

Victor José Ferreira de Moraes

**Prova de conceito de um modelo de
classificação de áreas irregulares na Floresta
Amazônica baseado em Transformers Visuais**

Belo Horizonte, Minas Gerais, Brasil

25 de Novembro de 2022

Moraes, Victor Moraes

Prova de conceito de um modelo de classificação de áreas irregulares na Floresta Amazônica baseado em Transformers Visuais/ Victor José Ferreira de Moraes. – Belo Horizonte, Minas Gerais, Brasil, 25 de Novembro de 2022-

64 p. : il. (algumas color.) ; 30 cm.

Orientador: Cristiano Castro

Monografia (Graduação) – Universidade Federal de Minas Gerais UFMG
Escola de Engenharia

Departamento de Engenharia Elétrica , 25 de Novembro de 2022.

1. Engenharia Elétrica 2. Reconhecimento de Padrões. 3. Manipulador Robótico.
4. ROS. I. Cristiano Castro. II. Universidade Federal de Minas Gerais. III. Escola de Engenharia da UFMG. IV. Departamento de Engenharia Elétrica V. Título

Victor José Ferreira de Moraes

**Prova de conceito de um modelo de classificação de áreas
irregulares na Floresta Amazônica baseado em
Transformers Visuais**

Monografia de conclusão de curso para obtenção do nível de Bacharel em Engenharia Elétrica pela Escola de Engenharia da Universidade Federal de Minas Gerais.

Trabalho aprovado. Belo Horizonte, Minas Gerais, Brasil, 10 de Dezembro de 2022:

Cristiano Castro
Orientador

Belo Horizonte, Minas Gerais, Brasil
25 de Novembro de 2022

*Este trabalho é dedicado à Rita e ao José,
que tanto me nutriram com amor e acreditaram em mim.*

Agradecimentos

Gostaria de agradecer a minha família que tanto apoiou e acreditou no poder transformador da educação.

Aos meus amigos companheiros da breve-longa jornada da graduação, que a tornaram mais agradável e instigante.

Aos professores, por pela dedicação à docência, por inspirarem o fascínio ao conhecimento e ciência.

E gratidão a todos os profissionais da universidade, que possibilitaram a UFMG ser a grande instituição que é.

*Em algum lugar,
há algo incrível esperando para ser descoberto.*

Carl Sagan

*Qualquer tecnologia suficientemente avançada
é indistinguível de magia.*

Arthur C. Clarke

Resumo

A Floresta Amazônica sofreu um aumento preocupante número de novos casos de desmatamento e de garimpos em terras protegidas na década anterior, além de garimpos fluviais no Rio Madeira em 2022. Com isso, agências de monitoramento requerem modernas técnicas para a identificação de irregularidades capazes de auxiliar na detecção de pequenas regiões no vasto território da bacia amazônica, para manutenção de políticas de preservação ambiental. Novas técnicas de visão computacional, reconhecimento de padrões e aprendizado profundo permitem soluções e a realização de tarefas em melhor nível de resolução e desempenho no contexto de sensoreamento remoto.

Partindo desse ponto, este trabalho estuda as limitações das técnicas já estabelecidas de redes convolucionais e realiza uma prova de conceito de uma técnica de visão computacional de arquitetura Swin para classificação de imagens de áreas irregulares da Amazônia, baseado em Transformers visuais. Além de prover uma metodologia e experimentos reprodutíveis para utilização da arquitetura Swin para problemas análogos, de diferentes conjuntos de dados de sensoriamento remoto.

Este trabalho documenta o experimento realizado em etapas como: aspectos da criação do ambiente em nuvem utilizando a plataforma GoogleColab para melhor explorar seus recursos computacionais; análise exploratória dos dados; construção um modelo base vascülhando diferentes configurações, componentes e características de treinamento, utilizando bibliotecas como Pytorch e SciKitLearn.

Dessa forma, foi desenvolvido um modelo base de uma arquitetura de rede neural convolucional ResNet50 com bom desempenho global utilizando métricas de classificação como a F_2 , e curva PR. Contudo, as classes raras do conjunto de dados, tiveram um mal desempenho. Seguindo as mesmas configurações de treino, o segundo modelo de arquitetura Swin foi ajustado e comparado com o baseado em ResNet50. Apresentou relevantes melhorias de desempenho e melhor capacidade de generalização com poucas amostras sob condições climáticas variadas na classificação de classes raras. Demonstrando assim a compatibilidade desta arquitetura para a aplicação.

Palavras-chave: Sensoriamento remoto; Transformers Visuais; Transformer Swin; Aprendizado profundo; Redes Neurais Convolucionais; Aprendizado de Máquina; Reconhecimento de Padrões; PyTorch; Visão computacional.

Abstract

The Amazon Forest has suffered a worrying increase in the number of new cases of deforestation and artisanal mining on protected lands in the previous decade, in addition to fluvials on the Madeira River in 2022. As a result, monitoring agencies require modern techniques to identify irregularities capable of helping in the detection of small regions in the vast territory of the Amazon basin, for the maintenance of environmental preservation policies. New techniques of computer vision, pattern recognition and deep learning allow solutions of tasks at better levels of resolution and performance in the context of remote sensing.

Based on this field, this work studies the limitations of the already established techniques of convolutional networks and seeks to carry out a proof of concept of a computer vision technique of Swin architecture for classifying images of irregular areas of the Amazon, based on visual transformers. In addition to providing a methodology and reproducible experiments for using the Swin architecture for similar problems, from different sets of remote sensing data.

This work documents the experiment carried out with steps such as: aspects of creating the cloud environment using the GoogleColab platform in order to better exploit its computational resources; exploratory data analysis; build a base model by searching through different configurations, components and training features, using libraries such as Pytorch and SciKitLearn.

In this way, a base model of a ResNet50 convolutional neural network architecture with good performance was developed using classification metrics such as F_2 , and PR curve, except for the rare classes of the dataset, which had a poor performance. Following the same training settings, the second Swin architecture model was fitted and compared with the one based on ResNet50. It presented relevant performance improvements and better generalization with few samples under varied climate conditions in the classification of rare classes. Thus demonstrating the capability and compatibility of this architecture for the application.

Keywords: Remote sensing; Visual Transformers; Transformer Swin; Deep learning; Convolutional Neural Networks; Machine Learning; Pattern Recognition; PyTorch; Computer vision.

