UNIVERSIDADE FEDERAL DO RIO GRANDE - FURG
CENTRO DE CIÊNCIAS COMPUTACIONAIS
CURSO DE ENGENHARIA DE AUTOMAÇÃO

Projeto de Graduação em Engenharia de Automação

Método baseado em autoencoder para geolocalização de veículos aéreos utilizando imagens de satélite

Lucas Benedetti Viana de Cordova

Projeto de Graduação apresentado ao Curso de Engenharia de Automação da Universidade Federal do Rio Grande - FURG, como requisito parcial para a obtenção do grau de Engenheiro de Automação

Orientador: Prof. Dr. Rodrigo da Silva Guerra

Rio Grande, 2024

Dados de catalogação na fonte:
colocar NOME DO BIBLIOTECÁRIO – CRB-colocar número do crb do bibliotecário
Biblioteca Central – FURG

A999a

Cordova, Lucas Benedetti Viana de

Método baseado em autoencoder para geolocalização de veículos aéreos utilizando imagens de satélite / Lucas Benedetti Viana de Cordova. – Rio Grande, 2024. – 50 f: gráf. – Projeto de Graduação – Engenharia de Automação. Universidade Federal do Rio Grande - FURG. Centro de Ciências Computacionais. Rio Grande, 2024. – Orientador Rodrigo da Silva Guerra.

I. Guerra, Rodrigo da Silva. II. Título.

CDD: 999.9

Projeto de Graduação em Engenharia de Automação

Método baseado em autoencoder para geolocalização de veículos aéreos utilizando imagens de satélite

Lucas Benedetti Viana de Cordova

Banca examinadora:

Prof. Dr. Paulo Lilles Jorge Drews Junior

Dr. Matheus Machado dos Santos

Dedico este trabalho à minha mãe Rozelaine, meu pai Marco Antônio, minhas avós Marne e Doralina e à minha companheira Gisele.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, Rozelaine e Marco Antônio, por nunca deixarem faltar amor, carinho e por todo o suporte ao longo da minha jornada acadêmica. À minha companheira Gisele, por ajudar na formatação deste trabalho e por ser o meu porto seguro em meio aos meus anseios. Ao meu orientador, prof. Dr. Rodrigo Guerra, por sua paciência, disponibilidade e seu olhar humano em cada conversa. Gratidão aos amigos Dave, Luciana, Pedro Moreira e Valter por todo o companheirismo e risadas. Aos amigos Alberto, Bruna, Diogo, Letieri e Pedro Lara por todas as palavras de incentivo. E aos colegas Amanda, Gustavo e Lucas pelo seu tempo e suporte técnico.

Sem vocês esse trabalho não seria possível.

”Com organização e tempo, acha-se o segredo de fazer tudo e bem feito.”
— PITÁGORAS

RESUMO

CORDOVA, Lucas Benedetti Viana de. **Método baseado em autoencoder para geolocalização de veículos aéreos utilizando imagens de satélite**. 2024. 50 f. Projeto de Graduação – Engenharia de Automação. Universidade Federal do Rio Grande - FURG, Rio Grande.

Este projeto de graduação investiga a aplicabilidade de um método de geolocalização de veículos aéreos utilizando visão computacional. O trabalho toma como ponto de partida um método originalmente proposto no contexto de voos curtos e de baixa altitude e avalia sua aplicabilidade para voos de maior altitude e mais longas distâncias. A estratégia apresentada visa corresponder uma imagem do solo, capturada em voo, com um banco de imagens de satélite georreferenciadas, para a geolocalização de uma aeronave em missões aéreas mais longas. O método proposto é baseado em uma rede neural do tipo autoencoder, que codifica as imagens de satélite em representações vetoriais (*embeddings*) antes de realizar a correspondência por meio de correlação-cruzada. O modelo foi avaliado em termos de acurácia e tempo de processamento. Por fim, foi feita uma avaliação desta abordagem simulando sua aplicação na geolocalização de aeronaves em um caminho de 200 km, rumando ao norte, a uma altitude fixa de 6000 m. Para simular a captura de imagem em voo foram utilizadas imagens de satélite capuradas em diferentes épocas daquelas presentes no banco de imanges georeferenciadas. Foi demonstrado que o método implementado é capaz de aprender representações discriminativas de imagens de satélite de médias altitudes, uma vez que durante os experimentos obteve-se acurácia de aproximadamente 90%.

Palavras-chave: Geolocalização por imagem, imagens de satélite, autoencoders, correlação-cruzada.

ABSTRACT

CORDOVA, Lucas Benedetti Viana de. **Autoencoder-Based Method for Geolocation of Aerial Vehicles Using Satellite Images**. 2024. 50 f. Projeto de Graduação – Engenharia de Automação. Universidade Federal do Rio Grande - FURG, Rio Grande.

This undergraduate project investigates the applicability of a computer vision-based method for geolocating aerial vehicles. The study takes as its starting point a method originally proposed in the context of short and low-altitude flights and evaluates its suitability for higher-altitude and longer-distance flights. The presented strategy aims to match an airborne-captured ground image with a database of georeferenced satellite images for the geolocation of an aircraft in longer aerial missions. The proposed method is based on an autoencoder neural network, which encodes satellite images into vector representations (embeddings) before performing matching through cross-correlation. The model was evaluated in terms of accuracy and processing time. Finally, an assessment of this approach was made by simulating its application in the geolocation of aircraft on a 200 km path, heading north, at a fixed altitude of 6000 m. To simulate the in-flight image capture, satellite images captured at different times from those present in the georeferenced image database were used. It was demonstrated that the implemented method is capable of learning discriminative representations of medium-altitude satellite images, achieving an accuracy of approximately 90% during the experiments.

Keywords: Image-based geolocation, satellite images, autoencoders, cross-correlation.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
BRIEF	Binary Robust Independent Elementary Features
CNNs	Convolutional Neural Networks
GEE	Google Earth Engine
GNSS	Sistema Global de Navegação por Satélite
GPS	Global Positioning System
GPU	Graphics processing unit
GLONASS	Global Navigation Satellite Systems
HSV	Hue, Saturation, Value
ID	Identificação
INS	Inertial Navigation System
IRNSS	Indian Regional Navigation Satellite System
ITRF	International Terrestrial Reference Frame
LoRaWAN	Low-power Wide Area Networking
MCL	Monte Carlo Localization
MI	Mutual Information
MODIS	Moderate Resolution Imaging Spectroradiometer
MSE	Mean Square Error
QZSS	Quashi-Zenith Satellite System
RAM	Random Access Memory
RGB	Red, Green and Blue
RPN	Region Proposal Network
SIG	Geographic Information System
SIRGAS	Sistema de Referência Geocêntrico para as Américas
TP	Tempo de Processamento
UAV	Unmanned Aerial Vehicle

UTM	Universal Transverse Mercator
WGS84	World Geodetic System 1984
2D	Two-dimensional
6DOF	Six Degrees Of Freedom

SUMÁRIO

1 INTRODUÇÃO

A navegação de veículos aéreos, sejam aeronaves tripuladas, veículos remotamente controlados ou mesmo autônomos, se beneficia largamente dos sistemas de posicionamento global via satélite como ferramenta essencial para executar suas missões. Ao se tratar de Sistema Global de Navegação por Satélite (GNSS) , existem quatro soluções completamente operacionais para uso ao redor do globo: GPS (Global Positioning System), Glonass (Global Navigational Satellite System), Galileo (União Européia) e BeiDou (China). Criado pelos Estados Unidos inicialmente para uso militar, o GPS é hoje em dia o sistema de navegação por satélite mais utilizado no mundo [21]. O Glonass foi desenvolvido pela antiga URSS, também com viés militar e posteriormente também abriu sua utilização para uso civil [21]. O Galileo, diferentemente do GPS e GLONASS, foi concebido para ser operado por civis para possibilitar à comunidade européia o acesso independente aos dados de posicionamento global [10]. Já a solução Chinesa, assegura que seu sistema terá uma precisão de localização de 10 centímetros, enquanto o GPS possui 30 centímetros [22]. Além desses, há também sistemas regionais, como o QZSS (Quashi-Zenith Satellite System) [37], solução japonesa disponível em regiões da Ásia-Oceania com longitudes próximas ao Japão e o IRNSS (Indian Space Research Organisation) [31], sistema indiano capaz de fornecer posição precisa até 1.500 km dos limites daquele país. Em decorrência disto, qualquer serviço que dependa exclusivamente de uma dessas tecnologias estará sujeito a questões diplomáticas, já que os países detentores do serviço têm autonomia para desativá-la a qualquer momento.

Para além da dependência política, a comunicação entre os satélites e o receptor do veículo aéreo podem sofrer interferências de construções em zonas urbanas, o que também compromete a disponibilidade de sinais de satélites. Ainda, o relevo (como regiões montanhosas) ou condições de tempo em uma determinada região também pode influenciar o sinal entre os satélites e os receptores. Como demonstrado em [36], durante eventos de chuva a exatidão da geolocalização fica comprometida em alguns metros para usuários de aparelhos celulares. O sinal também pode ser alvo de hackers, e sofrer interrupção de transmissão, o que causará mal funcionamento ou danificando diversos serviços que dependem dele para localização [19]. Nessa perspectiva, é relevante inves-

tigar outras formas para dispor de um sistema de geolocalização – que seja independente da navegação por satélite – que permita aos veículos aéreos terem maior resiliência e autonomia durante suas missões e viagens.

Atualmente, já existem algumas formas de garantir redundância aos sistemas que utilizam navegação por satélite. No âmbito de tecnologias de radiofrequência, com a miniaturização de dispositivos eletrônicos que operam com baixo consumo energético, o trabalho [32], propõe uma solução IoT (*Internet of Things*) para tolerância de falhas em um sistema de geolocalização baseado em rede LoRaWAN, que utiliza algoritmos de geolocalização para estimar a nova posição do objeto em caso de falha do GPS [32]. Com essa abordagem os autores conseguiram uma precisão na localização do objeto de 151 metros, e recomendam essa técnica de tolerância a falhas em aplicações onde a posição do objeto não precisa ser tão precisa.

Também é possível encontrar na literatura trabalhos que buscaram inspiração em princípios biológicos de navegação para melhorar sistemas de posicionamento de aeronaves. Em [2], é proposto uma solução baseada em fluxo óptico – que consiste em um método natural de evasão de obstáculos utilizado por pássaros e insetos – em conjunto com sensores inerciais, para fornecer capacidade de navegação ao UAV durante eventos de interrupção de sinal do GPS. Os autores desenvolveram um algoritmo inteligente, baseado em colônias de abelhas, que realiza mais rapidamente as medições de fluxo óptico, e por fim, utiliza filtro de Kalman Estendido para fundir os dados de fluxo óptico e sensores inerciais.

Com o barateamento de câmeras e sistemas embarcados de alta performance com GPU embarcada, bem como a abundância de bancos de imagens de satélite do planeta Terra – como as imagens disponibilizadas pelos satélites Landsat [41], MODIS [30] e Sentinel [11] -, as soluções de visão computacional também começam a se tornar alternativas viáveis de servirem como base para desenvolver um sistema de geolocalização nos dias de hoje. Nessa linha os autores propõe em [28] um método para localização global e rastreamento de UAV baseado em visão computacional a partir de imagens de satélites. Este sistema também foi desenvolvido com o intuito de oferecer redundância à aeronave em caso de falha no GPS. Neste mesmo trabalho, os autores apresentam o estado da arte na localização de veículos aéreos baseados em visão, deixando evidente que existe uma necessidade que esses sistemas sejam robustos e possuam baixo custo computacional.

Dado o contexto mencionado, este estudo busca contribuir no problema de geolocalização por meio de métodos de visão computacional. Tendo ciência da abrangência deste tema, foi delimitado como objeto de estudo o uso de métodos de correspondência entre uma imagem capturada da região sobrevoada por uma aeronave em uma base de imagens obtidas via satélite.

1.1 Estrutura do trabalho

Este trabalho foi dividido em introdução, objetivos, revisão bibliográfica, metodologia, resultados e conclusão. Na introdução, foi apresentado o contexto em que esta pesquisa se inseriu, abordando as soluções de GNSS existentes e os problemas associados a essas tecnologias. Por fim, foi feita uma breve introdução sobre os meios existentes de estimar a geolocalização de aeronaves em momentos de pane em sistemas de referência baseados em satélites.

Nos objetivos, foi descrito o propósito deste trabalho, evidenciando as atividades específicas realizadas nesta pesquisa para alcançar o objetivo.

Em seguida, foi realizada uma revisão bibliográfica sobre o tema. Foi apresentada uma série de trabalhos que compõem o estado da arte na problemática de geolocalização de veículos aéreos utilizando visão computacional. Com base na análise dos trabalhos relacionados, foi definido o escopo desta pesquisa no contexto do tema. Referente à fundamentação teórica, foi feita uma breve apresentação do problema de localização utilizando visão. Em seguida são introduzidos os conceitos de autoencoder e correlação-cruzada, que são fundamentais para o entendimento deste trabalho.

