidade e acurácia tornaram-no uma escolha popular para muitas aplicações que exigem detecção de objetos em tempo real (Li et al., 2022).

Desenvolvimento

Este trabalho utilizou dados experimentais para desenvolver um estimador do número de passageiros num vagão de trem. Devido a dificuldade de se encontrar imagens públicas de um vagão de trem urbano ocupado, bem como o número de passageiros presentes, além de haver possíveis problemas relacionados à privacidade de pessoas aparecendo em imagens usadas para experimentos, este trabalho utilizou o software Blender como ferramenta para criação de cenários virtuais. O programa YOLO foi usado como ferramenta que conta automaticamente as pessoas nas imagens.

O Blender é um programa livre, de código aberto, para modelagem tridimensional e animação (Blender Foundation, 2023). Para as imagens, foi utilizado um modelo 3D para simular o trem com passageiros, conforme Figura 2. Esse modelo foi obtido do sítio Blend Swap, onde ele está sob o nome Mexico City Subway, disponibilizado pelo usuário sob o pseudônimo materielpneu (Blend Swap, 2023).

Figura 2: Modelo 3D de trem.

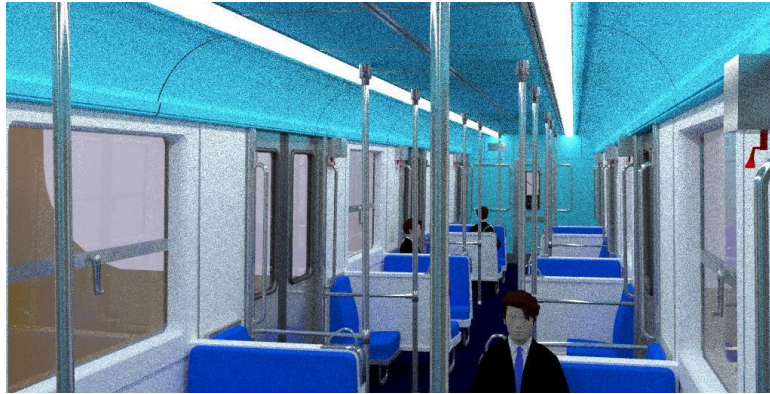


Fonte: Autores, a partir de modelo publicado por materielpneu (Blend Swap, 2023).

As imagens do interior de um dos vagões foram geradas de duas posições de câmeras e com 18 diferentes números de passageiros presentes para cada posição de câmera. O software Blender foi utilizado para modelar e renderizar os cenários virtuais do interior do vagão. O modelo possui detalhes, como assentos, portas e janelas realistas. Foram inseridos modelos de corpos humanos, sempre distribuídos no vagão de maneira aproximadamente uniforme, buscando-se simular o que ocorre na realidade, sem a preocupação de eles estarem no campo de visão das câmeras. Foram geradas imagens com resoluções de 1024 x 512 pixels para treinamento e

teste do modelo. Gerando-se as imagens artificialmente, sempre se sabe qual é o número exato de passageiros no vagão, em cada caso, incluindo-se aqueles que estejam fora do campo de visão das câmeras. Exemplos dessas imagens estão na Figura 3 e na Figura 4.

Figura 3: Exemplo de imagem de vagão em 3D com passageiros, visão da primeira câmera.



Fonte: Autores

Figura 4: Exemplo de imagem de vagão em 3D com passageiros, visão da segunda câmera.



Fonte: Autores

Para a utilização do YOLO, é necessário usar uma linguagem de programação de alto nível, como Python. Por conta disso, foi utilizada a plataforma Google Colab, que possibilita o desenvolvimento de programas na linguagem Python e sua execução em recursos de hardware poderosos (Google Colab, 2023), como GPUs (*Graphics Processing Units*). Nele, foram importadas as bibliotecas principais para a utilização do YOLO. Após a instalação do programa no Google Colab, foi realizado um teste

padrão, com duas imagens, com uma rede pré-treinada, no qual buscava-se verificar como ele se comportava. Com esse teste, ficou claro que a YOLOv5 (YOLO versão 5) conseguia identificar muito mais do que apenas pessoas. Ele também identificou outros objetos para os quais estava pré-treinado, como gravatas. Para resolver esse problema, adicionou-se um filtro dentro do arquivo detect.py do YOLOv5, com o intuito de detectar apenas as pessoas.