Listas de ilustrações

Figura 1 – Garimpo ilegal na Terra Indígena Munduruku, município de Jacareacanga. Foto: Marizilda Cruppe/Amazônia Real	16
Figura 2 – Sensoriamento remoto Foto:(INSTRUTORGIS, 2022)	18
Figura 3 – Ilustração das métricas de precisão e revocação. Fonte: WikiCommons .	21
Figura 4 – Matriz Confusão, Equações de Precisão, revocação, F_β e F_2	21
Figura 5 – Função de ativação <i>Sigmoid</i> . Fonte: WikiCommons	23
Figura 6 – Exemplo de algoritmo de otimização: gradiente descendente estocástico.	23
Figura 7 – Exemplo de uma rede de perceptrons de multiplas camadas - MLP. . .	24
Figura 8 – Arquitetura de uma rede convolucional. Filtros extratores de características são aplicados em diferentes resoluções e campos visuais. A saída de cada imagem convoluta alimenta a próxima camada. As últimas camadas completamente conectadas realizam a classificação.	25
Figura 9 – Filtro convolucional que aplica uma janela deslizante aplicando operação de convolução e pooling.	25
Figura 10 – Arquitetura de modelos ResNet e conexões saltos. Fonte:WikiCommons	26
Figura 11 – O ViT divide uma imagem em uma grade de recortes quadrados, cada fragmento é achatado em um vetor único contendo todos os canais de todos os pixels, e projetando-os em uma dimensão de entrada desejada, alimentando a camada de múltiplos Encoders em paralelo. (DOSOVITSKIY et al., 2020)	28
Figura 12 – Esquerda: ViT aprende a estrutura de grade via representações de posição. Direita: Camadas inferiores do ViT contem ambas características locais e globais, o quanto mais profundas as camadas, mais globais as características.(DOSOVITSKIY et al., 2020)	29
Figura 13 – O modelo lida com as regiões que são semanticamente relevantes para classificação. (CHEN; HSIEH; GONG, 2021)	30
Figura 14 – Arquitetura swin (LIU et al., 2022)	32
Figura 15 – Amostras de classes do dataset Amazônia do Espaço Fonte:(PLANETCO, 2016)	34
Figura 16 – Amostragem de cada classe de rótulo. Fonte: Autor	39
Figura 17 – Matriz de co-ocorrência. Fonte: Autor	40
Figura 18 – Agrupamento de amostras via técnica TSNE. Fonte: Autor	40
Figura 19 – Arquitetura de modelo base. Fonte:WikiCommons	41
Figura 20 – Arquitetura de modelo proposto. Fonte: (LIU et al., 2022)	41
Figura 21 – Métrica F2 em treino e validação por época para a rede ResNet50. Fonte: Autor	44

Figura 22 – Métrica de perda em treino e validação por época para a rede ResNet50. Fonte: Autor	45
Figura 23 – Métrica F2 em treino e validação por época para a rede Swin-T. Fonte: Autor	46
Figura 24 – Métrica de perda em treino e validação por época para a rede Swin-T. Fonte: Autor	47
Figura 25 – Matriz Confusão para o modelo base Resnet-50. Fonte: Autor	49
Figura 26 – Curva COR para o modelo base. Fonte: Autor	51
Figura 27 – Curva PR para o modelo base. Fonte: Autor	51
Figura 28 – Arquitetura de modelo Swin-T proposto. Fonte: Autor	52
Figura 29 – Probabilidade de inferência para cada classe. Fonte: Autor	54
Figura 30 – Matriz Confusão SwinT. Fonte: Autor	61
Figura 31 – Curva COR Swin-T. Fonte: Autor	62
Figura 32 – ACurva PR Swin-T. Fonte: Autor	63

Lista de tabelas

Tabela 1 – Proporção de Classes do conjunto de dados <i>Planet</i>	38
Tabela 2 – Características do modelo fornecida pela biblioteca	41
Tabela 3 – Resultados do Modelo ResNet50	50
Tabela 4 – Classes Raras Dataset <i>Planet</i> — Valores do conjunto de dados inteiro .	50
Tabela 5 – Resultados do Modelo Base	50
Tabela 6 – Resultados do Modelo proposto Swin-T	52
Tabela 7 – Comparação de resultados da métrica F2 entre Modelo base e proposto.	52
Tabela 8 – Comparação de resultados da métrica PR-AUC entre Modelo base e proposto.	53
Tabela 9 – Resultados do Modelo Swin-T	63
Tabela 10 – Resultados do Modelo ResNet50	64

List of abbreviations and acronyms

CPU	Unidade de Processamento Central do inglês <i>Central Process Unit</i>
TPU	Unidade de Processamento de Tensores do inglês <i>Tensor Process Unit</i>
GPU	Unidade de Processamento Gráfico do inglês <i>Central Process Unit</i>
ViT	Transformer visual do inglês <i>Visual Transformer</i>
ADAM	Otimizador adaptativo com momento do inglês <i>Adaptive Momentum Optimizer</i>
MLP	múltiplas (MLP) camadas de perceptrons do inglês <i>Multi Layer Perceptron</i>
LN	Normalização de camada do inglês <i>Layer Normalization</i>
Swin	Transformer de janela deslocada do inglês <i>Shifted window transformer</i>
CNN	Rede Neural Convolucional do inglês <i>Convolutional Neural Network</i>
Resnet	Rede Neural residual inglês <i>Residual network</i>
BCE	Entropia binária cruzada do inglês <i>Binary Cross Entropy</i>
PR	Precisão-Revocação do inglês <i>Precision Recall</i>
AUC	Área abaixo da curva do inglês <i>Area Under The curve</i>
ROC	Característica de operador receptor do inglês <i>Receive Operator Characteristic</i>
MINDS ^{Lab}	Laboratório de Machine learning, Inteligencia computacional e Data Science
UFMG	Universidade Federal de Minas Gerais
t-SNE	Representação de vizinhos estócastica T-Distribuida do inglês <i>T-distributed Stochastic Neighbor Embedding</i>

List of symbols

p	Número de Classificações Positivas
n	Número de Classificações Negativas
tp	Número de Classificações Verdadeiro Positivas
tn	Número de Classificações Verdadeiro Negativas
fp	Número de Classificações Falso positivas
fn	Número de Classificações Falso Negativas
P	Precisão
R	Revocação
F_2	Métrica F2 de desempenho baseadas em índices de precisão e revocação.
BCE	Métrica de perda Entropia binária cruzada do inglês <i>Binary Cross Entropy</i>

Sumário

1	INTRODUÇÃO	15
2	REVISÃO BIBLIOGRÁFICA	18
2.1	<i>Domínio do problema</i>	18
2.2	Revisão Teórica	19
2.2.1	Aprendizado de máquina	19
2.2.2	Métricas de desempenho	19
2.2.3	Função de perda	22
2.2.4	Redes Neurais Artificiais	22
2.2.5	Redes Neurais Artificiais Profundas	23
2.2.6	Redes Neurais Convolucionais	24
2.2.7	<i>ResNet</i>	26
2.2.8	<i>O problema de rotulagem e variabilidade de amostras de treino</i>	26
2.2.9	<i>Aprendizado semi-supervisionado</i>	27
2.2.10	<i>Transferência de aprendizado</i>	27
2.3	Transformer Visual	27
2.3.1	<i>Arquitetura do transformer visual</i>	28
2.4	Trabalhos anteriores	30
2.4.1	Problema de detecção de mudanças em minas de menor escala	30
2.4.2	Classificação de minas e represas	30
2.4.3	Pré-treino de <i>transformers</i> visuais para sensoriamento remoto	31
2.4.4	Transformer Swin	31
2.4.5	ForestViT Modelo de classificação de desmatamento	31
3	METODOLOGIA	33
3.1	Conjunto de dados	33
3.1.1	Dataset Amazônia do Espaço	33
3.1.2	Dataset poças de garimpo	33
3.2	Premissas	34
3.3	Proposta de solução	34
3.4	Ambiente e ferramentas	35
3.4.1	<i>Ambiente</i>	35
3.4.2	<i>Bibliotecas</i>	35
3.4.3	<i>PyTorch</i>	35
3.5	Experimentos	36
3.6	Procedimentos	36