A quarta parte compreendeu a metodologia deste trabalho. Foi definido, justificado e descrito o métodos escolhido para criação de mapa de referência e modelagem de uma rede neural do tipo autoenconder. Também foi apresentado o roteiro de experimentos de validação do modelo e foi feita uma simulação desta abordagem como algoritmo de geolocalização.

Na seção de resultados, foram apresentados os resultados dos procedimentos descritos na metodologia referentes ao treinamento da rede neural, experimentos de validação e simulação. Em experimentos, foi feita uma análise do desempenho do algoritmo em termos de acurácia do modelo e tempo de processamento - que são duas variáveis críticas no desenvolvimento de sistemas de localização baseados em visão. Na simulação, foram apresentados os resultados do algoritmo de geolocalização em um caminho de 200 quilometros em linha reta, com rumo ao norte.

Por fim, na conclusão este trabalho avaliou o método proposto, evidenciando os resultados alcançados e que precisam de melhorias, bem como sugeriu caminhos para trabalhos futuros que possam utilizar esta abordagem.

Após a conclusão, foram inseridos os anexos que em algum momento complementaram o desenvolvimento deste trabalho.

2 OBJETIVOS

2.1 Objetivo Geral

Avaliar a viabilidade de uma abordagem de geolocalização por imagens de satélite de média altitude, utilizando autoencoder e correlação cruzada .

2.2 Objetivos Específicos

- Realizar uma revisão do estado da arte em visão computacional para geolocalização de aeronaves utilizando imagens de satélite;
- Elencar um método viável para aplicação embarcada e em tempo real, que seja robusto às variações de aparências entre imagens de diferentes datas;
- Criar um banco de dados com imagens de satélite georreferenciadas;
- Simular a implementação do método;
- Avaliar e documentar os resultados.

3 REVISÃO BIBLIOGRÁFICA

Este capítula tem como objetivo apresentar uma revisão da bibliografia acerca do problema de correspondência de imagens de satélite para a navegação de veículos aéreos. Dessa forma pretende-se ter um maior entendimento dos desafios relacionados ao tema, identificar lacunas na literatura que tenham potencial de exploração, mapear de maneira geral as ferramentas, banco de dados e equipamentos necessários para implementar esta pesquisa, bem como a disponibilidade destes itens e também, avaliar as metodologias existentes para corresponder imagens na temática de geolocalização de veículos aéreos por meio de visão computacional.

3.1 Trabalhos Relacionados

O trabalho de [1] faz uma revisão bibliográfica sobre os métodos mais utilizados para localização de veículos aéreos usando imagem e categoriza os principais métodos em 3 grupos: (I) algoritmos de navegação para correspondência de imagens por correlação extrema, (II) algoritmos de navegação para correspondência de imagens usando pontos chaves e (III) algoritmos de navegação para correspondência de imagens usando redes neurais. Os autores concluíram que atualmente os algoritmos de correspondência por pontos-chave, também conhecidos por codificadores, extratores de features ou descritores, são os mais amplamente utilizados. E ainda destacam que abordagem que utilizam descritores são capazes de extrair informações importantes e resistentes a distorções das imagens.

Conforme afirmado por [1], uma das formas mais comuns de realizar a busca inteligente por correspondência de imagens é através de descritores. Descritores são algoritmos que extraem informações de uma imagem, como pontos, linhas, arestas, cantos, pixels, cores, histogramas ou entidades geométricas [18], para que posteriormente possam ser usadas para realizar a correspondência de imagens. Com esse intuito, o trabalho dos autores de [28], apresentado anteriormente, visa mitigar este problema utilizando um descritor binário – denominado abBRIEF – para realizar a correspondência entre a imagem capturada por um drone e um mapa global georreferenciado, aliado a um algoritmo de otimização de filtro de partícula. Nesse caso, o uso do descritor abBRIEF [28] resultou

em um baixo tempo de execução e boa taxa de acerto de correspondência entre imagens para os testes realizados. Apesar dos autores conseguirem resultados promissores em seus experimentos, destaca-se a importância de considerar o espaço de busca que foi utilizado durante os experimentos. No trabalho proposto, o experimento com a maior distância foi o vôo número 2, que percorreu um total de 2.400 metros em um mapa de 1,16 km². Levando em consideração que nesse trabalho o descritor percorre o espaço de busca por completo para realizar a correspondência entre a imagem capturada pelo drone e o mapa global, é possível que utilizar esta abordagem demande um custo computacional alto para veículos aéreos que sobrevoam áreas maiores, uma vez que não existe nenhuma etapa de filtragem que elenque regiões candidatas antes de realizar a correspondência entre imagens.

Outro método de extrair informações de uma imagem é com o uso de Redes Neurais Convolucionais (CNNs) – que ganharam visibilidade após o sucesso da rede AlexNet no Imagenet Challenge [35] – que identificam elementos pontuais em uma imagem para classificá-la em determinado grupo. Nessa linha, em [34], apresentam um sistema de detecção de objetos que usa RPN (Region Proposal Network) para realizar a proposta de regiões candidatas antes do processo de detecção. Nessa abordagem eles demonstraram uma diminuição significativa no tempo total no processo do sistema de detecção, bem como uma melhora na precisão geral da detecção de objetos. Dessa forma, conforme apresentado em [34], ao realizar a etapa de filtragem semântica a busca inteligente irá preservar recursos computacionais mais intensos que seriam usados na correlação de imagens numa busca bruta por correspondências como é feito em [28].

Ainda na linha de redes neurais, um trabalho que investigou como obter robustez do sistema de localização em relação a variações de perspectiva e aparência entre as imagens, foi proposto pelos autores em [23]. Esse método baseado em visão utiliza um modelo neural que aprende representações discriminativas a partir de imagens de navegação do veículo em solo, em conjunto com imagens de satélite. O sistema possui como entrada imagens de uma câmera, odômetro e sensor inercial para estimar a pose do veículo em relação a uma imagem de satélite georreferenciada. Ao contrário da proposta de [28], este modelo utiliza uma rede siamesa – que consiste em duas redes neurais convolucionais (CNNs) – para realizar a correspondência entre imagens. Cada uma das redes é utilizada de forma distinta, uma rede é treinada com imagens do solo enquanto a outra é treinada utilizando imagens do satélite (mapa global). Assim, as representações aprendidas são comparadas para detectar regiões candidatas. Similar à solução de [28], após a correspondência, os autores utilizam uma abordagem de filtro de partícula, mas para esse caso com o viés de minimizar o efeito de aliasing perceptivo gerado pela rede.

Em [5], também utilizando imagens de satélite georreferenciadas, os autores propõe uma abordagem para localização de UAV em baixas altitudes baseada na arquitetura de uma CNN denominada autoencoder [24]. Esse método demonstrou-se rápido, robusto à variações de aparência da imagem e com baixo custo computacional em comparação

com outro trabalho que utilizou o mesmo mapa de referencia, mas que usou a abordagem de *Mutual Information* (MI) [33]. Os autores comparam seu método com [33] e apresentaram bons resultados em termos de otimização do tempo de processamento para comparação das imagens, 0.26ms contra 109ms, bem como o custo total de armazenamento, 423 Mb contra 794 Mb, mantendo a taxa de acerto do modelo na correspondência de imagens igual ao método apresentado por [33]. A metodologia dos autores consiste em duas etapas: (1) treinar offline o autoencoder com imagens de satélite com o intuito de que a rede seja capaz de aprender as representações discriminativas. Foi utilizada como função de aprendizagem o Erro Quadrático Médio (RMSE) da perda fotométrica entre as imagens de entrada e saída da rede, em conjunto com o RMSE da perda entre as camadas intermediárias da rede. Após atingirem resultados satisfatórios, utiliza-se a rede treinada para codificar todo o mapa da missão em vetores no espaço latente (*embeddings*) e armazenam-se estes dados no drone;(2) Para realizar os voos, apenas a camada de codificação da arquitetura, encoder, da rede foi embarcada no drone. Durante o voo, o drone captura imagens aéreas em tempo real. Na sequencia, o codificador comprime a imagem capturada. Com base em uma previsão da pose atual, o espaço de busca é reduzido em 4 metros para frente e 4 metros para trás. A busca da imagem codificada é feita comparando a imagem codificado com os embeddings que compreendem esse espaço de busca reduzido por meio do kernel de produto interno para estimar a localização do drone. Entre as melhorias para a proposta, os autores sugerem investigar formas de generalizar a rede para ser capaz de atuar em diferentes tipos de mapas, pois a solução dos autores requer retreinamento da rede para uso em um novo mapa.

Ao contrário dos trabalhos apresentados até aqui, o trabalho de [39] aborda o problema de localização de UAVs em médias altitudes. Os autores utilizaram um UAV modelo Orion-E para realizar um experimento a uma altitude de 3000 metros. O experimento durou 150 s, cobrindo uma região de 7 quilometros. O trabalho utilizou um mapa topográfico vetorial como referencia para comparação com a imagem em tempo real do UAV. A imagem capturada pela aeronave inicialmente é segmentada utilizando uma rede U-Net [6] para destacar estradas, rios e fundo. Posteriormente, foi utilizado o detector SIFT [27] para extração dos pontos-chaves da imagem segmentada e por fim, o método RANSAC [12] foi usado para cálculo da matriz de homografia. O método proposto foi capaz de fornecer a correção da localização do UAV com precisão de aproximadamente 100 metros, em eventos de indisponibilidade de sinal do GNSS (Global Navigation Satellite Systems).

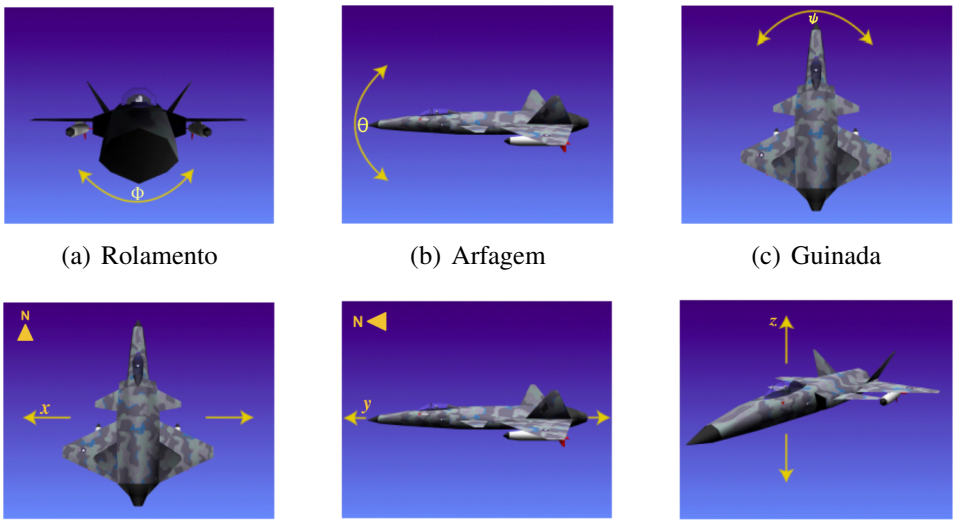
Outro trabalho que explorou os desafios de localização global de um UAV em larga escala de mapa, no entanto, em baixas altitudes foi apresentado por [25]. Os autores propuseram uma abordagem para localizar UAV capaz de lidar com variações e ambiguidades naturais presentes nas cenas das imagens capturadas por um drone em um mapa de 100 km², com a ressalva de que este método não precisa de nenhuma informação prévia

sobre a pose inicial da aeronave. Esta abordagem utiliza uma rede chamada CapsNets [20] como descritor que compacta as informações aprendidas de imagens de satélite do mapa da missão. A escolha desta rede para a aplicação consiste na capacidade da CapsNets em formar representações estáveis e robustas em relação a perturbações de entrada [20]. Durante o experimento o UAV obteve precisão de 12,6 à 18,7 metros em uma mapa que apresentava diferença de sazonalidade entre as imagens obtidas pela UAV e o mapa de referencia.

Dentre os desafios enfrentados pelos trabalhos apresentados, foi verificado uma escassez na literatura a respeito de abordagens localização para veículos aéreos que investiguem voos em médias altitudes, superiores a 2000 metros [7]. Desta forma, direcionou-se esta pesquisa para utilizar abordagens de visão computacional para geolocalização de aeronaves em mapas com área de milhares de quilômetros quadrados e em médias altitudes.

3.2 Fundamentação Teórica

O desafio de geolocalização de veículos aéreos envolve estimar a posição e a atitude de uma aeronave em 6 graus de liberdade (6DOF) [1]. Quando diz-se que um objeto possui 6DOF, refere-se à capacidade desta aeronave se deslocar no espaço tridimensional em três graus de orientação (ϕ,θ,ψ) e três graus de translação (x,y,z). A Figura 1 ilustra os 6DOF de um veículo aéreo. A Figura 1(a) apresenta a capacidade de rotação em torno eixo ϕ. A Figura 1(b) diz respeito a inclinação da aeronave, eixo θ. Já a Figura 1(c) ilustra a direção do movimento de guinada de uma aeronave, eixo ψ. As Figuras 1(d), 1(e) e 1(f) correspondem respectivamente aos deslocamentos translacionais de longitude (eixo x), latitude (eixo y) e altitude (eixo z) da aeronave.