Após o teste inicial, utilizou-se o YOLO para contar o número de passageiros numa daquelas imagens internas de vagão de trem. O treinamento do modelo foi conduzido utilizando o banco de dados COCO, que consiste em uma coleção de imagens anotadas. O banco de imagens do treinamento possui cerca de 5000 exemplos. O treinamento foi realizado, em um primeiro momento, durante três épocas e, posteriormente, durante cinco, em que foi identificada uma melhor evolução dos resultados de reconhecimento.

Com os resultados obtidos com o YOLO, ficou claro que ele tinha dificuldade em identificar um grande número de pessoas quando elas estão aglomeradas, devido à obstrução de partes das imagens de umas pessoas por outras à sua frente.

Por conta disso, foi criado um programa em Python que utiliza a biblioteca OpenCV, para comparar todos os pixels de uma imagem do vagão vazio e de outra imagem deste mesmo vagão ocupado com quantidades variáveis de passageiros, sob a mesma visão de câmera. A OpenCV é uma biblioteca computacional livre, que contém implementações de muitos algoritmos para processamento de imagens (Opencv Team, 2023). O programa de cálculo da diferença mede a norma L2 da diferença entre cada imagem com passageiros e a imagem sem passageiros, sob a mesma câmera, usando uma função da OpenCV, chamada *norm*. O cálculo consiste na raiz quadrada da soma dos quadrados das diferenças entre os valores dos pixels de uma imagem e os pixels nas posições correspondentes na outra imagem.

Um teste desse programa foi realizado com todas as imagens e posições de câmeras geradas pelo Blender, em relação a imagens do vagão vazio. Com base nas diferenças de pixels calculadas, a Tabela 1 foi construída, onde cada linha contém a norma L2 da diferença de pixels em relação à imagem do vagão vazio e o número correspondente de pessoas presentes na imagem.

Tabela 1: Valores das contagens de passageiros nas imagens pelo YOLO e da norma L2 da diferença entre as imagens do vagão para cada número real de passageiros e a imagem do vagão vazio, sob a mesma posição da câmera.

número real	Câmera 1		Câmera 2		Média das Normas
	YOLO	norma L2	YOLO	norma L2	
3	2	34559	1	32438	33499
5	5	22625	1	40737	31681
7	5	36350	1	32608	34479
10	8	31800	2	49859	40830
13	8	48678	2	34131	41405
17	10	51171	2	50302	50737
25	13	59412	4	53332	56372
30	17	58775	5	66430	62603
37	16	63193	6	64465	63829
42	12	61336	9	71719	66527
47	11	78654	5	93350	86002
57	10	92313	7	97087	94700
75	8	102201	4	123141	112671
81	1	107565	2	124745	116155
87	1	107565	2	124745	116155
93	1	107565	2	124745	116155