3.6.1	<i>Configuração de Ambiente</i>	37
3.6.2	<i>Análise de dados exploratória</i>	37
3.6.3	<i>Pré-processamento</i>	38
3.6.4	<i>Definição do modelo</i>	41
3.6.5	<i>Treino</i>	42
3.6.6	<i>Validação</i>	42
3.6.7	<i>Seleção de modelos</i>	42
3.6.8	<i>Treinamento ResNet</i>	44
3.6.9	<i>Treinamento Swin-T</i>	44
4	RESULTADOS	48
4.1	<i>Classificador Base Resultados Iniciais</i>	48
4.2	<i>Classificador Base Análise Para classes Escassas</i>	48
4.3	<i>Classificador Proposto</i>	49
4.4	<i>Comparação de resultados</i>	49
5	CONCLUSÃO	55
5.1	<i>Trabalhos futuros</i>	55
	REFERÊNCIAS	57
	ANEXOS	60

1 Introdução

A floresta amazônica sofre uma degradação histórica, com perda de até 19% em área de vegetação desde os anos 1970, atingindo uma perda de 10000km^2 em 2020 (AMIGO, 2020). Há previsões que ao atingir um limiar de 20% a 25% de perda do bioma, acarretará um ponto de inflexão, onde a floresta deixará de ser autossustentável e será substancialmente transformada, se tornando mais infértil e árida (LOVEJOY; NOBRE, 2018). Isto se deve ao fato do ecossistema possuir alta densidade de vegetação e taxa de evapotranspiração, formando os chamados rios aéreos, que sustentam as monções e alta taxa de precipitação (SATYAMURTY; COSTA; MANZI, 2013). Ao se remover a vegetação desse ciclo, o volume de precipitação pode decair abruptamente. Já existem evidências (AMIGO, 2020) de vegetação adaptada ao cerrado e climas mais áridos predominando na região leste da amazônia, território que sofreu grandes perdas de vegetação. Predições para tal ponto de inflexão apontam que até 2050 cerca de metade da floresta em território brasileiro não sobreviverá (LOVEJOY; NOBRE, 2018).

Tais evidências demonstram o quanto urgente são necessárias medidas para monitorar e coibir práticas ilegais de desflorestamento. Atualmente estas áreas podem ser detectadas remotamente pelo MapBiomass, o sistema de monitoramento de tempo real do Instituto Nacional de Pesquisa Espacial, o INPE. São realizadas múltiplas varreduras em amplo espectro, desde o micro-ondas, infravermelho ao espectro visível. Aplicando técnicas de fusão sensorial, é possível identificar as regiões onde tais práticas acontecem extensivamente (VALERIANO; NARVAES; MAIA, 2016). Dentre elas, queimadas, desmatamento extrativo, garimpo, agricultura e agropecuária irregulares. Contudo, soluções automatizadas atualmente implementadas possuem baixa resolução, de áreas de $250 \times 250\text{m}$. O que significa que explorações de menores escalas ou esparsas ainda podem ser difíceis de serem detectadas (MAPBIOMAS, 2021). Temos ainda que o INPE, conta atualmente com satélite CBERS-4 que possuem sensores na faixa do espectro visível e de maior resolução espacial. Isso permite realizar a detecção de ocupações irregulares ainda em fase iniciais, bem como identificar práticas que são menos extensivas em área, embora ainda muito prejudiciais. Como exemplo, temos o recente caso de explosões de inúmeras áreas de garimpo do Rio Madeira, em 2022, que passariam desapercebidas das detecções de piores resoluções espaciais (MAPBIOMAS, 2021).

O garimpo também é uma das principais causas da degradação do bioma, já que a Amazônia concentra 94% (mais de 100 mil hectares) da área garimpada brasileira, sendo mais de 50% potencialmente ilegais, por ocorrerem dentro em Terra Indígenas (TIs) e Unidades de Conservação (UCs). A área de garimpo no bioma cresceu 10x nas últimas três décadas, com 301% de expansão em UCs e 495% em TIs (MAPBIOMAS, 2021).



Figura 1 – Garimpo ilegal na Terra Indígena Munduruku, município de Jacareacanga.
Foto: Marizilda Cruppe/Amazônia Real

Uma possível solução para detecção de tais irregularidades são modelos de visão computacional e aprendizado profundo para classificação automatizada de sub regiões na faixa de captura dos satélites. Tais sistemas seriam treinados com amostras de regiões regulares de mata nativa e regiões onde ocorrem irregularidades, para conseguirem classificar e diferenciar cada categoria.

Uma das técnicas que se destacaram, na última década, pela atuação em visão computacional foram as redes neurais convolucionais. Conseguiram um salto de precisão ao resolver a competição de identificação de imagens ImageNet, por modelos como AlexNet e ResNet ([ALOM et al., 2018](#)). Contudo, para aplicações de imagens de satélite, também chamada sensoriamento remoto, podem não possuir o mesmo desempenho em relação a objetos do cotidiano ([WANG et al., 2022](#)). Isso se deve a alta variabilidade entre amostras dentro de uma classe a ser identificada e similaridades com amostras fora da classe.

Outra adicional dificuldade enfrentada por redes convolucionais em sensoriamento remoto é o grande volume de dados necessários para treinar o modelo, sob diferentes condições, como imagens rotacionadas, diferentes sensores, luminosidade no momento da captura, condições climáticas e nuvens. Tais variabilidades limitam a robustez e demandam um conjunto de treino que represente estatisticamente as possíveis diferentes condições ([SEDAGHAT; MOKHTARZADE; EBADI, 2011](#)).

Já uma recente família de modelos inicialmente publicada em ([DOSOVITSKIY](#)

et al., 2020), chamados Transformers Visuais, tem ganhado espaço no campo de visão computacional nos últimos anos. Foram capazes de superar o desempenho de modelos baseados em CNN, em muitas aplicações, utilizando menos pesos e sendo mais barato computacionalmente (WANG et al., 2022). Isto pode ser atribuído à chamada propriedade “atenção”, onde o contexto de cada parte de entrada, em relação às demais partes, tem peso para classificação (DOSOVITSKIY et al., 2020). Dessa forma, tais classificadores têm potencial de terem maior robustez em relação às variações de entradas.

Dado todos esses fatores, este trabalho terá como objetivo: comparar arquiteturas estabelecidas em relação a novas para o problema de classificar regiões irregulares em sensoriamento remoto. Mais especificamente utilizando modelos de redes convolucionais residuais (ResNets) e Transformers Visuais pré-treinados com imagens de satélite de diferentes condições. Contemplará as limitações de quantidade reduzida de amostras de treino e desbalanceada para a quantidade de classes de interesse. Dessa forma, medindo a capacidade de generalização e viés indutivo de tais arquiteturas para o campo de sensoriamento remoto. Tem também como objetivo secundário, disponibilizar o experimento e metodologia facilmente reproduzíveis para futuros trabalhos em aprendizado profundo com Transformers visuais.

O trabalho consiste em mais quatro capítulos, respectivamente: Revisão Bibliográfica, Metodologia para a implementação de solução, Análise De Resultados e a Conclusão, incluindo contribuições e direções futuras para este problema estudado.