(a) Rolamento (b) Arfagem (c) Guinada

(d) Longitude (e) Latitude (f) Altitude

Figura 1: 6DOF de um veículo aéreo (adaptado de [29]).

Tipicamente os parâmetros de orientação ϕ, θ e ψ, podem ser obtidos por meio de Sistemas de Navegação Inercial (INS). INS são unidades de medição equipadas com acelerômetros, giroscópios e magnetômetros que são responsáveis, respectivamente, por realizar a medição de deslocamento linear, orientação e medida de norte magnético do globo [38]. Em conjunto, estes sensores são capazes de integrar leituras ao longo do tempo, fornecendo informações de orientação, deslocamento e rumo para navegação e controle das aeronaves. Entretanto, a estimação de posição e atitude através da integração de leituras de um INS acumula erros ao longo do tempo. Para obter-se informações globais de translação em x, y e z, são largamente utilizados GNSS, como o GPS. Uma aeronave equipada com INS e GNSS é capaz de medir todas as orientações e translações necessárias para funcionamento pleno. Em momentos de falha do GNSS a geolocalização pode ser fornecida pelas medidas do INS, mas por um curto período de tempo [1].

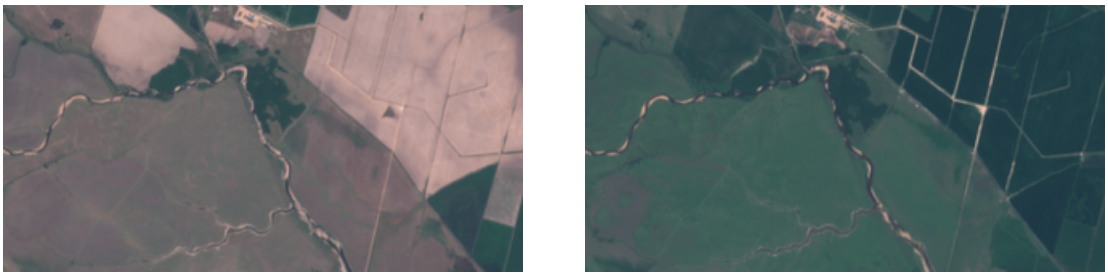
É nesta lacuna, composta pela falha de funcionamento do GNSS e a imprecisão das unidades INS em fornecer a localização da aeronave por longos períodos, que este trabalho visa contribuir para os avanços no tema de geolocalização de veículos aéreos por meio de visão computacional.

Visão computacional é o campo de estudo que busca extrair informações relevantes do mundo por meio de imagens capturadas por uma ou mais câmeras [17]. No âmbito de geolocalização de veículos aéreos utilizando imagens de satélite, informações relevantes que podem auxiliar no reconhecimento de uma região seriam estradas, construções, fazendas, rios, lagos e zonas urbanas. Uma das formas de utilizar visão computacional para fornecer a geolocalização de uma aeronave é com imagens de satélite georreferenciadas. Este é um tema amplo, que compreende questões relacionadas à calibração de câmeras que estabelece a relação entre o mundo real 3D e a informação bidimensional capturada pela câmera [3], odometria visual, responsável por recuperar o movimento da câmera observadora a partir da sequência de imagens obtida [13] - também conhecido como ego-motion -, e a busca de correspondências entre imagens que consiste em encontrar pontos, formas, regiões em comum entre duas imagens [4].

Correspondência de imagens

A geolocalização de UAVs por meio de visão computacional é baseada em algoritmos de correspondência entre imagens. Neste cenário dada uma imagem x de interesse, a correspondência de imagem diz respeito à busca por x em um conjunto de imagens de referência. No entanto, como destacado em [34], o problema de localização global de robôs utilizando visão computacional é desafiador devido às variações de aparência resultantes de mudanças de perspectiva, conteúdo da cena (construções, estradas ou veículos), presença de nuvens (quando em aplicações de médias e altas altitudes), iluminação (como o horário do dia e estações do ano) entre as imagens capturadas pela câmera e as imagens

do banco de dados. A Figura 2, ilustra variação de aparência entre duas imagens de satélite de um mesmo local com diferença de um ano.



(a) Imagem de janeiro de 2022. (b) Imagem de janeiro de 2023.

Figura 2: Diferença de aparência entre duas imagens de satélite de datas distintas [15].

Uma vez que as regiões de interesse podem apresentar variações de aparência e desfoque de movimento [26], entre outras condições que se alteram entre a captura original da imagem presente na base de dados e a captura posterior pela aeronave, é crucial mitigar meios de corresponder pares de imagens de forma robusta. Nos trabalhos [5], [23] e [25], os autores tiveram boas taxas de sucesso na localização global de UAVs utilizando algoritmos de codificação de imagens em representações vetoriais (*embeddings*), para comprimir imagens de satélite em uma forma compacta, mantendo as representações discriminativas das imagens.

Uma maneira de representar imagens de forma compacta é por meio de *autoencoders*. Um *autoencoder* é uma CNN projetada para aprender vetores densos que codifiquem de forma eficiente as representações discriminativas de uma imagem, a ponto de, dada uma imagem de entrada, conseguir reconstruí-la em sua saída [24]. Conforme ilustrado pela Figura 3, a arquitetura de um *autoencoder* é dividia em três partes principais.

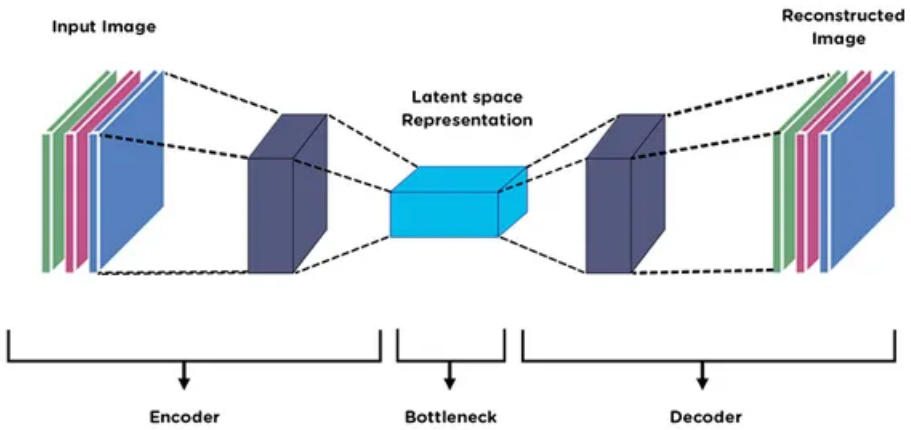


Figura 3: Arquitetura de um autoencoder [8].

A primeira, denominada *encoder*, recebe a imagem de entrada (*Input Image*) e atua como um gargalo, comprimindo a imagem de maneira a forçar que as informações dis-

criminativas da imagem sejam mantidas em uma espaço muito pequeno, denominado espaço latente. Esta representação no espaço latente, etapa denominada *bottleneck* na Figura 3, deve conter as informações necessárias para representar a imagem de entrada. A terceira parte da rede, chamada *decoder*, realiza exatamente o processo inverso do *encoder*. Nessa etapa, a rede opera para reconstruir a imagem (*Reconstructed Image*) de entrada a partir da informações contidas na representação vetorial no espaço latente. Para fins de simplificação, daqui em diante será utilizado o termo *embedding* para referir-se a representação da imagem neste espaço latente.

Em [5], o método utilizado para realizar a correspondência entre as imagens codificadas é o cálculo de produto interno, também chamada de correlação cruzada. Em análise de séries temporais a correlação cruzada é um método que mede o grau de similaridade entre dois conjuntos de números [9]. Esta abordagem, fundamenta-se na premissa de que ao multiplicar-se ponto a ponto duas séries temporais, a soma dos produtos será uma quantificação da sua relação [9]. A Eq. 1 representa a função da correlação cruzada.

$$r_{x,y} = \sum_{i=0}^{N-1} x_i y_i \tag{1}$$

Onde N é o número de dados em dada série temporal, x_i é o i-ésimo elemento da primeira série de dados, y_i é o i-ésimo elemento da segunda série de dados, e r, é a correlação cruzada. Para fins de compreensão a Eq. 1 pode ser comparada, em notação matricial, ao produto escalar de dois vetores, x e y.

É conveniente antes de realizar a correlação cruzada normalizar os dados. Isto pode ser feito pela Eq. 2

$$z_i = \frac{x_i - \mu}{\sigma} \tag{2}$$

sendo μ descrito pela Eq. 3

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{3}$$

e σ pela Eq. 4

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2} \tag{4}$$

onde x_i é o i-ésimo elemento de um conjunto de dados a ser normalizado, μ é a média de valores do conjunto de dados, σ é o desvio-padrão associado ao conjunto de dados e z_i compreende a i-ésima amostra normalizada de um conjunto de dados.

No contexto de localização, a correspondência de imagens pode tanto servir como uma estimativa auxiliar no objetivo de encontrar a pose de uma aeronave, como realizado pelos autores em [25], como também pode atuar como o próprio sistema de navegação do

veículo aéreo, conforme desenvolvido em [5].

4 METODOLOGIA

Este capítulo detalha o desenvolvimento de um método para geolocalização global de aeronaves em média altitudes, acima de 2000 metros, com o uso de técnicas de visão computacional e imagens de satélite georreferenciadas.

Este trabalho visa corresponder corretamente a imagens de satélite de uma mesma região mas com um ano de diferença. O intuito dessa metodologia é avaliar a robustez do sistema à variações de aparências entre essas imagens ao longo do tempo. Nesse cenário, foi utilizado como mapa de referência imagens de satélite de janeiro de 2022, enquanto para o mapa de voo (que simula uma região sobrevoada por uma aeronave) contém imagens de satélite de janeiro de 2023. O método consiste em utilizar uma rede neural codificadora em conjunto com um algoritmo de correlação cruzada. Não é o foco deste trabalho estimar a rolagem, arfagem e altitude da aeronave, uma vez que estes parâmetros podem ser estimados com equipamentos que independem de um GNSS, como mencionado anteriormente no capítulo 3.2. Para fins de simplificação para este experimento e simulação, foi assumido que a aeronave manteve altitude constante de 6000 metros. Por fim, espera-se que a abordagem proposta seja capaz de corresponder imagens de satélite que possuem variação de aparência, em mapas de larga escala, com um tempo de processamento próximo ao estado da arte (ver Tabela 2).

A primeira etapa consiste na criação dos mapas. Foi escolhida a plataforma Google Earth Engine (GEE) [15] para criação do mapa de referência e de voo. Optou-se por essa plataforma por possuir uma ampla gama de imagens dos satélites MODIS, Landsat e Sentinel. Por exemplo, o satélite Sentinel-2 Nível C1 (escolhido para este trabalho) revisita a mesma região do globo a cada 5 dias desde 25 de junho de 2015 com resolução de imagem máxima de 10 m/pixel. Neste trabalho, foi fixada a resolução de 10 m/pixel na escala de altitude desejada. Outra característica que destaca o GEE é a computação em nuvem, o que permite paralelizar tarefas de exportação de imagens e criação de mapas georreferenciados. Toda a criação de mapas foi feita utilizando a API-Client para Python [16]. Dessa forma é possível automatizar processos para gerar mapas de diversos períodos do ano, captando assim variações de relevo referente plantações, construções e estradas, bem como um massivo conjunto de dados. Este último item que é um requisito no contexto de

treinamento de redes neurais.

Na sequência, inspirado por [5], este trabalho utilizou as imagens do mapa de referência para treinar um autoencoder a aprender a codificar as informações representativas de imagens de satélite na forma de *embeddings*. Salienta-se que todas as imagens estão orientadas para o norte. Posteriormente, foi utilizada a abordagem de correlação cruzada para corresponder imagens codificadas. Essa abordagem comparou imagens de forma mais rápida do que métodos baseado em Informação Mútua (MI) [33], que segundo os autores de [5], é uma das abordagens atuais com maior taxa de sucesso na correspondência entre imagens de satélite. O trabalho de [5] manteve taxa de sucesso similar à do trabalho [33], porém corresponde um par de imagens codificadas em 0.26 ms, enquanto o método MI corresponde duas imagens em 109 ms. Essa redução no tempo e processamento é evidente, uma vez que a busca por correspondência é feita sobre a imagem bruta - 176.400 pixels por imagem [33] -, enquanto o método utilizando autoencoder usa correlação cruzada para comparar *embeddings* de dimensão 1x1000. Em [28], os autores evidenciam que o desenvolvimento de sistemas de localização baseados em visão, além de precisarem ser robustos às variações de aparência entre as imagens correspondidas, é necessário que estes possuam baixo custo computacional e tempo de execução para realizar a correspondência entre imagens e disponibilizá-la em tempo real. Nesse contexto, sustenta-se a escolha deste método que representa as imagens de forma compacta antes de realizar a busca entre imagens brutas.