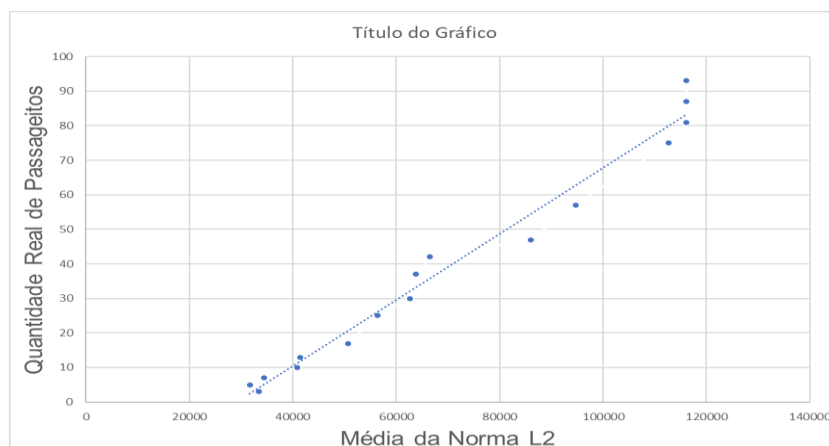
Fonte: Autores

Na Tabela 1, vê-se que há uma tendência de a norma L2 da diferença entre cada imagem e a imagem do vagão vazio aumentar com o número de passageiros. Também se vê que, a partir de 81 passageiros, há uma saturação no valor da norma L2, ou seja, a norma para de aumentar. Quanto à contagem feita pelo YOLO, devido ao problema de obstrução de imagens de uns passageiros por outros, vê-se, na Tabela 2, que ela não apresenta um comportamento simples em relação ao número de passageiros. Vê-se que o número detectado pelo YOLO cresce até números reais de passageiros até as faixas de 30 ou 40 e passa a decrescer para maiores números

reais, devido ao fato de que, para muitos passageiros, as obstruções, mesmo que parciais, prejudicam a detecção. (Deve-se observar que o número real de passageiros tende a ser maior do que o número detectado pelo YOLO, também, pelo fato de que há passageiros reais fora do campo de visão das câmeras.) Portanto, prosseguiu-se o trabalho explorando-se exclusivamente a norma L2 da diferença de imagens, não mais a contagem pelo YOLO.

A Figura 5 é um gráfico do número real de passageiros em função da média das normas L2, usando os dados da Tabela 1.

Figura 5: Gráfico do número real de passageiros em função da média das normas L2, usando os dados da Tabela 1.



Fonte: Autores

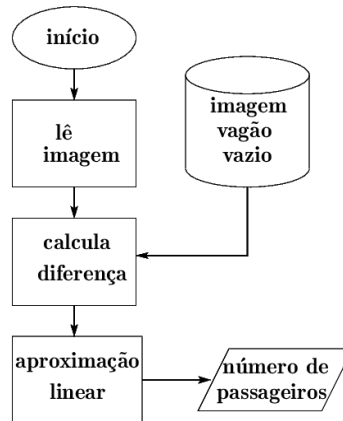
No gráfico da Figura 5, também é indicada uma reta que corresponde à aproximação linear por mínimos quadrados entre as duas grandezas, obtida por meio do programa Microsoft Excel. Essa reta tem a equação:

$$y = 0,00095x - 27,793 \quad (1)$$

em que x representa a norma L2 da diferença entre imagens e y é o número estimado de passageiros no vagão. Ela deve ser considerada válida apenas para normas L2 que estejam entre cerca de 35.000 e 115.000. Para normas L2 abaixo de 35.000, deve-se considerar que o número de passageiros estimado é zero. Para normas acima de 115.000, deve-se considerar, simplesmente, que o número é 81 ou mais (saturação).

Com esse gráfico e a correspondente função em mãos, pode-se estimar um valor máximo de diferença para que o sistema entenda que o vagão está cheio e, então, diminuir o intervalo entre trens de uma malha ferroviária. A Figura 6 é um fluxograma desse processo.

Figura 6: Fluxograma do processo de estimativa do número de passageiros pela diferença entre imagens.



Fonte: Autores.

Tabela 2: Erro Percentual do sistema

número real de passageiros	diferença da norma L2	estimativa de passageiros usando a expressão 1	erro percentual
3	33499	4,0	34%
5	31681	2,3	54%
7	34479	5,0	29%
10	40830	11,0	10%
13	41405	11,5	11%
17	50737	20,4	20%
25	56372	25,8	3%
30	62603	31,7	6%
37	63829	32,8	11%
42	66527	35,4	16%
47	86002	53,9	15%
57	94700	62,2	9%
75	112671	79,2	6%
81	116155	82,6	2%
87	116155	82,6	5%
93	116155	82,6	11%

Fonte : Autores

É importante ressaltar que, por mais que o método da diferença de imagens seja mais acurado, ele tem o problema da saturação, ou seja, a partir de um certo número de passageiros, a diferença de pixels se estabiliza e, com isso, perde-se acurácia. Na Tabela 2, são mostrados os erros percentuais por estimativa de passageiros. Nela, fica claro que o sistema não consegue estimar pequenas quantidades de passageiros. Porém, quanto mais passageiros estão no vagão, maior a acurácia do sistema, além do que é possível que nesses casos em específico, os passageiros estariam aglomerados, ou próximos um do outro, o que diminui a acurácia.