2 Revisão Bibliográfica

Neste capítulo, será apresentado inicialmente definições do domínio do problema de classificação e identificação de cenas em sensoriamento remoto na seção 2.1, bem como suas características e contexto. Seguindo por um apanhado teórico da seção envolvendo aprendizado de máquina, redes neurais, redes neurais profundas, convolucionais e *transformers*, no que concerne interseções com possíveis soluções para o problema. E por fim, na seção 2.4 uma revisão da literatura e do atual estado da arte sobre o problema de classificação visual aplicadas ao sensoriamento remoto. Apresentando trabalhos que envolveram soluções tanto para o campo, quanto para implementações de redes neurais profundas.

2.1 Domínio do problema

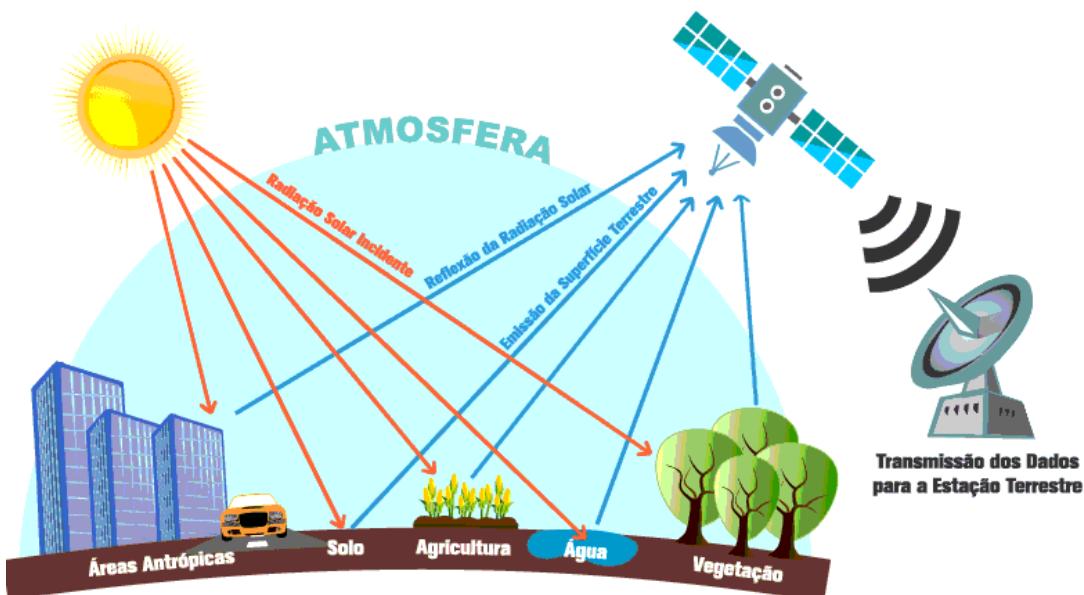


Figura 2 – Sensoriamento remoto Foto:([INSTRUTORGIS, 2022](#))

O problema de identificação de regiões com garimpo em imagens de satélite, pertence ao campo de estudos de sensoriamento remoto. Este consiste em técnicas de análises e medições de uma região geográfica terrestre, ou atmosférica. Podem ser realizadas por imagens aéreas ou por satélites, podendo abranger diferentes partes do espectro eletromagnético. Frequentemente consistem em métodos de aquisição ou processamento de sinais e imagens, para obter características ou reconhecer padrões em tais localidades ([EMERY; CAMPS; RODRIGUEZ-CASSOLA, 2017](#)). O termo foi cunhado referir-se a medição realizada por algum meio indireto ou “remoto”, em vez de um contato direto com sensores no ambiente medido ([EMERY; CAMPS; RODRIGUEZ-CASSOLA, 2017](#)).

Ainda mais especificamente, o problema introduzido também faz parte do campo de reconhecimento de padrões em imagens. Temos que a área varrida por sensoriamento remoto é extensiva para ser vasculhada manualmente. Portanto, se faz necessário o uso de algorítimos que automatizem a detecção ou classificação do objeto a ser encontrado. O campo de reconhecimento de padrões possui algorítimos tradicionais para identificar e localizar objetos de interesse, porém podem ser muito caras computacionalmente e pouco efetivas, dependendo das características do problema. Por isso, tem recentemente sido frequentemente empregadas técnicas baseadas em aprendizado de máquina e aprendizado profundo, que atualmente são o estado da arte em muitos problemas de reconhecimento de padrões.

2.2 Revisão Teórica

2.2.1 Aprendizado de máquina

Aprendizado de máquina é um campo que estuda algorítimos capazes de aprender e realizar inferências a partir de dados. O termo foi cunhado em 1959, por Artur Samuel, como “Campo de estudos que visa a dar computadores a habilidade de aprender sem serem explicitamente programados para determinada tarefa.” ([SAMUEL, 1959](#)).

Já em ([MITCHELL, 1997](#)) define um algoritmo de aprendizado como “Um algoritmo dito conseguir uma experiência E com respeito a determinada classe de tarefas T e com medidas de desempenho P, se seu desempenho nas tarefas em T, medidas por P, melhoram a partir da experiência E.”

Para melhor entender cada constituinte dessa definição, podemos utilizar de um exemplo. A tarefa T temos como exemplo o problema de classificação, que consiste do algorítimo responder quais das k categorias para o qual ele experimentou em E, certas amostras de entradas pertencem. Para resolver tal tarefa, tal algorítimo de aprendizado deve produzir uma função $f : \mathbb{R}^n \rightarrow \{1, \dots, k\}$. Quando $y = f(x)$, o modelo atribui uma entrada descrita pelo por x , que nosso caso constitui uma amostra de entrada, a uma categoria identificada por um código numérico y . Uma variante do mesmo problema é em vez de classificar qual classe, retornar a distribuição de probabilidade sobre as classes ([GOODFELLOW; BENGIO; COURVILLE, 2016](#)). A medida de desempenho P, necessária para avaliar as habilidades de um dado algorítimo. Tal medida de desempenho é geralmente atrelada ao tipo de tarefa sendo realizada pelo sistema.

2.2.2 Métricas de desempenho

Para tarefas de classificação, é comumente utilizada a métrica de acurácia do modelo. Consiste na proporção de amostras na qual o modelo produz a saída correta. Sejam

verdadeiros positivos vp , falsos positivos fp , verdadeiros negativos vn , falsos negativos fn , temos que a acurácia AC é:

$$AC = \frac{tp + tn}{tp + tn + fp + fn} \quad (2.1)$$

- Verdadeiros Positivos: classificação correta da classe Positivo
- Falsos Positivos (Erro Tipo I): erro onde o modelo previu a classe Positivo quando o rótulo era classe Negativo
- Falsos Negativos (Erro Tipo II): erro onde o modelo previu a classe Negativo quando o rótulo era classe Positivo
- Verdadeiros Negativos: classificação correta da classe Negativo

Cada uma dessas métricas são representadas na chamada matriz confusão, como ilustra a figura 2.2.2.

Para problemas de multiclasses e multirótulos, ainda temos a métrica acurácia até o n -ésimo candidato, denotado por $AC@n$. Por exemplo: $AC@5$ é acurácia do verdadeiro positivo estar dentre os primeiros 5 candidatos sugeridos.