Por fim, foram conduzidos experimentos para validar a acurácia do método desenvolvido para mapas com 10 tamanhos de áreas diferentes (ver Tabela 1). O tamanho de mapa que apresentou melhor resultado foi escolhido como mapa global para simulação do método como algoritmo de geolocalização em um caminho.

Criação do mapa de referência e de voo

Para criação do mapa de referência, foi utilizada a API em Python do GEE no ambiente de desenvolvimento do Google Colab [14]. O Colab contendo os códigos para criação dos mapas georreferenciados está disponível publicamente [1]. Foi escolhida arbitrariamente uma área do Rio Grande do Sul para construção dos mapas. Inicialmente, foram inseridas as coordenadas de latitude e longitude, em graus, para criar as geometrias de referência para o mapa global. Abaixo a Figura 4 apresenta as quatro geometrias utilizadas para criação do mapa de referência.

Em seguida, define-se a dimensão de uma única imagem que irá compor este mapa. Para este caso a dimensão de 160x320 pixels foi escolhida, seguindo as mesmas dimensões de imagem usadas pelos autores de [5]. Para criação das geometrias foi utilizado

[1] https://colab.research.google.com/drive/1AuB5yIOxcDcV6HIgSPOrfXcDknYrE2Vu?usp=sharing.

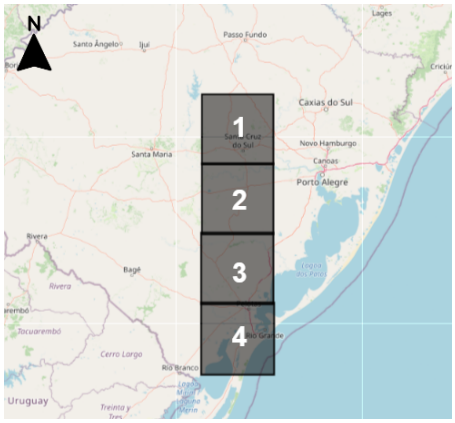


Figura 4: Geometrias que compõe o mapa global.

o sistema de referência SIRGAS 2000, com zona UTM22S [2], que corresponde à projeção para a área de interesse. O sistema de referência SIRGAS 2000 tem como objetivo conectar sistemas de altitude da América do Sul ao ITRF e utiliza um total de 184 estações distribuídas na América do Norte, Central e do Sul [3]. Para fins de exemplo, a Figura 5 abaixo apresenta a geometria 1 e suas respectivas sub-geometrias. As demais geometrias passaram pelo mesmo procedimento.

[2]Em sistemas baseados em UTM, a letra 'S' maiúscula acrescida ao número da zona diz respeito a regiões abaixo da linha do equador.

[3]O mapa do SIRGAS 2000 utilizado está disponível no Anexo A.

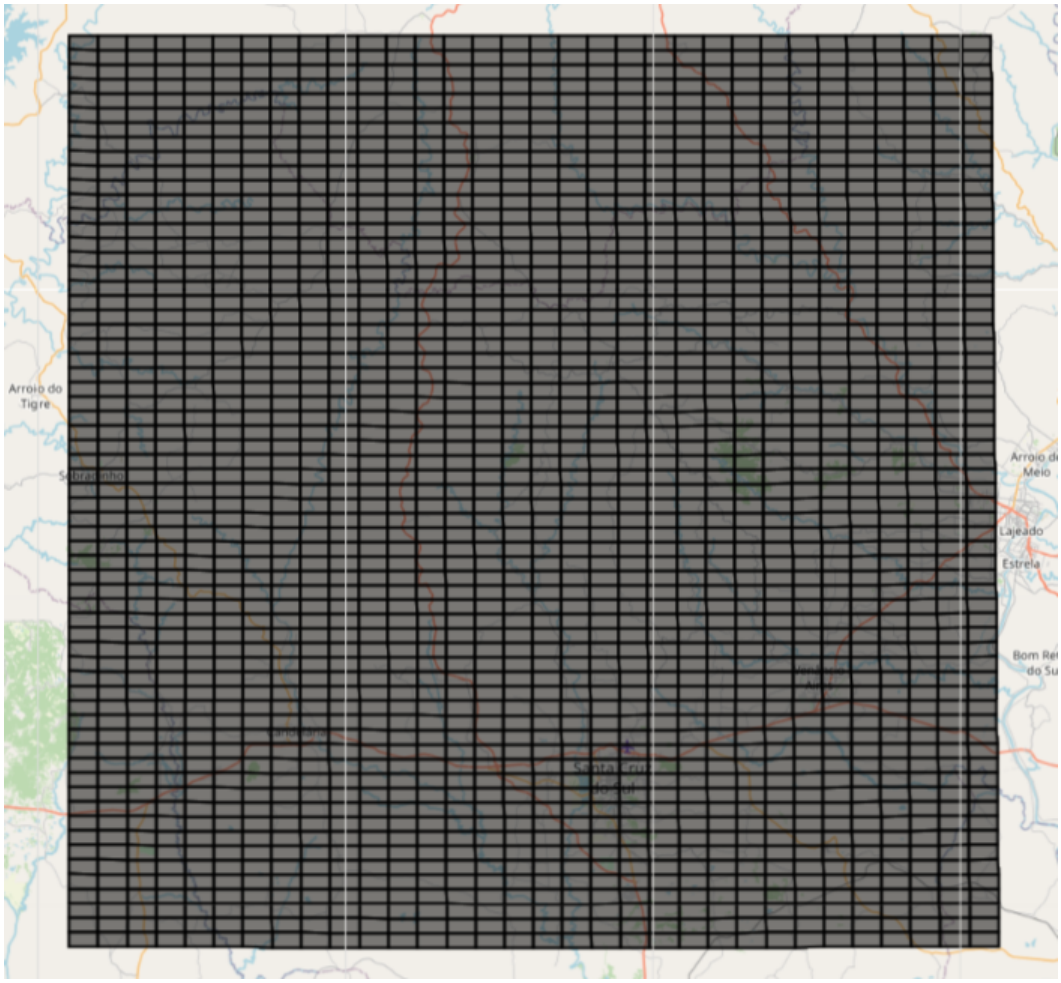


Figura 5: Sub-geometrias da geometria 1.

Após criadas, as sub-geometrias são percorridas iterativamente a uma mesma escala de exibição para simular a altitude de 6000 m (procedimento demonstrado no Anexo B) e para cada ciclo calcula-se o centroide da geometria, seleciona-se o satélite Sentinel-2, especifica-se a data da imagem desejada e se atribui um ID à imagem. Este valor de centroide é o que será usado para geolocalizar o veículo em latitude e longitude após a busca pela correspondência. Por fim, para cada exportação, foi recortada a imagem de satélite bruta na área de cada uma das sub-geometrias. A Figura 6 contém um diagrama que ilustra esta etapa.

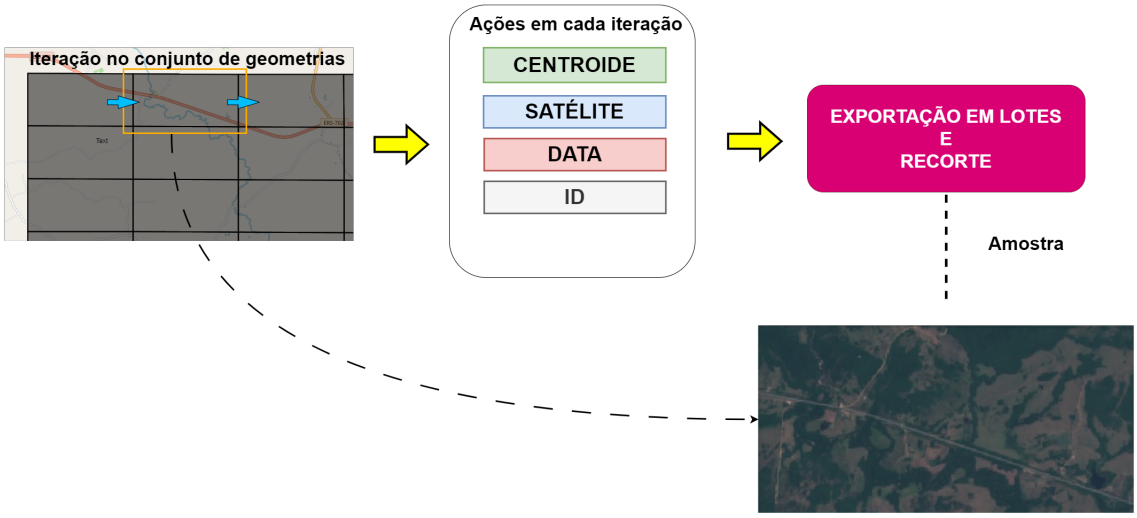


Figura 6: Processo de exportação das imagens georreferenciadas. Fonte: autor.

A exportação das imagens foi realizada em lotes no formato .tiff. Para este caso, o mapa construído contém 252 geometrias de norte a sul e 32 geometrias de leste a oeste, gerando um total de 8064 imagens que serão divididas em dados de treinamento e teste. Sendo a área de um pixel equivalente a 100 m², a área total do mapa exportado é de aproximadamente 41.287,68 km². As informações de coordenadas de latitude e longitude foram exportadas em um arquivo .csv. Para cada imagem foi criado um ID que corresponde ao mesmo ID de cada coordenada exportada, mantendo-se assim a correspondência correta entre elas. O mesmo procedimento foi realizado para exportação do mapa de voo, com exceção que o mapa de referência é datado entre janeiro e fevereiro de 2022 e o mapa de voo entre janeiro e fevereiro de 2023. Portanto as imagens de voo e de satélite estão alinhadas. Na prática o alinhamento das imagens capturadas em voo poderiam ser realizados utilizando a orientação do INSS. A Figura 2 apresenta amostras do dois conjuntos de dados, evidenciando as diferenças visuais entre eles.

Modelagem e treinamento do autoencoder

A arquitetura do autoencoder baseia-se em [5], exceto pelo fato de que neste trabalho a imagem de entrada é em RGB, enquanto no referido trabalho os autores utilizaram imagens em escala de cinza. A Figura 7 apresenta a arquitetura da rede neural, bem como evidencia as medidas utilizadas na função de perda.

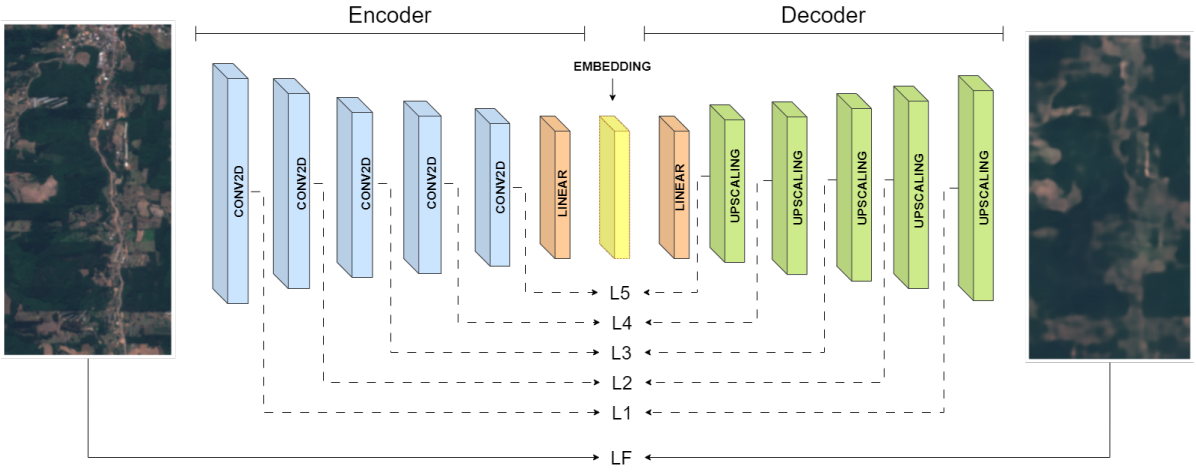


Figura 7: Arquitetura do autoencoder. LF corresponde a perda fotométrica entre a imagem de entrada e a imagem reconstruída. L1, L2, L3, L4 e L5 são as perdas entre as camadas intermediárias. Fonte: autor.

Na etapa de *encoder*, os componentes responsáveis por comprimir a imagem de entrada são cinco camadas de convolução 2D. A Figura 8 ilustra a redução de dimensionalidade que as convoluções efetuam na imagem de entrada, comprimindo-a a cada camada. Após as operações de convolução, a camada de linearização (linha pontilhada) transforma a saída do *encoder* em um *embedding* de 1000 elementos.