Conclusão

O projeto de estimador do número de passageiros no vagão buscou desenvolver uma solução eficiente e acurada para essa finalidade, usando técnicas de processamento de imagem. Inicialmente, foi explorado o programa YOLO para detecção e contagem de passageiros. No entanto, após se constatar que esse método não fornecia resultados suficientemente acurados, uma nova estratégia foi empregada.

Nesta nova abordagem, de medição da norma L2 da diferença entre a imagem do vagão com passageiros e o vagão vazio, um programa foi desenvolvido utilizando a biblioteca OpenCV.

Ao usar o Blender para gerar imagens e explorar diferentes abordagens, este projeto demonstrou uma combinação eficaz de processamento de imagem e experiência em aprendizado de máquina. A utilização do Google Colab como ambiente de desenvolvimento facilitou a criação e execução de programas, permitindo análises mais detalhadas e otimização dos algoritmos de contagem.

Apesar dos resultados promissores do projeto, vale ressaltar que ainda há espaço para melhorias. Novas tecnologias e algoritmos podem ser explorados para melhorar ainda mais a precisão e a velocidade dos contadores de passageiros. Além disso, um próximo passo pode ser integrar o sistema em um ambiente real de transporte público, permitindo a validação e refinamento da solução em situações reais.

No estudo bibliográfico, foram encontrados trabalhos em que a detecção por YOLO foi combinada com outra técnica. Como trabalho futuro, sugere-se a realização

de experimentos com métodos de aprendizagem de máquina, para se buscar uma relação entre as quatro medições apresentadas na Tabela 1 (duas contagens pelo YOLO e dois cálculos de normas L2) com o número real de passageiros, verificando-se se a incorporação das contagens pelo YOLO proporciona melhores resultados do que o método da diferença isolado.

Referências

Blend Swap. Welcome to Blend Swap! Disponível em <https://blendswap.com/>. Acesso em 11/06/2023.

Blender Foundation. Blender 3.5. Disponível em <https://www.blender.org/>. Acesso em 11/06/2023.

Bochkovskiy, A.; Wang, C.; Liao, H. M. YOLOV4: Optimal speed and accuracy of object detection. **arXiv preprint** arXiv:2004.10934, 2020.

Bozzoli, M.; Cinque, L.; Sangineto, E.. A Statistical Method for People Counting in Crowded Environments. **14Th International Conference On Image Analysis And Processing (Iciap 2007)**, Modena, It, p. 506-511, set. 2007. IEEE. <http://dx.doi.org/10.1109/iciap.2007.4362828>.

Brito, Fausto. O deslocamento da população brasileira para as metrópoles. **Estudos Avançados**, [S.L.], v. 20, n. 57, p. 221-236, ago. 2006. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s0103-40142006000200017>.

Chagas, Edgar Thiago de Oliveira. **Deep Learning e suas aplicações na atualidade**. Revista Científica Multidisciplinar Núcleo do Conhecimento. Ano 04, Ed. 05, Vol. 04, pp. 05-26 Maio de 2019. ISSN: 2448-0959, Link de acesso: <https://www.nucleodoconhecimento.com.br/administracao/deep-learning>

Costa, Marco Aurélio (org.). **O ESTATUTO DA CIDADE E A HABITAT III: um balanço de quinze anos da política urbana no Brasil e a Nova Agenda Urbana**. [S.I.]: Revisão e Editoração Editorar Multimídia, 2016. 368 p. Disponível em: <https://repositorio.ipea.gov.br/bitstream/11058/7121/1/O%20Estatuto%20da%20Cidade%20e%20a%20Habitat%20III.pdf>. Acesso em: 7 jun. 2023.

Gonzalez, R. C.; Woods, R. E.; Digital Image Processing. Pearson. 3 ed. 2007.

Google Colab. Conheça o Colab. Disponível em <https://colab.research.google.com>. Acesso em 14/06/2023.

Google. **Get information about busy areas from Google Maps**. Disponível em: <https://support.google.com/maps/answer/11323117?hl=en#:~:text=To%20protect%2>

Our%20users'%20privacy,which%20is%20off%20by%20default.. Acesso em: 7 jun. 2023.