Outras métricas também podem ser interessantes em casos onde amostras das classes são muito desbalanceadas, métricas F_β e curva precisão-revocação. Sejam Precisão P e revocação R , temos que:

Altas taxas de precisão e revocação mostram que o classificador está retornando resultados relevantes, com poucos falsos positivos (Precisão) e também retornando a maioria de todos os resultados positivos (Revocação) (HASTIE; TIBSHIRANI; FRIEDMAN, 2001). Um sistema com alta revocação e baixa precisão retorna muitos resultados, mas muitos deles incorretos, enquanto um com alta precisão e baixa revocação retorna na maioria corretos, mas poucos resultados.

A métrica F_β é uma média harmônica entre precisão e revocação, portanto sumariza o compromisso entre esses dois índices e pesa a importância de cada a partir de β .

Podemos comparar métricas e modelos a partir do chamado classificador ingênuo, que atribui a probabilidade de uma classe ser positiva apenas pela frequência de positivos de tal classe no conjunto de dados:

$$ProbClassIng(X|y=1) = \frac{P}{P+N} \quad (2.6)$$

Em um conjunto de dados muito desbalanceado, métrica de acurácia é pouco capaz de refletir o quanto a desempenho de um modelo é superior ao classificador ingênuo, pois

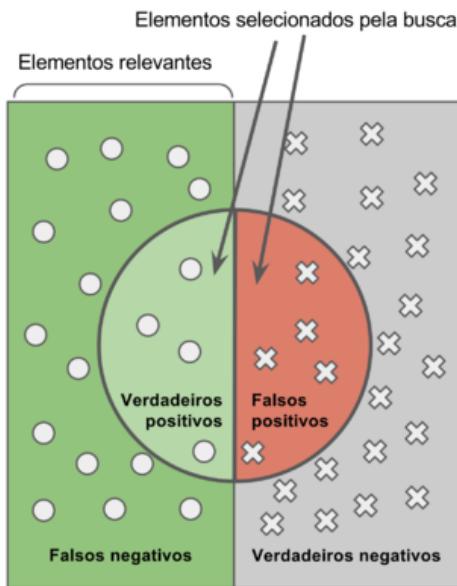


Figura 3 – Ilustração das métricas de precisão e revocação.
Fonte: WikiCommons

possuem acuráncias próximas. Enquanto Precisão-Revocação e métrica F_2 conseguem melhor diferenciar do classificador ingênuo e é capaz de avaliar o índice de revocação.

Um classificador que retorna a probabilidade de pertencimento de cada classe pode classificar tal classe como positiva ou não ao se comparar com determinado limiar.

A curva precisão-revocação ilustra a relação de compromisso entre os dois índices para diferentes limiares. Uma área grande abaixo da curva representa ambos alta precisão e revocação para muitos limiares diferentes. Do contrário, uma área grande abaixo da curva ¹ representa que independente do limiar escolhido para o classificador, apresentará baixa precisão e revocação.

Outra métrica para avaliação de classificadores é a curva COR ², contudo apresenta resultados otimistas para conjuntos de dados balanceados, assim como a acurácia.

Para ambas curvas PR e ROC, um modelo é qualitativamente bom quanto mais

¹ Area under the curve - AUC

² Comumente chamada de ROC - Receive Operator Characteristic

		Detectada	
		Sim	Não
Real	Sim	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

$$P = \frac{vp}{vp + vn} \quad (2.2)$$

$$R = \frac{vp}{vp + fp} \quad (2.3)$$

$$F_{\beta} = (1 + \beta^2) \times \frac{P \times R}{\beta^2 P + R} \quad (2.4)$$

$$F_2 = 5 \times \frac{P \times R}{5P + R} \quad (2.5)$$

Figura 4 – Matriz Confusão, Equações de Precisão, revocação, F_{β} e F_2

desempenho tiver em comparação com o classificador ingênuo, que escolhe classifica com a probabilidade da frequência daquela classe ou aleatoriamente.

2.2.3 Função de perda

Como métricas de perda para tarefas de classificação, podem ser utilizadas, por exemplo, a entropia cruzada, dada por:

Otimizadores possuem como objetivo ajustar pesos de uma função de custo para minimizar ou maximizar tal custo. Para quantificar o custo são utilizadas métricas de perda. Dentre elas, as funções de custos ideais para classificação binária multiclasses e multirótulos temos a entropia binária cruzada, definida por

$$ECB(y_c) = - \sum_{c=0}^M y_c \times \log(\hat{y}_c)$$

sendo $p(\hat{y}_c)$ a probabilidade.

Para conjuntos de dados muito desbalanceados, pode-se mitigar o sobre-ajuste da classe dominante utilizando BCE com pesos C_i correspondentes ao inverso da frequência das classes positivas, ou ao inverso da proporção de positivos de tal classe para o total de amostras negativas:

$$C_i = \frac{fn}{vp_i}$$

$$ECB(y_c) = - \sum_{c=0}^M C_i \times y_c \times \log(\hat{y}_c)$$

Outra solução para mitigar o mesmo problema é a função de perda focal. Consiste em uma *ECB* penalizando exponencialmente por um peso gama as classificações com probabilidades altas.([STEINER et al., 2021](#))

$$Focal(y_c) = - \sum_{c=0}^M (1 - w_i \times y_c)^\gamma \times \log(\hat{y}_c)$$

2.2.4 Redes Neurais Artificiais

O termo redes neurais embarca uma grande classe de modelos e métodos de aprendizados. O modelo mais simples, também podendo ser chamado rede de camada oculta única de perceptrons. Já o perceptron é a célula de uma rede neural. Se trata de um modelo matemático análogo a um neurônio. Possui um vetor de entradas e sobre essas entradas é aplicada uma combinação linear utilizando os pesos sobre cada entrada. O resultado de tal operação passa por uma função de ativação que resulta numa saída binária de classificação do perceptron. Dessa forma, ao se otimizar os pesos e função de ativação a determinadas amostras de treino e suas respectivas saídas rotuladas, podemos criar um

classificador linear simples, caso seja um problema linearmente separável. Portanto, o processo de treinamento de uma rede neural se trata de otimizar os pesos dos neurônios. Uma etapa importante dos ajustes dos pesos é a retro-propagação de erro¹ a qual é um algoritmo que optimiza os pesos da camada oculta (HASTIE; TIBSHIRANI; FRIEDMAN, 2001).

Para classificações multiclasses e multirótulos, isto é, uma amostra pode pertencer a mais de uma classe simultaneamente, são utilizadas camadas de saída que implementam normalização para selecionar a probabilidade de cada classe. Temos como exemplo a camada de ativação *Sigmoid* 2.2.4. Pode ser interpretada como probabilidade, portanto $\sigma(z(x))$ é a probabilidade de X pertencer à classe positiva e $1 - \sigma(z(x))$ a de não pertencer à classe positiva.