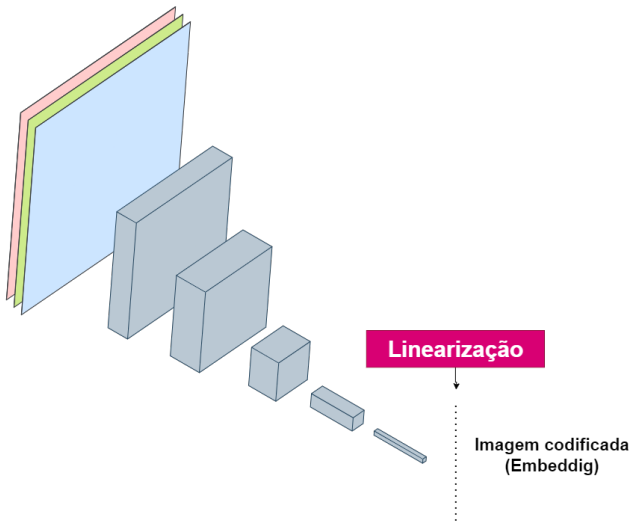


Figura 8: *Encoder* codificando uma imagem em *embedding*. Fonte: autor.

O *decoder* atua de forma oposta ao *encoder*. A reconstrução do *embedding* na imagem de saída é feita por uma camada linear e cinco camadas de *Upscaling*. A Figura 22 do Anexo C apresenta uma descrição completa das camada da rede, evidenciando os tipos de camadas utilizadas, formato de saída e número de parâmetros.

A função de perda utilizada foi Erro Médio Quadrático (MSE) da perda fotométrica entre a imagem de entrada e a imagem reconstruída. Em conjunto com a perda fotométrica foi utilizada a medida de MSE entre as camadas intermediárias correspondentes do *encoder* e *decoder* com um valor de ponderação α igual à 0,01. Em [5], foi avaliado que utilizar o erro das camadas intermediárias incentiva o *decoder* a aprender o caminho reverso do *encoder*, melhorando o desempenho de aprendizagem da rede. A Eq. 5 descreve a função de perda utilizada.

$$Loss = LF + \alpha(L1 + L2 + L3 + L4 + L5) \tag{5}$$

Para treinamento do autoencoder foi utilizado um computador com 16 GB RAM, equipado com placa de vídeo NVIDIA GeForce GTX Titan X de 12 GB. A rede foi treinada com imagens de satélite do mapa de referência por 200 épocas. O conjunto de 8064 imagens foi divido 85% em dados de treinamento e 15% em dados de teste, com taxa de aprendizagem de $1e^{-4}$. O código foi implementado utilizando o *framework pyTorch*.

Experimentos de validação e simulação

A validação do modelo foi feita por meio de experimentos com imagens de satélite do mapa de voo. No mapa de referência constam imagens datadas entre janeiro e fevereiro de 2022, e no mapa de voo, entre janeiro e fevereiro de 2023. Os experimentos foram realizados em 10 tamanhos de mapas diferentes, conforme ilustrado pela Tabela 1. Para cada tamanho de mapa foram realizados 10 experimentos com 50 amostras cada. Vale ressaltar que em voo, as camadas de *decoder* não são necessárias, tendo sua única função o treinamento do modelo.

Mapa	Redução (%)	Área do mapa (km²)	Nº de imagens
1	89,68	4.259,84	832
2	79,76	8.355,84	1632
3	69,84	12.451,84	2432
4	59,92	16.547,84	3232
5	50,00	20.643,84	4032
6	39,68	24.903,68	4864
7	29,76	28.999,68	5664
8	19,84	33.095,68	6464
9	9,92	37.191,68	7264
10	Original	41.287,68	8064

Tabela 1: Áreas dos mapas utilizados nos experimentos.

A Figura 9 apresenta o fluxograma do experimento. Essa abordagem utiliza uma etapa de preparação, 'offline', em que codifica cada imagem do mapa de referência em *embedding* e normaliza os dados usando a equação 2. A simulação inicia sorteando uma imagem do mapa de voo. Essa imagem é codifica em *embedding* e normalizada usando a Eq. 2. Posteriormente é feita a busca pela correspondência do *embedding* da imagem de voo, no mapa de referencia codificado utilizando correlação cruzada. Após a etapa de correlação, por meio de ranqueamento elenca-se o resultado que apresentou maior similaridade. Por fim, compara-se o ID da imagem de voo com o ID da imagem escolhida para verificar a correspondência.

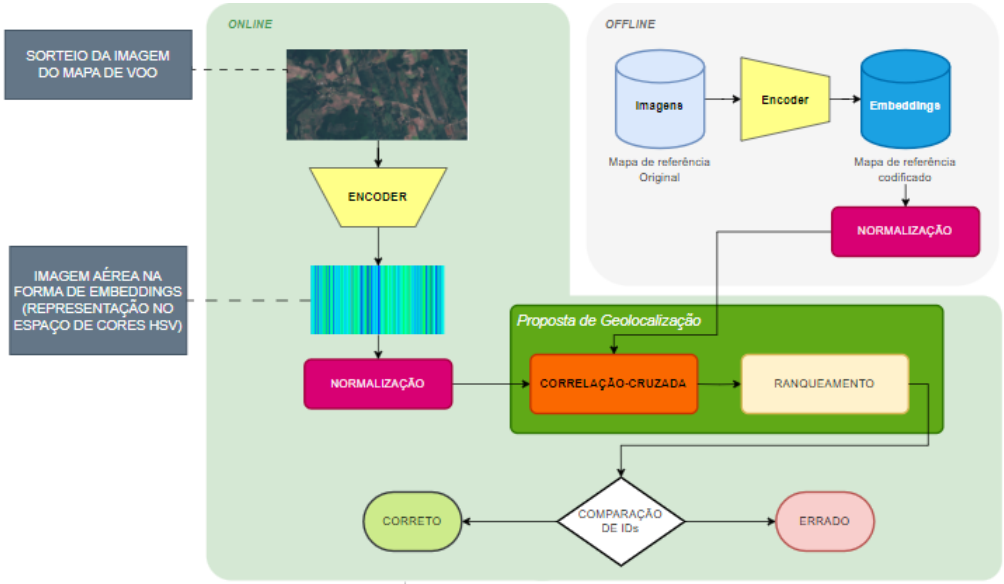


Figura 9: Fluxograma dos experimentos para validação do modelo de correspondência de imagens.

A simulação é conduzida de forma similar ao fluxograma da Figura 9. Com a ressalva de que para a simulação as imagens não são sorteadas, e sim inseridas de forma ordenada para formar o caminho em linha reta com rumo para o norte.

5 RESULTADOS

Neste capítulo são apresentados os resultados de treinamento do modelo referente à função de perda e ao tempo de treinamento, bem como a acurácia do método nos experimentos de validação. Nos experimentos é feita uma análise com foco no consumo de memória e tempo de processamento. Por fim, é demonstrada uma simulação utilizando esta abordagem como algoritmo de geolocalização em um caminho no sentido sul-norte. Neste tópico, são apresentados os resultados do modelo em termos de acurácia e robustez às variações de aparência.

A Figura 10 apresenta o gráfico do erro relacionado à função de perda, MSE, ao longo das épocas. Em azul, é apresentado o erro do modelo em dados de treinamento e em verde o erro em dados de teste. O autoencoder foi treinado até que não se observasse mais melhora significativa nos dados de teste. Empiricamente, foi assumido que a curva da função de perda referente aos dados de teste saturou em aproximadamente 200 épocas. As 8064 imagens do mapa de referência foram divididas 85% em dados de treinamento e 15% em dados de teste.

A unidade do erro é referente ao MSE associado da perda fotométrica entre a imagem de entrada e a imagem reconstruída, adicionado a soma do MSE da camadas intermediárias correspondentes do encoder e decoder, multiplicadas por um fator $\alpha = 0.01$ de ponderação. O tamanho do kernel, stride e padding foram respectivamente 4, 2, e 1, para codificar a imagem, enquanto para decodificar foram utilizados os tamanhos 3,1 e 0, respectivamente. Em um computador equipado com CPU de 16 GB de memória RAM e GPU NVIDIA GeForce GTX Titan X de 12 GB, o tempo de treinamento do modelo foi de aproximadamente 11 horas.

Na sequência os resultados dos experimentos são apresentados. O modelo foi testado em 10 tamanhos de mapas diferentes conforme Tabela 1, seguindo a configuração de experimento do fluxograma da Figura 9. Os resultados alcançados para cada tamanho de mapa são exibidos no gráfico da Figura 11. Do mapa 1 ao 10 é acrescido 10% ao tamanho do mapa, sendo o mapa 1 com 10% do tamanho total, até o mapa 10 referente ao mapa completo. Em preto é mostrado o erro do modelo para ±2 desvios-padrão. Ao

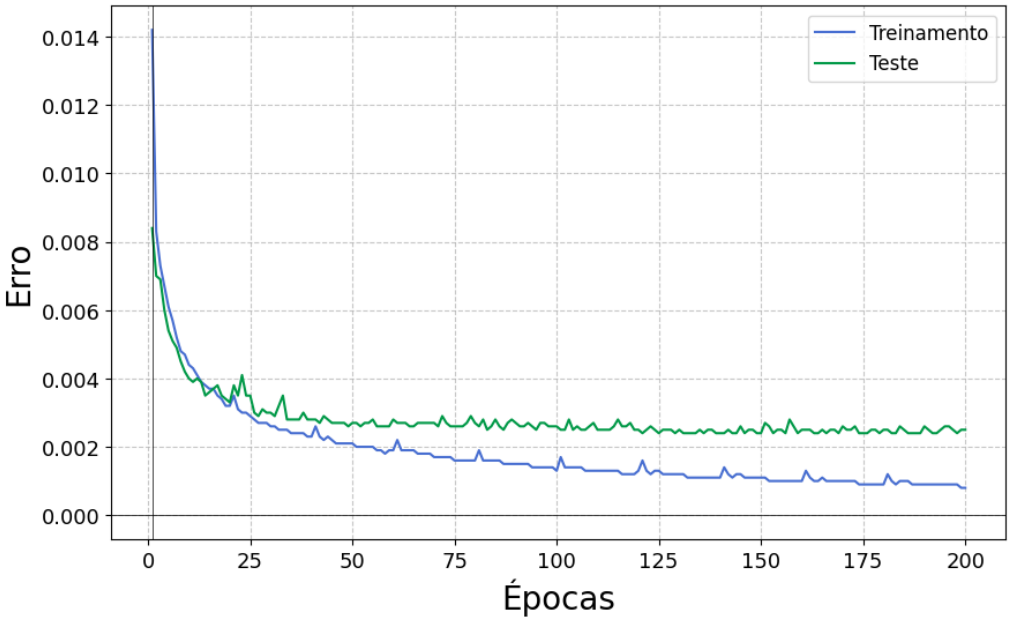


Figura 10: Curva do erro nos conjuntos de treinamento e teste.

analisar a linha de tendência (representada em vermelho tracejado), revela-se uma relação inversamente proporcional entre o tamanho do mapa de busca e a acurácia do método proposto.

Para escolha do mapa de simulação, foi considerado o caso que apresentou acurácia próxima de 90%. Como o objetivo deste trabalho, que é explorar mapas em largas escalas de altitude e área, foi optado por escolher o maior mapa que mantivesse resultados próximos de 90% de acurácia. As melhores opções para simulação foram os mapas 5 e 7, que possuem erro máximo estimado de 89,26% e 89,52%, respectivamente.

Conforme apresentado no capítulo 4, o custo computacional é um requisito importante no projeto de sistemas de geolocalização baseados em visão, visando aplicações em tempo real. Por esse fator, foi feita uma análise do tempo de processamento na correspondência das imagens na forma de *embeddings* como critério de desempate entre os mapas 5 e 7. Também mapeou-se o consumo de memória RAM e GPU, utilizada por essa abordagem. Prover a informação de consumo de memória servirá como material de suporte para análise na especificação de componentes de hardware para trabalhos futuros que implementem este método em ambiente relevante. A medida de comparação para avaliação baseou-se no tempo de processamento de trabalhos do estado da arte em correspondência de imagens para geolocalização de veículos aéreos. A Tabela 2, adaptada de [28], apresenta as abordagens de correspondências mais utilizadas na área. Para este trabalho, espera-se que a abordagem de correlação cruzada atinja desempenho similar com o apresentado na Tabela 2.