Haq, Ejaz Ul *et al.* A fast hybrid computer vision technique for real-time embedded bus passenger flow calculation through camera. **Multimedia Tools And Applications**, [S.L.], v. 79, n. 1-2, p. 1007-1036, 14 out. 2019. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/s11042-019-08167-y>.

Haykin, Simon. *Neural Networks and Learning Machines*. Pearson. 3 ed.. 2008.

Hejin, Yuan. A bus passenger flow estimation method based on feature point's trajectory clustering. **2010 IEEE International Conference On Intelligent Computing And Intelligent Systems**, [S.L.], p. 426-430, out. 2010. IEEE. <http://dx.doi.org/10.1109/iccisys.2010.5658589>.

Hsu, Ya-Wen; Chen, Yen-Wei; Perng, Jau-Woei. Estimation of the Number of Passengers in a Bus Using Deep Learning. **Sensors**, [S.L.], v. 20, n. 8, p. 2178, 12 abr. 2020. MDPI AG. <http://dx.doi.org/10.3390/s20082178>.

Hsu, Ya-Wen; Wang, Ting-Yen; Perng, Jau-Woei. Passenger flow counting in buses based on deep learning using surveillance video. **Optik**, [S.L.], v. 202, n. 163675, p. 1-13, fev. 2020. Elsevier BV. <http://dx.doi.org/10.1016/j.ijleo.2019.163675>.

Huang, Wenhao *et al.* A Bus Crowdedness Sensing System Using Deep-Learning Based Object Detection. **Ieice Transactions On Information And Systems**, [S.L.], v. 105, n. 10, p. 1712-1720, 1 out. 2022. Institute of Electronics, Information and Communications Engineers (IEICE). <http://dx.doi.org/10.1587/transinf.2022pcp0007>.

Lengvenis, Paulius *et al.* Application Of Computer Vision Systems For Passenger Counting In Public Transport. **Electronics And Electrical Engineering**, [S.L.], v. 19, n. 3, p. 69-72, 12 mar. 2013. Kaunas University of Technology (KTU). <http://dx.doi.org/10.5755/j01.eee.19.3.1232>.

Lerman, Laura Visintainer *et al.* Comparative analysis between transportation modes for sustainability perspective in one metropolitan region of southern Brazil. **Production**, [S.L.], v. 30, p. 1, 2020. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/0103-6513.20190038>.

Li, Zewen *et al.* A Survey of Convolutional Neural Networks: analysis, applications, and prospects. **IEEE Transactions On Neural Networks And Learning Systems**, [S.L.], v. 33, n. 12, p. 6999-7019, dez. 2022. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tnnls.2021.3084827>.

Mazzeo, Pier Luigi *et al.* MH-MetroNet—A Multi-Head CNN for Passenger-Crowd Attendance Estimation. **Journal Of Imaging**, [S.L.], v. 6, n. 7, p. 62, 2 jul. 2020. MDPI AG. <http://dx.doi.org/10.3390/jimaging6070062>.

Mccarthy, Chris et al. A Field Study of Internet of Things-Based Solutions for Automatic Passenger Counting. **IEEE Open Journal Of Intelligent Transportation Systems**, [S.L.], v. 2, p. 384-401, 2021. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/ojits.2021.3111052>.

Metrô de São Paulo. **Indicadores 2021**. 2022. Disponível em: <https://transparencia.metrosp.com.br/dataset/demanda/resource/f5043ccf-5e00-4aa9-a6be-b1c64b02759b>. Acesso em: 7 jun. 2023.

Metrô De São Paulo. **Indicadores 2022**. Disponível em: https://transparencia.metrosp.com.br/sites/default/files/Indicadores%20-%202022_8.pdf. Acesso em: 7 jun. 2023.

Ming, Yu; Wei, Liu. A New Method for Passenger Flow Counting System Based on Surveillance Video. **2009 Second International Conference On Intelligent Networks And Intelligent Systems, Tianjian, Cn**, p. 453-456, nov. 2009. IEEE. <http://dx.doi.org/10.1109/icinis.2009.122>.