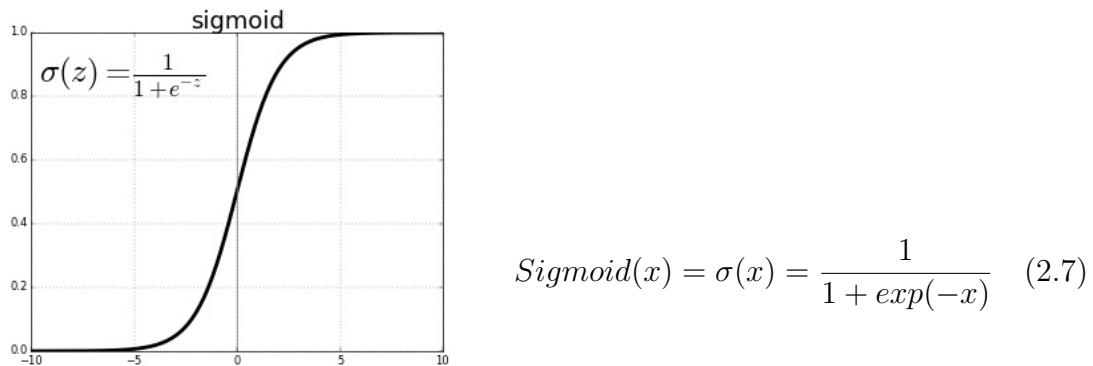


Figura 5 – Função de ativação *Sigmoid*. Fonte: WikiCommons

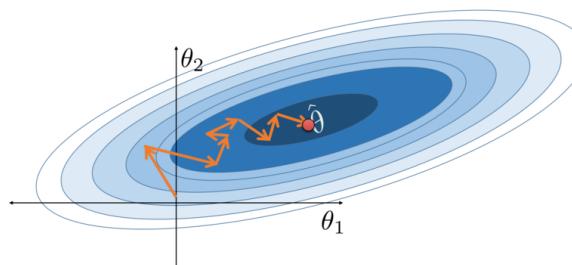


Figura 6 – Exemplo de algoritmo de otimização: gradiente descendente estocástico.

2.2.5 Redes Neurais Artificiais Profundas

O termo redes neurais profundas e o aprendizado profundo, ou comumente chamado de *deep learning*, se refere a redes neurais artificiais com múltiplas camadas ocultas. Foram uma das tecnologias maior desenvolvidas nos últimos anos, e se tornaram cada vez mais populares. Devido a sua superior *performance* em extração de características, teve sucesso

¹ Frequentemente citada como *Backpropagation*

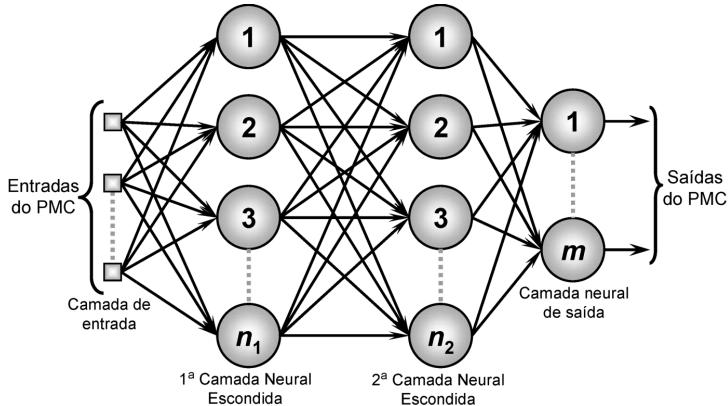


Figura 7 – Exemplo de uma rede de perceptrons de multiphas camadas - MLP.

por distintos domínios, como visão computacional, reconhecimento de fala, processamento natural de linguagem e em big data.

Um dos riscos envolvendo o treinamento de redes neurais profundas é o problema de *overfitting*². Se trata de quando o modelo é treinado e para gerar uma função próxima demais aos dados de treino, e perdem generalidade, falhando em previsões em dados fora do conjunto de treino. Para mitigar o surgimento de *overfitting* durante o treino, são utilizadas técnicas de regularização. Consistem em adicionar penalidade à complexidade do modelo, de forma que o treino otimize para se tornar uma função genérica. Dentre as técnicas de regularização possíveis de redes neurais profundas, podemos citar o *Dropout*, *Drop-connect* e *pruning*, que consistem em respectivamente a remoção, adição de conexão entre neurônios e remoção de neurônios (HASTIE; TIBSHIRANI; FRIEDMAN, 2001).

2.2.6 Redes Neurais Convolucionais

Uma arquitetura clássica é a rede neural convolucional (CNN³, ou Redes Neurais Convolucionais), que utiliza convoluções para extraer características de uma imagem entre cada camada de filtros. Também possui camadas de *pooling*, não lineares e camadas completamente conectadas (ALBAWI; MOHAMMED; AL-ZAWI, 2017). Uma das pressupostos das CNNs é os filtros serem indiferentes a translações das características na imagem, possibilitando assim uma eficiente extração de características para composição e identificação da imagem.

Em 2012, o modelo de CNN AlexNet ultrapassou em desempenho os modelos do estado da arte em uma grande margem. Dois fatores promoveram tal degrau de evolução: a disponibilidade de grandes *datasets*⁴ como o ImageNet e a comoditização de placas gráficas, que promoveram significativamente mais poder computacional para treino, já que

² Do inglês, traduzido como Sobre-ajuste.

³ Do inglês *Convolutional Neural Network*

⁴ Conjunto de dados

são aceleradores de operações vetorizadas e sobre matrizes. Dessa forma, desde 2012 CNNs se tornaram o modelo padrão para tarefas de visão computacional (ALOM et al., 2018).

A maior vantagem dos modelos CNN em comparação com os métodos anteriores era de conseguir ser treinados ponta a ponta, sem a necessidade de criação de filtros ou a criação manual de extratores de características visuais. Também possuem duas importantes propriedades como invariância translacional, isto é, o sistema produz a mesma resposta independente de translação; e campo receptivo restrito, o que significa que neurônios das primeiras camadas capturam detalhes finos e locais.

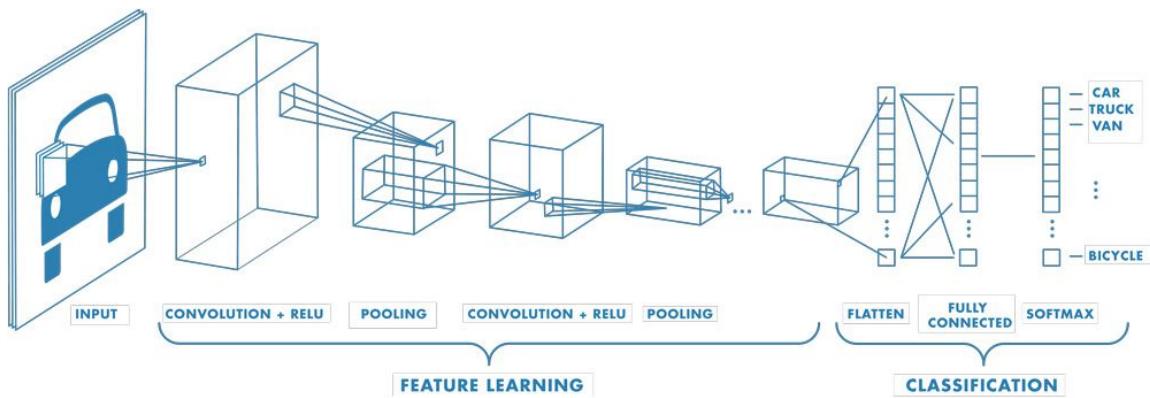


Figura 8 – Arquitetura de uma rede convolucional. Filtros extratores de características são aplicados em diferentes resoluções e campos visuais. A saída de cada imagem convoluta alimenta a próxima camada. As últimas camadas completamente conectadas realizam a classificação.