A Tabela 3 demonstra os resultados da média de Tempo de Processamento (TP) dos mapas 5 e 7 em termos da correspondência e ranqueamento para proposta de

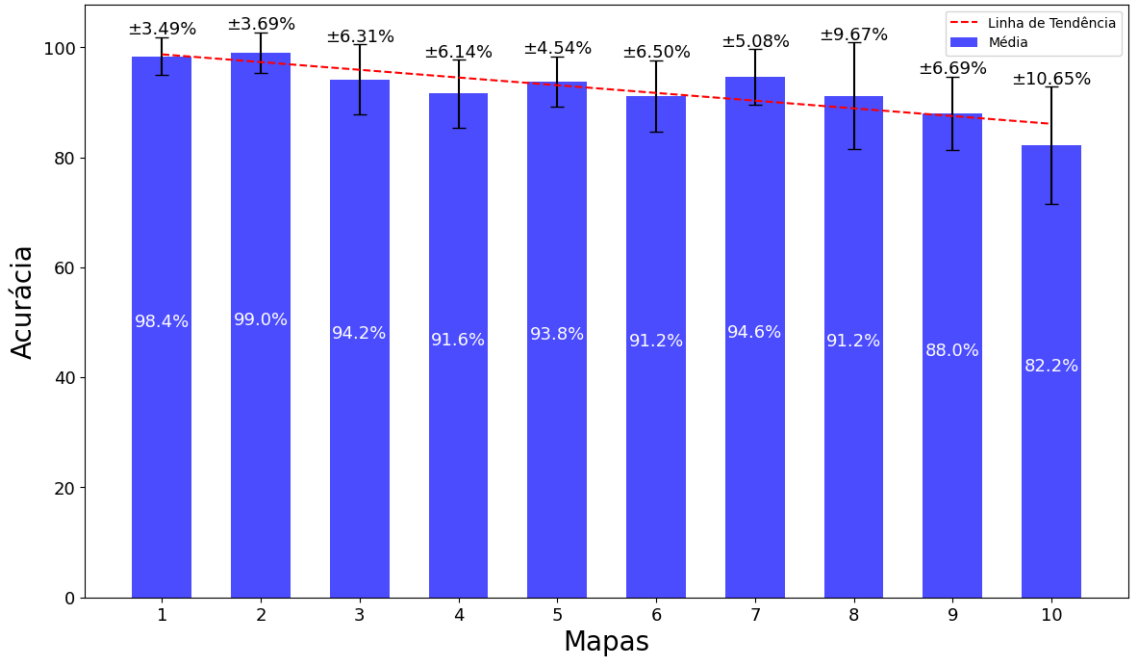


Figura 11: Gráfico com valores de acurácia média e desvio-padrão para 10 tamanhos de mapa.

Método	**Tempo de Processamento (milissegundos)**
abBRIEF	0.3 ± 0.029
BRIEF	0.3 ± 0.029
Mutual Information	28 ± 1
Template	50 ± 1
Histogram	2 ± 1
ORB	77 ± 3
SIFT	1336 ± 104
Correlação cruzada	0.26

Tabela 2: Tempo de processamento para corresponder imagens, referente ao estado da arte em métodos de correspondência.

geolocalização. Em termos de tempo de correspondência não houve mudança significativa entre os dos dois mapas. O algoritmo de correlação cruzada obteve desempenho similar ao esperado em comparação com a abordagem equivalente mostrada na Tabela 2. No entanto, o tempo médio para processar a proposta de geolocalização, que depende da computação do ranqueamento, mostrou-se excessivamente custoso, cerca de 2 s para o Mapa 7.

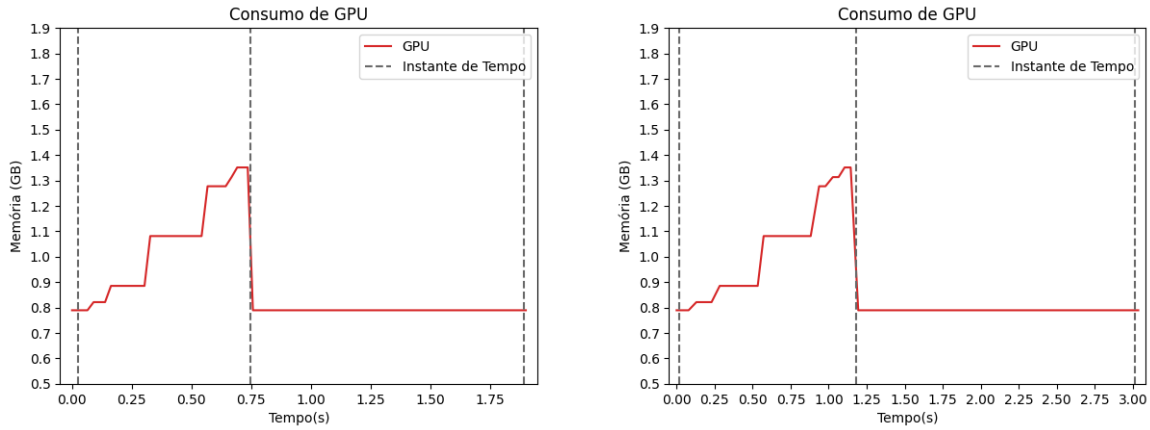
O tempo total para proposta de geolocalização utilizando o mapa 5 é cerca de 0,64 s mais rápido. A diferença entre ambos os mapas na etapa de ranqueamento pode estar diretamente associada ao tamanho dos mapas. Isso é deduzido pelo motivo do algoritmo de ranqueamento fazer a busca percorrendo por completo o vetor de ranque

Mapa	TP Correspondência	TP Ranqueamento
5	0.16 ± 0.0431 ms	1.33 ± 0.0162 s
7	0.14 ± 0.0154 ms	1.97 ± 0.0196 s

Tabela 3: Comparação entre as Médias de Tempos de Processamento (TP) dos Mapas candidatos.

para selecionar o *embedding* que possui maior similaridade com a imagem de voo. Dessa forma, quanto maior o mapa, maior será o tempo de processamento para realizar o ranqueamento, afetando o tempo total da abordagem. Nesse cenário, foi escolhido o mapa 5 para realizar a simulação.

A Figura 12 apresenta o gráfico de uso de GPU associado ao experimento do mapa 5. O melhor resultado é apresentado na Figura 12(a) e o pior resultado na Figura 12(b). Para ambos os gráficos, o instante de tempo 0 s (primeira linha tracejada), corresponde ao início do processo de codificação da imagem de voo, o que eleva o uso da GPU até 1,4 GB. Após codificar a imagem em *embedding*, indicado pela segunda linha tracejada, o consumo da GPU mantem-se constante em aproximadamente 0,8 GB. Entre a segunda linha tracejada e a terceira, ocorre a etapa de correlação cruzada e ranqueamento. Nessa etapa, a GPU não é utilizada pelo algoritmo. O tempo deste algoritmo para codificar uma imagem de voo foi de 0.75 s, para o melhor caso, e 1.18 s para o pior caso.



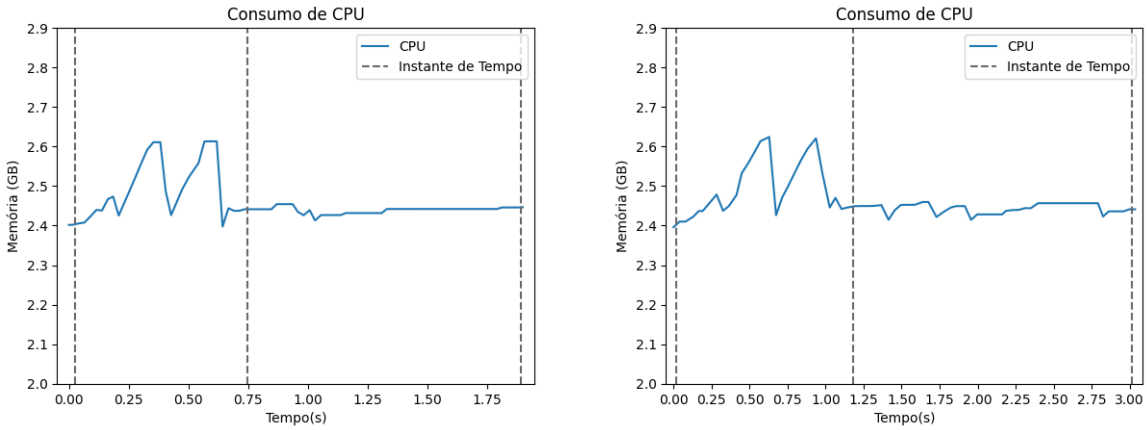
(a) Melhor resultado

(b) Pior resultado

Figura 12: Comparação em termos de tempo de processamento do consumo de GPU associado ao *encoder* no processo de codificação de imagem de voo.

Já a Figura 13 evidencia o uso da memória RAM do experimento no mapa 5. O melhor resultado é apresentado na Figura 13(a) e o pior resultado na Figura 13(b). Para os dois casos, nota-se que no intervalo de tempo que ocorre a codificação da imagem (entre a primeira e a segunda linha tracejada), a memória RAM atinge duas vezes o pico de uso em 2,6 GB. Nesse período, a imagem do voo é submetida a etapas de pré-processamento para ficar em conformidade com o shape de entrada da rede neural. Entre a segunda e

terceira linha tracejada, ocorre a etapa de proposta de localização. Nessa parte a memória RAM mantem-se entre 2,4 e 2,5 GB.



(a) Melhor resultado

(b) Pior resultado

Figura 13: Consumo de CPU associado ao pré-processamento da imagem de voo e durante a proposta de localização (correspondência e ranqueamento).

A Tabela 4 apresenta os resultados de tempo de processamento para o melhor e o pior caso no experimento do Mapa 5.

Experimento mapa 5	**TP do *encoder***	**TP Proposta de Localização**	**TP Total**
Melhor caso	0.75 s	1.15 s	1.90 s
Pior caso	1.18 s	1.84 s	3.02 s

Tabela 4: Tempo de processamento para o melhor e o pior caso no experimento do Mapa 5.

A simulação do método como algoritmo de geolocalização foi realizada no mapa 5. O caminho proposto possui aproximadamente 200 km em linha reta. Esse trajeto simula a captura de 126 imagens, referentes ao mapa de voo, no sentido sul para norte, uma vez que todas as imagens estão orientadas para o norte. As Figuras 14 (referente à chegada) e 15 (referente à saída) evidenciam o mapa 5 durante a simulação. Cada retângulo representa uma imagem georreferenciada - para fins de simplificação, projetou-se apenas a geometria da imagem. Os pontos em verde e amarelo desenham o caminho. Os pontos em verde representam às coordenadas da imagem georreferenciada em uma correspondência correta. Os pontos amarelos são sobrepostos pelos pontos verdes, pois servem apenas para indicação do caminho. Em vermelho são os pontos referentes às coordenadas de correspondências erradas. Nessa simulação a abordagem correspondeu corretamente 96,83% das imagens, acima do mínimo estipulado em 89,26%. O voo foi simulado de sul à norte para evitar lidar com a questão de orientação das imagens. Ressalva-se que para uma aplicação real, além de traçar o caminho de voo e capturar as imagens seguindo a orientação do caminho, essa abordagem precisa retreinar o autoencoder para cada caminho que não esteja contido no mapa global. E também, para uma aplicação real seria

necessário definir um modelo de voo da aeronave para atuar como limitador da janela de busca, semelhante à abordagem de [5]. Este modelo pode ser otimizado a partir da performance que se é desejada, uma vez que quanto menor for a janela de busca melhor a performance da aeronave, aumentando a velocidade de resposta e diminuindo as chances de erro.

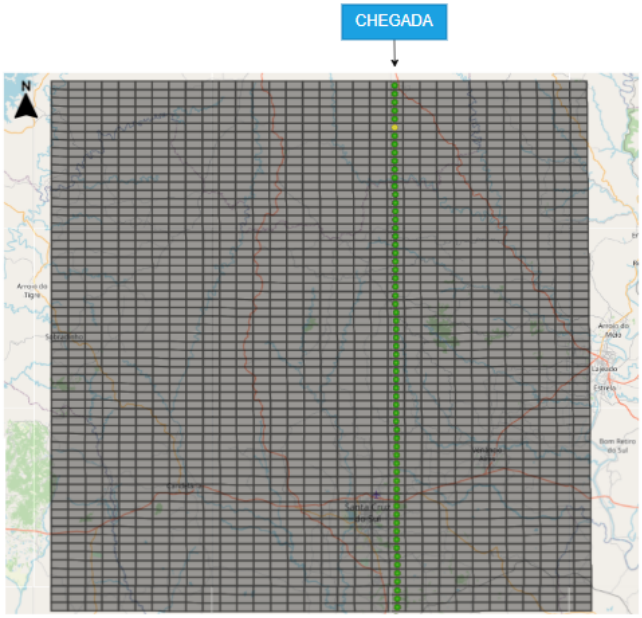


Figura 14: Simulação - parte de chegada.

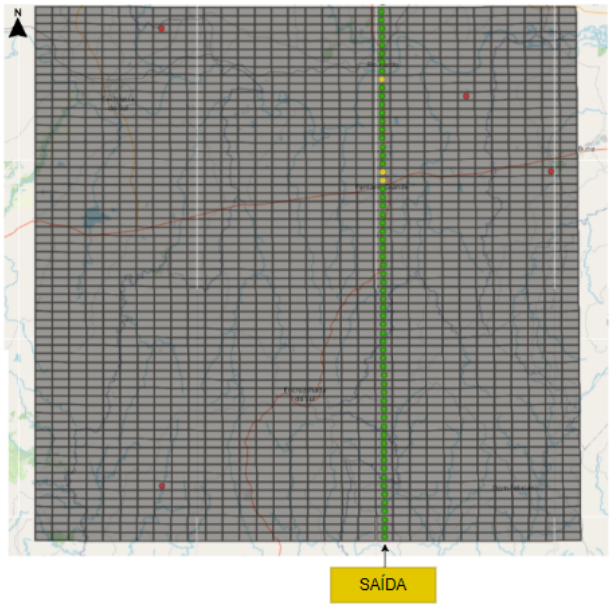


Figura 15: Simulação - parte de saída.