Mukherjee, Satarupa *et al.* A novel framework for automatic passenger counting. **2011 18Th IEEE International Conference On Image Processing**, [S.L.], p. 2969-2972, set. 2011. IEEE. <http://dx.doi.org/10.1109/icip.2011.6116284>.

Nakashima, Hayato; Arai, Ismail; Fujikawa, Kazutoshi. Passenger Counter Based on Random Forest Regressor Using Drive Recorder and Sensors in Buses. **2019 IEEE International Conference On Pervasive Computing And Communications Workshops (Percom Workshops)**, [S.L.], p. 561-566, mar. 2019. IEEE. <http://dx.doi.org/10.1109/percomw.2019.8730761>.

Nakashima, Hayato; Arai, Ismail; Fujikawa, Kazutoshi. Proposal of a Method for Estimating the Number of Passengers with Using Drive Recorder and Sensors Equipped in Buses. **2018 IEEE International Conference On Big Data (Big Data)**, [s. l.], p. 5396-5398, dez. 2018. Seattle, US. IEEE. <http://dx.doi.org/10.1109/bigdata.2018.8621983>.

OpenCV Team. OpenCV. 2023. Disponível em <https://opencv.org/>. Acesso em 14/06/2023.

Rendon, William David Moreno et al. Passenger Counting in Mass Public Transport Systems using Computer Vision and Deep Learning. **IEEE Latin America Transactions**, [S.L.], v. 21, n. 4, p. 537-545, abr. 2023. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tla.2023.10128885>.

Sipetas, Charalampos; Keklikoglou, Andronikos; Gonzales, Eric J.. Estimation of left behind subway passengers through archived data and video image processing. **Transportation Research Part C: Emerging Technologies**, [S.L.], v. 118, p. 102727, set. 2020. Elsevier BV. <http://dx.doi.org/10.1016/j.trc.2020.102727>.

Sutopo, Ricky et al. Appearance-based passenger counting in cluttered scenes with lateral movement compensation. **Neural Computing And Applications**, [S.L.], v. 33, n. 16, p. 9891-9912, 6 fev. 2021. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/s00521-021-05760-x>.

Yang, Biao et al. Counting crowds using a scale-distribution-aware network and adaptive human-shaped kernel. **Neurocomputing**, [S.L.], v. 390, p. 207-216, maio 2020. Elsevier BV. <http://dx.doi.org/10.1016/j.neucom.2019.02.071>.

Yang, Biao et al. Edge computing-based real-time passenger counting using a compact convolutional neural network. **Neural Computing And Applications**, [S.L.], v. 32, n. 9, p. 4919-4931, 29 nov. 2018. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/s00521-018-3894-2>.

Yan-Yan, Chen *et al.* Pedestrian Detection and Tracking for Counting Applications in Metro Station. **Discrete Dynamics In Nature And Society**, [S.L.], v. 2014, p. 1-11, 2014. Hindawi Limited. <http://dx.doi.org/10.1155/2014/712041>.

Yu, Haibin; HE, Zhiwei; LIU, Jilin. A vision-based method to estimate passenger flow in bus. **2007 International Symposium On Intelligent Signal Processing And Communication Systems**, [S.L.], p. 6999-7019, 2007. IEEE. <http://dx.doi.org/10.1109/ispacs.2007.4445972>.

Zhang, Jun; Liu, Jiase; Wang, Zhizhong. Convolutional Neural Network for Crowd Counting on Metro Platforms. **Symmetry**, [S.L.], v. 13, n. 4, p. 703, 17 abr. 2021. MDPI AG. <http://dx.doi.org/10.3390/sym130407>.

Zhang, Jun; Zhu, Gaoyi; Wang, Zhizhong. Multi-Column Atrous Convolutional Neural Network for Counting Metro Passengers. **Symmetry**, [S.L.], v. 12, n. 4, p. 682, 24 abr. 2020. MDPI AG. <http://dx.doi.org/10.3390/sym12040682>.

Zhu, Gaoyi et al. Metro passengers counting and density estimation via dilated-transposed fully convolutional neural network. **Knowledge And Information Systems**, [S.L.], v. 63, n. 6, p. 1557-1575, 18 abr. 2021. Springer Science and Business Media LLC. <http://dx.doi.org/10.1007/s10115-021-01563-7>.