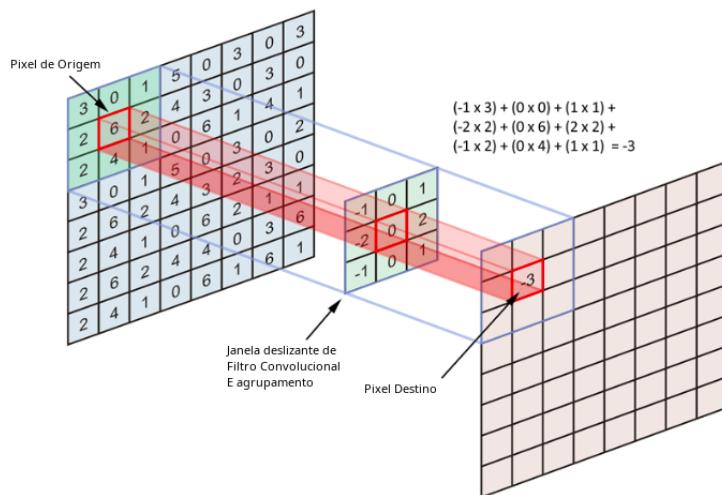


Figura 9 – Filtro convolucional que aplica uma janela deslizante aplicando operação de convolução e pooling.

2.2.7 ResNet

Após a primeira arquitetura AlexNet, a arquitetura subsequentes utilizaram mais camadas em uma rede neural profunda para gerar melhores modelos. Contudo, sofreram de um problema de convergência de otimização e regularização, chamado explosão de gradiente. Consiste no fenômeno da tendência de surgimento de gradientes muito altos ou se tornam zero, fazendo com que o treinamento não tenha convergência, para redes com número elevado de camadas. Para solucionar este problema, foi proposta a arquitetura de redes residuais, que possuem conexões saltos, conectando camadas de diferentes profundidades, como ilustra 19. Possibilitando a propagação de gradiente saltando camadas. Assim mitigou o problema de explosão de gradientes e permitiu, desde então, a escalabilidade de redes neurais profundas.

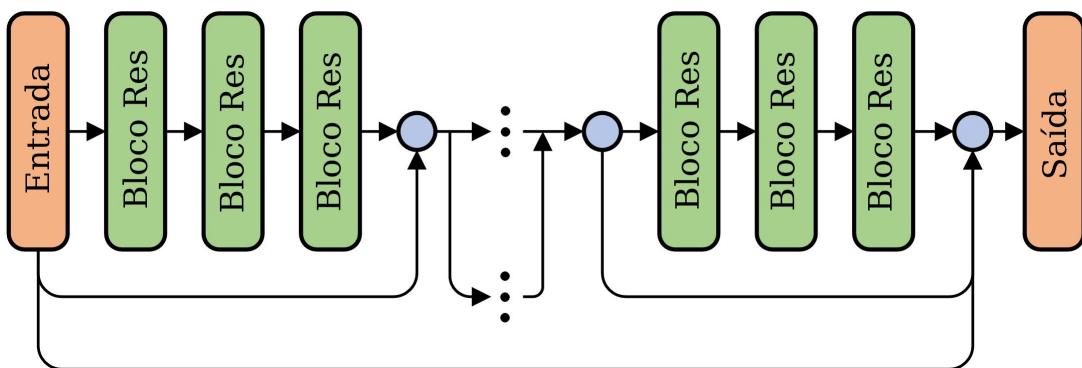


Figura 10 – Arquitetura de modelos ResNet e conexões saltos. Fonte:WikiCommons

2.2.8 O problema de rotulagem e variabilidade de amostras de treino

Um dos principais desafios envolvido o treino de *CNNs* aplicado a sensoriamento remoto é representar um estado de características que cubram as variações fotográficas, tanto em características do sensor, como variações da imagem no dia, clima, estação e plataforma da câmera, o que se torna uma tarefa difícil. Para uma localização efetiva, o modelo deve ser robusto a todas essas variações, que requer um grande conjunto de treino que cobre boa parte das diversas condições possíveis. Tal conjunto de dados não é disponível e nem viável de obter, pois se trata um volume muito grande de amostras. Tais limitações levam a necessitar o desenvolvimento de algorítimos que aprendem seletivamente para que o poder computacional seja utilizado eficientemente, bem como reutilizar conhecimento prévio e evitar treinamento redundante (ROSTAMI, 2019). Dentre as técnicas utilizadas para implementar esses modelos mais eficientes, temos como exemplo o aprendizado supervisionado fraco, e a transferência de aprendizado.

2.2.9 Aprendizado semi-supervisionado

As técnicas de aprendizado semi-supervisionado consistem em treinar um modelo com apenas um conjunto reduzido de amostras rotuladas de treino, e as demais amostras serem não supervisionadas. As demais amostras de treino podem ser utilizadas, por exemplo, agrupando-as com as amostras rotuladas e classificando-as como a amostra mais próxima, como apresenta o trabalho de (SANCHES, 2003). Outras propostas envolvem *data augmentation*⁵, que consistem em gerar um conjunto de treino maior, dados as amostras de treino disponíveis.

2.2.10 Transferência de aprendizado

Já técnicas de transferência de aprendizado⁶, ou *few shots learning*, consistem em redes treinadas para um conjunto limitado de testes (ROSTAMI, 2019) e reutilizam esse conhecimento através de diferentes domínios, tarefas ou agentes. Consistem primariamente de um problema origem e um problema objetivo e como podemos suceder em transferir conhecimento dado o problema origem. A abordagem de transferência de conhecimento se inspira em replicar a habilidade humana em que é possível transferir conhecimento de experiências passadas para lidar com tarefas com poucas amostras rotuladas. Este fato inspirou em representar dados de diferentes problemas de aprendizado de máquina em um espaço embutido onde as representações utilizam de diversas relações entre diferentes domínios de conhecimentos e tarefas. Uma implementação encontrada na literatura, proposta em (ROSTAMI, 2019), consistiu de transplantar a camada de características de uma CNN derivada do domínio de origem para inicializar outra rede, do domínio objetivo, composta por uma camada final fortemente conectada. Assim foi aproveitada as primeiras camadas e a rede foi trenada para o domínio objetivo com uma quantidade menor de amostras.

2.3 Transformer Visual

Temos que a arquitetura de CNNs em si, é projetada especificamente para imagens e podem ser computacionalmente onerosas, dessa forma podendo não escalar ou generalizar o suficiente para modelos multimodais. Diante dessas limitações, foram pesquisados modelos que possam escalar e serem agnósticos em quesito de domínios de aplicação. Nessa direção que em (DOSOVITSKIY et al., 2020) é criado o *Transformer* Visual ou ViT⁷, é modelo de visão baseado em *encoders Transformer*, originalmente projetada para tarefas baseadas em texto.

⁵ Aumento de dados

⁶ Comumente citado como Transfer Learning

⁷ Vision Transformer

O ViT representa uma imagem de entrada como uma sequência de recortes de imagem, similarmente a uma sequência de representações de palavras quando aplicado Transformers para texto, e diretamente predizendo a classe de rótulo da imagem. Tal arquitetura e suas variações apresentaram excelente desempenho, superando a de CNNs até então estado-da-arte, a ResNet, consumindo até um quarto de recursos computacionais destas.

O *transformer* original recebe como entrada uma sequência de palavras, que então usa para classificação, tradução ou outras tarefas de processamento natural de linguagem. Para o ViT, em (DOSOVITSKIY et al., 2020) foram aplicadas poucas modificações, para mesma arquitetura ser aplicada a imagens em vez de palavras, e assim observar como o modelo aprende por si próprio.