Conforme já mencionado, a imagem capturada pela aeronave muitas vezes difere em textura, brilho e aparência da imagem usada como referência para treinamento do modelo. Portanto, obter uma abordagem que seja robusta a essas variações é imprescindível. A Figura 16 a seguir, faz uma comparação entre as imagens da simulação, voo e referência, que apresentaram variação de aparência e o método foi capaz de corresponder corretamente.

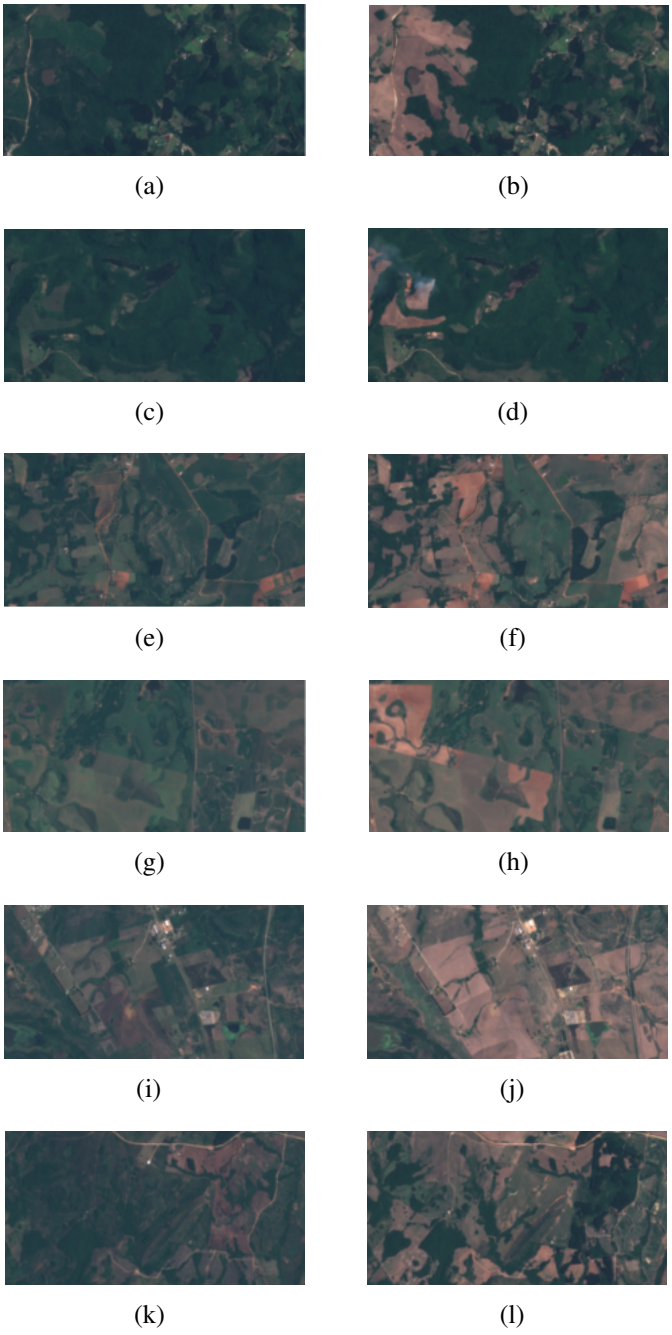
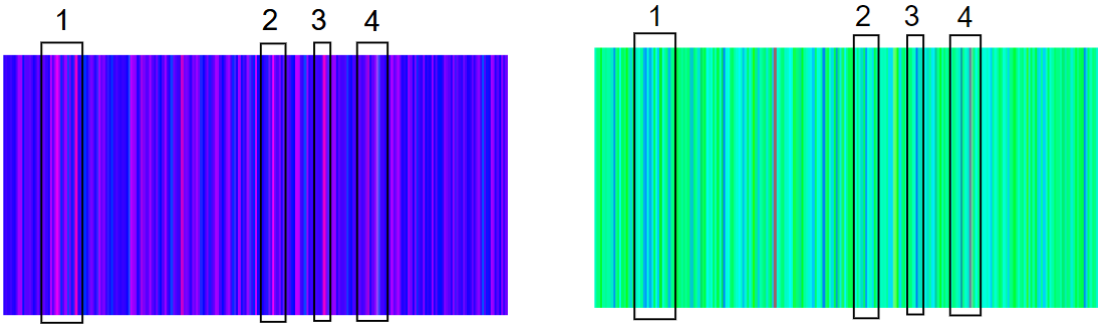


Figura 16: Pares mapeados corretamente.

Em um primeiro instante, ao investigar puramente a imagem bruta, as Figuras 16(b), 16(j) e 16(l) não aparentam ser correspondentes ou possuir semelhança com

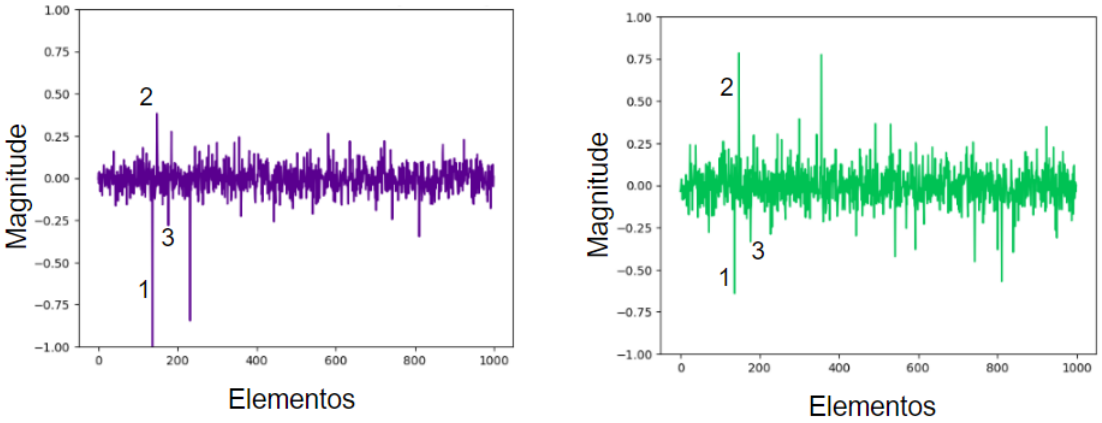
as imagens de voo associadas. No entanto, tome como exemplo as representações das Figuras 16(k) e 16(l), na forma de *embedding* no espaço de cores HSV e em séries temporais. A representação em HSV torna possível encontrar variações de níveis de cores e brilhos distintos com ocorrência nas mesmas posições dos vetores de *embeddings*. Isto é ilustrado na Figura 17, em que os retângulos pretos de mesma numerações correspondem à variações de níveis de cor e brilho nas mesmas posições dos *embeddings*. A diferença de cores entre a Figura 17(a) (predominantemente em roxo) e a Figura 17(b) (predominantemente em verde) ilustra a diferença de cores e brilho entre as Figuras 16(k) e 16(l).



(a) Imagem de voo (b) Imagem de referência

Figura 17: *Embeddings* representados no espaço e cores HSV.

Já com a representação em série temporal é possível visualizar a magnitude de uma similaridade em posições iguais, o que na representação HSV não deixa evidente. A Figura 18 apresenta o *embedding* referente à imagem da Figura 16(k) e o *embedding* referente à imagem de sua localização correspondente, Figura 16(l), no formato de séries temporais. As numerações indicam elementos correspondentes.



(a) Imagem de voo (b) Imagem de referência

Figura 18: *Embeddings* representados em séries temporais.

Essa análise foi apresentada para ilustrar as similaridades existentes que não são vistas na análise bruta das imagens. O algoritmo de correlação cruzada atua quantificando a

similaridade entre dois *embeddings* elemento a elemento. Os elementos similares tendem a valores próximos de 1. Quanto mais distintos forem dois elementos, mais próximo de -1 será o seu valor. Ao final da correlação cruzada o produto escalar corresponde ao grau de similaridade entre dois *embeddings*. Quanto maior esse número, mais semelhantes as imagens são.

A Figura 19 apresenta as correspondências incorretas durante a simulação. A primeira coluna representa as imagens de voo, a segunda coluna a imagem de referência localizada e a terceira a imagem de referência correta. Ao fazer análise bruta das imagens, pode-se avaliar que as Figuras 19(a), 19(d) e 19(j) referentes à captura em voo, parecem mais semelhante, respectivamente, com as imagens localizadas 19(b), 19(e) e 19(k) do que com as corretas 19(c), 19(f) e 19(l), respectivamente.

(a) Voo (b) Localizada (c) Correta

(d) Voo (e) Localizada (f) Correta

(g) Voo (h) Localizada (i) Correta

(j) Voo (k) Localizada (l) Correta

Figura 19: Correspondências incorretas durante a simulação.

6 CONCLUSÃO

Este trabalho explorou o uso de um algoritmo baseado em rede neural do tipo auto-encoder em conjunto com algoritmo de correlação cruzada, como uma abordagem para realizar a correspondência entre imagens de satélite no problema de geolocalização de veículos aéreos utilizando visão computacional.

Como inovação, explorou-se a capacidade deste método em termos de acurácia e custo computacional, para corresponder imagens de satélite de médias altitudes em dois conjuntos de dados de anos distintos, que possuem diferenças de aparência significativa nas imagens. Foi demonstrado que o método implementado é capaz de aprender representações discriminativas de imagens de satélite de médias altitudes, uma vez que durante os experimentos foi obtido acurácia de aproximadamente 90% em todos experimentos, exceto no experimento com a área total do mapa, que obteve 82% de acurácia.

Durante os experimentos mapeou-se o consumo de memória RAM e da GPU, bem como o tempo de processamento para realizar a correspondência entre as imagens. Neste ponto, verificou-se que o algoritmo possui um gargalho relacionado ao método de ranqueamento para proposta de localização. Neste trabalho a proposta de localização atual é o que mais consome tempo de processamento, o que torna a resposta do algoritmo mais lenta do que o método apresentado por [5]. Quanto maior o mapa, mais este problema é agravado.

Também foi demonstrado que este método possui potencial para ser parte integrada em um algoritmo de geolocalização. Durante a simulação, foi correspondido corretamente 96,83% da imagens aéreas simuladas em uma mapa de aproximadamente 200 km.

E ainda, este trabalho disponibilizou material e procedimentos referente a criação de mapas georreferenciados com auxílio de computação em nuvem na plataforma Google Earth Engine.

Como trabalhos futuros, é possível destacar a melhoria do algoritmo de ranqueamento para diminuir o tempo de processamento do modelo. Com a finalidade de aumentar a robustez desta abordagem, também seria interessante treinar a rede neural em imagens de satélite com diferente orientações, conforme realizada em [5], para melhorar a sua capacidade de correspondência.

REFERÊNCIAS

[1] Ali, B., Sadekov, R., and Tsodokova, V. (2023). A review of navigation algorithms for unmanned aerial vehicles based on computer vision systems. *Gyroscopy and Navigation*, 13(4):241–252.

[2] Ashraf, S., Aggarwal, P., Damacharla, P., Wang, H., Javaid, A. Y., and Devabhaktuni, V. (2018). A low-cost solution for unmanned aerial vehicle navigation in a global positioning system–denied environment. *International Journal of Distributed Sensor Networks*, 14(6). Acessado em 17 de julho de 2023.

[3] Basu, A. (1995). Active calibration of cameras: theory and implementation. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(2):256–265. Acessado em 08 maio de 2023.

[4] Belo, F. A. (2006). *Desenvolvimento de Algoritmos de Exploração e Mapeamento Visual para Robôs Móveis de Baixo Custo*. PhD thesis, PUC-Rio. Acessado em 08 maio de 2023.

[5] Bianchi, M. and Barfoot, T. D. (2021). Uav localization using autoencoded satellite images.

[6] Buslaev, A., Seferbekov, S., Iglovikov, V., and Shvets, A. (2018). Fully convolutional network for automatic road extraction from satellite imagery. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 197–1973.

[7] Casagrande, V. (2021). A que altitude os aviões podem chegar e como isso afeta seu voo? Acessado em 08 agosto 2023.

[8] Deepak Birla (2022). Autoencoders. Acessado em 05 de janeiro de 2024.

[9] Derrick, T. and Thomas, J. (2004). Time-series analysis: The cross-correlation function. In Stergiou, N., editor, *Innovative Analyses of Human Movement*, pages 189–205. Human Kinetics Publishers, Champaign, Illinois.

[10] EMBRAPA (2013). Satélites e monitoramento. Acessado em 02 maio de 2023.

[11] ESA (2023). Sentinel. Acessado em 17 maio de 2023.

[12] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

[13] Fitzgibbon, A. W. and Zisserman, A. (1998). Automatic camera recovery for closed or open image sequences. In *Computer Vision—ECCV'98: 5th European Conference on Computer Vision Freiburg, Germany, June, 2–6, 1998 Proceedings, Volume I 5*, pages 311–326. Springer.

[14] Google (2023). Google colab. Acessado em 08 agosto 2023.

[15] Google (2023). Google earth. https://www.google.com.br/earth/. Acessado em 08 agosto 2023.

[16] Google Earth Engine (2023). Google earth engine python installation guide. Acessado em 08 agosto 2023.

[17] Guerra, R. d. S. (2004). Calibração automática de sistemas de visão estéreo a partir de movimentos desconhecidos.

[18] GÓMEZ-Reyes, J., Benítez-Rangel, J., Morales-Hernández, L., Resendiz-Ochoa, E., and Camarillo-Gomez, K. (2022). Image mosaicing applied on uavs survey. *Applied Sciences*, 12(5):2729. Acessado em 29 de junho de 2023.

[19] Hartford, T. (2019). Could the world cope if gps stopped working? Acessado em 26 abril 2023.

[20] Hinton, G. E., Krizhevsky, A., and Wang, S. D. (2011). Transforming auto-encoders. In Honkela, T., Duch, W., Girolami, M., and Kaski, S., editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 44–51, Berlin, Heidelberg. Springer Berlin Heidelberg.

[21] IBGE (2023). Atlas escolar. Acessado em 26 abril 2023.

[22] Jakhar, P. (2020). Bds: como é o novo sistema de navegação por satélite chinês que quer concorrer com o americano gps. Acessado em 02 maio de 2023.

[23] Kim, D.-K. and Walter, M. R. (2017). Satellite image-based localization via learned embeddings. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*.

[24] Kingma, D. P. and Welling, M. (2022). Auto-encoding variational bayes.

[25] Kinnari, J., Renzulli, R., Verdoja, F., and Kyrki, V. (2023). Lsvl: Large-scale season-invariant visual localization for uavs. *Robotics and Autonomous Systems*, 168:104497.

[26] Li, Y., Fu, C., Huang, Z., Zhang, Y., and Pan, J. (2021). Intermittent contextual learning for keyfilter-aware uav object tracking using deep convolutional feature. *IEEE Transactions on Multimedia*, 23:810–822. Acessado em 16 maio de 2023.

[27] Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110.

[28] Mantelli, M., Pittol, D., Neuland, R., Ribacki, A., Maffei, R., Jorge, V., Prestes, E., and Kolberg, M. (2019). A novel measurement model based on abbrief for global localization of a uav over satellite images. *Robotics and Autonomous Systems*, 112:304–319. Acessado em 24 de janeiro 2023.

[29] Mohajerani, Z. (2008). Vision-based uav pose estimation. Master's thesis, Northeastern University, Boston, Massachusetts. Acessado em 08 agosto de 2023.

[30] NASA (2023). Moderate resolution imaging spectroradiometer. Acessado em 02 maio de 2023.

[31] of Space, D. (2023). Indian space research organisation. Acessado em 02 maio de 2023.

[32] Pastório, A. F. and Camargo, E. T. d. (2021). Técnicas de geolocalização em redes lorawan como abordagem de tolerância a falhas para dispositivos iot baseados em gps. In *Anais do Workshop de Testes e Tolerância a Falhas*, pages 29–42. Sociedade Brasileira de Computação. Acessado em 26 abril 2023.

[33] Patel, B., Barfoot, T. D., and Schoellig, A. P. (2020). Visual localization with google earth images for robust global pose estimation of uavs. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6491–6497.

[34] Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster r-cnn: Towards real-time object detection with region proposal networks.

[35] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

[36] Saraiva, M., Sousa, G., Marreira, R., and Pinheiro, F. (2022). Correlação entre a exatidão da medida de posição do gps e as condições atmosféricas. *Revista Brasileira de Meteorologia*. Acessado em 26 abril 2023.

[37] Secretariat, N. S. P. (2023). Quashi-zenith satellite system. Acessado em 02 maio de 2023.

[38] Sivakumar, M. and TYJ, N. M. (2021). A literature survey of unmanned aerial vehicle usage for civil applications. *Journal of Aerospace Technology and Management*, 13:e4021.

[39] Tanchenko, A., Fedulin, A., Bikmaev, R., and et al. (2020). Uav navigation system autonomous correction algorithm based on road and river network recognition. *Gyroscopy Navig.*, 11:293–299.

[40] University of California at Santa Barbara (2023). UCSB Library. Acessado em 08 agosto de 2023.

[41] USGS (2023). Earthexplorer. Acessado em 02 maio de 2023.

A SIRGAS 2000

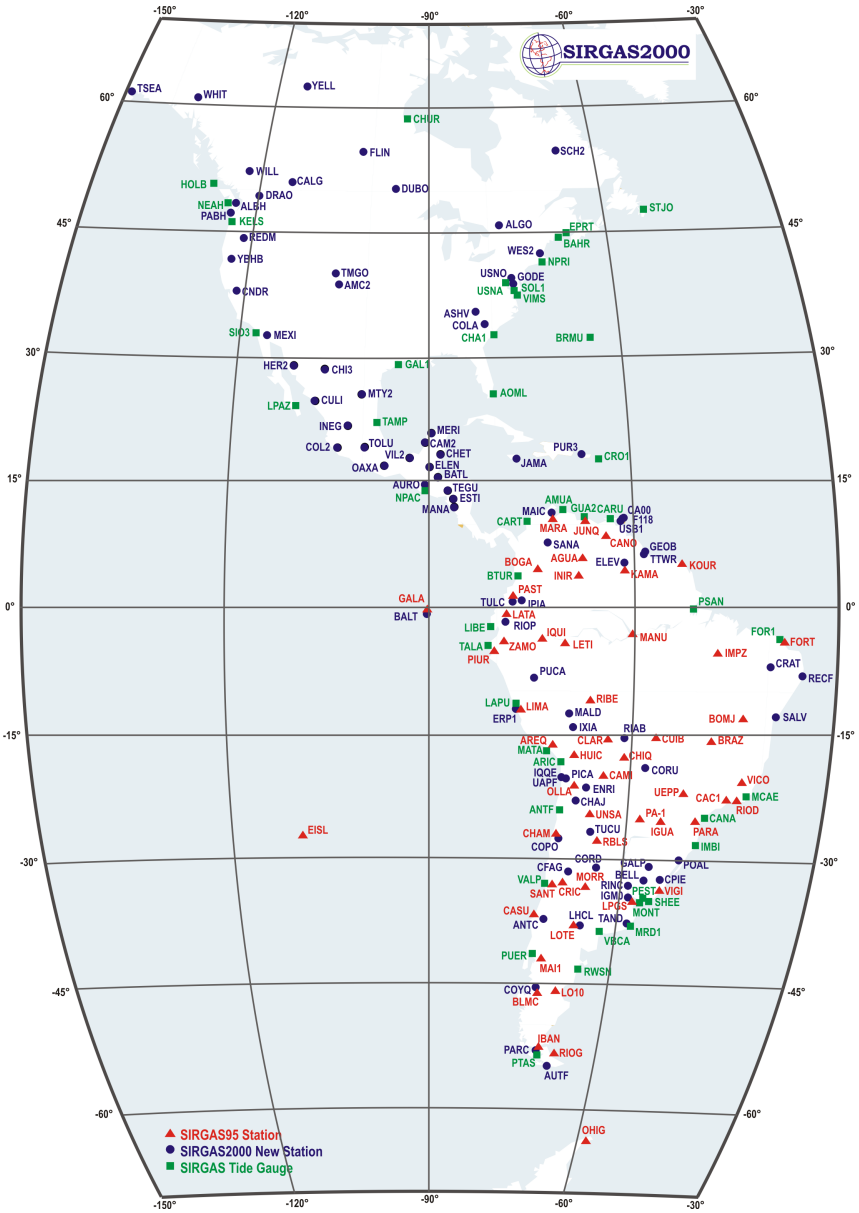


Figura 20: Mapa SIRGAS 2000.

B PROCEDIMENTO DE DEFINIÇÃO DE ESCALA PARA ALTITUDE

Para simular a escala de uma imagem de satélite do GEE à 6000 m de altitude foi primeiro pesquisado uma base de dados que possui imagens reais de voos na referida altitude. Nesse contexto, a Universidade da Califórnia em Califórnia [40] provem um vasto catalogo[1] de imagens aéreas em voos de diversas regiões do estado da Califórnia . Esta base de dados também disponibiliza um relatório de imagens, que contém informações sobre a data do voo, altitude, orientação câmera utilizada e distância focal da lente.

Como referência utilizou a imagem pw sb 86 (Figura 21(a)) do Flight PW-SB-15, que foi fotografada à 6000 m de altitude na cidade de Santa Bárbara. De posse de uma imagem de base de dados confiável, utilizou-se comparação visual para posicionar as imagens do Google Earth em altitude similar. Por fim, definiu-se a escala de zoom a ser configurada no GEE como o fator 14, que foi índice que mais próximo correspondeu à 6000 m de altitude. A Figura 21 apresenta a comparação da imagem de referência 21(a), com as imagens da plataforma Google Earth Pro 21(b), Google Earth Engine 21(c) e Google Maps 21(d) para altitude de aproximadamente 6000 m.

(a) Universidade da Califórnia (b) Google Earth Pro (c) Google Earth Engine (d) Google Maps

Figura 21: Comparação da imagem aérea pw-sb-86de com imagens de satélite posicionadas a aproximadamente 6000 m de altitude.

[1] Catálogo de imagens aéreas da Universidade da Califórnia em Santa Bárbara disponível em Catálogo UCSB.

C CAMADAS DO AUTOENCODER

A Figura 22 apresenta as camadas da rede neural.

Camada (tipo)	Formato da saída	Nº de parâmetros
Conv2d-1	[-1, 128, 160, 80]	6.272
BatchNorm2d-2	[-1, 128, 160, 80]	256
LeakyReLU-3	[-1, 128, 160, 80]	0
Conv2d-4	[-1, 256, 80, 40]	524.544
BatchNorm2d-5	[-1, 256, 80, 40]	512
LeakyReLU-6	[-1, 256, 80, 40]	0
Conv2d-7	[-1, 512, 40, 20]	2.097.664
BatchNorm2d-8	[-1, 512, 40, 20]	1.024
LeakyReLU-9	[-1, 512, 40, 20]	0
Conv2d-10	[-1, 1.024, 20, 10]	8.389.632
BatchNorm2d-11	[-1, 1.024, 20, 10]	2.048
LeakyReLU-12	[-1, 1.024, 20, 10]	0
Conv2d-13	[-1, 1.024, 10, 5]	16.778.240
BatchNorm2d-14	[-1, 1.024, 10, 5]	2.048
LeakyReLU-15	[-1, 1.024, 10, 5]	0
Linear-16	[-1, 1000]	51.201.000
Linear-17	[-1, 51200]	51.251.200
ReLU-18	[-1, 51200]	0
UpsamplingNearest2d-19	[-1, 1024, 20, 10]	0
ReplicationPad2d-20	[-1, 1024, 22, 12]	0
Conv2d.21	[-1, 1024, 20, 10]	9.438.208
BatchNorm2d-22	[-1, 1024, 20, 10]	2.048
LeakyReLU-23	[-1, 1024, 20, 10]	0
UpsamplingNearest2d-24	[-1, 1024, 40, 20]	0
ReplicationPad2d-25	[-1, 1024, 42, 22]	0
Conv2d-26	[-1, 512, 40, 20]	4.719.104
BatchNorm2d-27	[-1, 512, 40, 20]	1.024
LeakyReLU-28	[-1, 512, 40, 20]	0
UpsamplingNearest2d-29	[-1, 512, 80, 40]	0
ReplicationPad2d-30	[-1, 512, 82, 42]	0
Conv2d-31	[-1, 256, 80, 40]	1.179.904
BatchNorm2d-32	[-1, 256, 80, 40]	512
LeakyReLU-33	[-1, 256, 80, 40]	0
UpsamplingNearest2d-34	[-1, 256, 160, 80]	0
ReplicationPad2d-35	[-1, 256, 162, 82]	0
Conv2d-36	[-1, 128, 160, 80]	295.040
BatchNorm2d-37	[-1, 128, 160, 80]	256
LeakyReLU-38	[-1, 128, 160, 80]	0
UpsamplingNearest2d-39	[-1, 128, 320, 160]	0
ReplicationPad2d-40	[-1, 128, 322, 162]	0
Conv2d-41	[-1, 3, 320, 160]	3.459
Sigmoid-42	[-1, 3, 320, 160]	0

Figura 22: Descrição das camadas da rede neural.