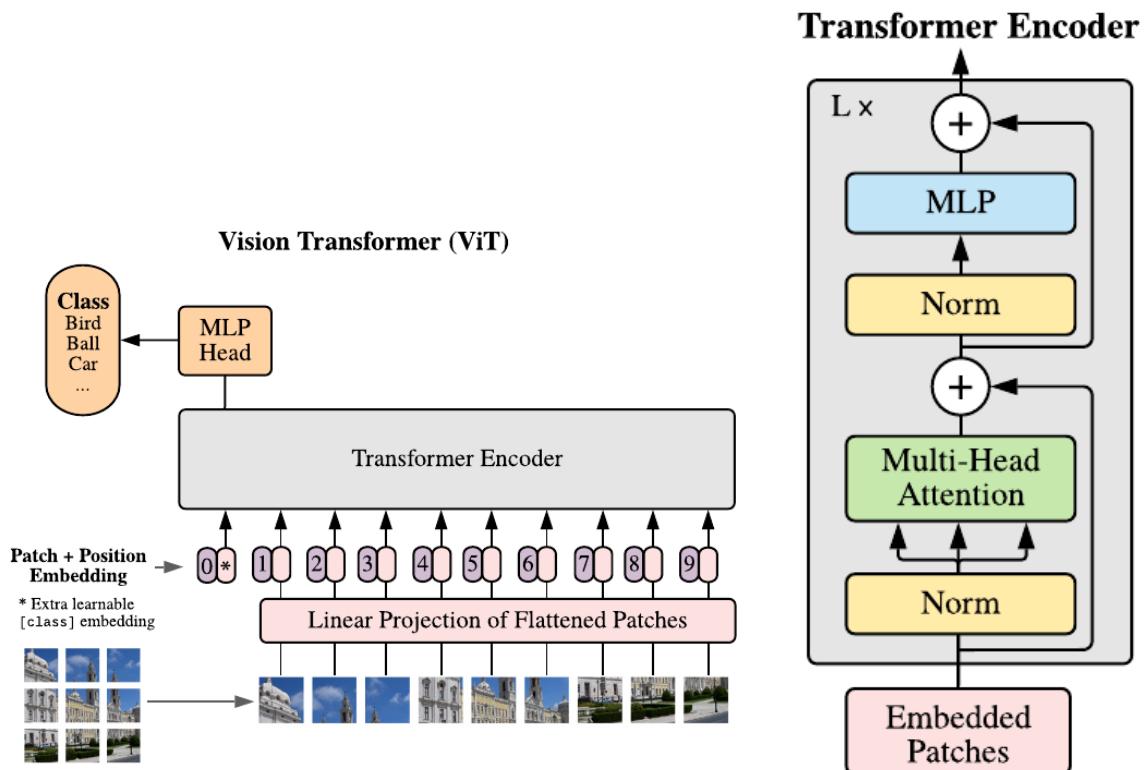


Figura 11 – O ViT divide uma imagem em uma grade de recortes quadrados, cada fragmento é achatado em um vetor único contendo todos os canais de todos os pixels, e projetando-os em uma dimensão de entrada desejada, alimentando a camada de múltiplos Encoders em paralelo. (DOSOVITSKIY et al., 2020)

2.3.1 Arquitetura do transformer visual

O *encoder do transformer* é composto por:

- Camada de múltiplas cabeças de atenção (MSP): Esta camada concatena todas as

saídas de atenção linearmente para a dimensão correta. As várias cabeças de atenção ajudam a treinar local e globalmente dependências em uma imagem.

- Múltiplas (MLP) camadas de perceptrons: contém um MLP de duas camadas com função de ativação GELU (Unidade de erro linear Gaussiano)
- Camada de normalização (LN): Adicionada antes de cada bloco e normaliza os pesos em uma mesma camada.
- Saltos de conexão: somam a saída de blocos a entrada. Desempenha papel de auxiliar a convergência da otimização do gradiente descendente e explosão de gradientes.

A MSP é o núcleo do *encoder*⁸. E apresenta o mecanismo de autoatenção. Consiste em relacionar diferentes posições de uma única sequência para se computar uma representação desta sequência (VASWANI et al., 2017).

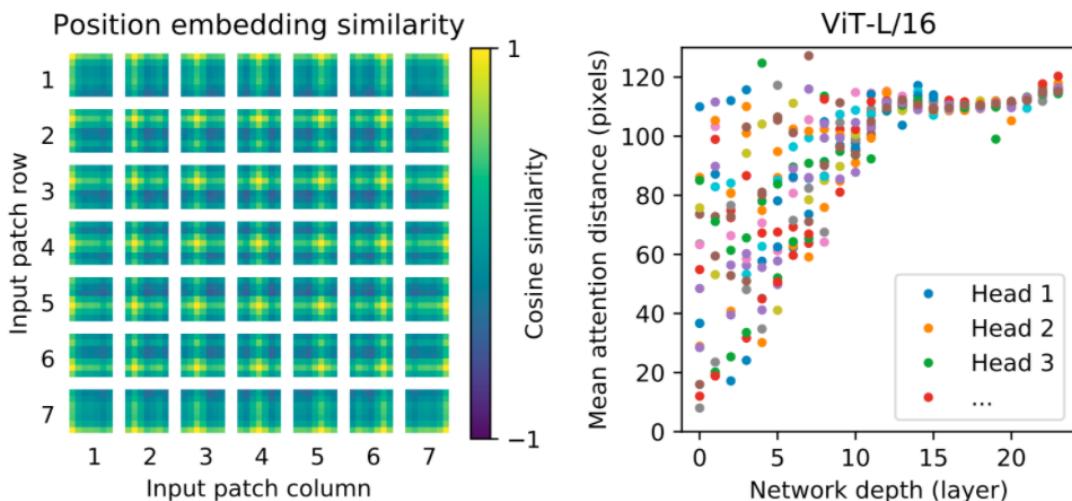


Figura 12 – Esquerda: ViT aprende a estrutura de grade via representações de posição. Direita: Camadas inferiores do ViT contêm ambas características locais e globais, o quanto mais profundas as camadas, mais globais as características.(DOSOVITSKIY et al., 2020)

É possível observar como o modelo aprende ao visualizar alguns comportamentos internos. Ao observar as representações de posição, os parâmetros do modelo que aprendem a codificar a representação relativa dos recortes, encontramos que o ViT consegue reproduzir intuitivamente uma estrutura de imagem. Cada representação de posição é mais similar a outras da mesma coluna e linha, indicando que o modelo recuperou a estrutura de grade das imagens originais. Em camadas mais profundas, apenas características globais são utilizadas, ou seja, grandes distâncias de atenção. Enquanto camadas mais rasas capturam ambas características globais e locais, indicado por uma faixa grande de atenção. Em

⁸ Codificador

contraste, apenas características locais são presentes nas camadas superficiais das CNNs. Estes experimentos indicam que ViT aprendem características codificadas manualmente nas CNNs, como percepção da estrutura de grade. E principalmente revela que tais modelos são livres para aprender padrões mais genéricos, como mistura de características globais e locais, dessa forma contribuindo para generalização (DOSOVITSKIY et al., 2020).

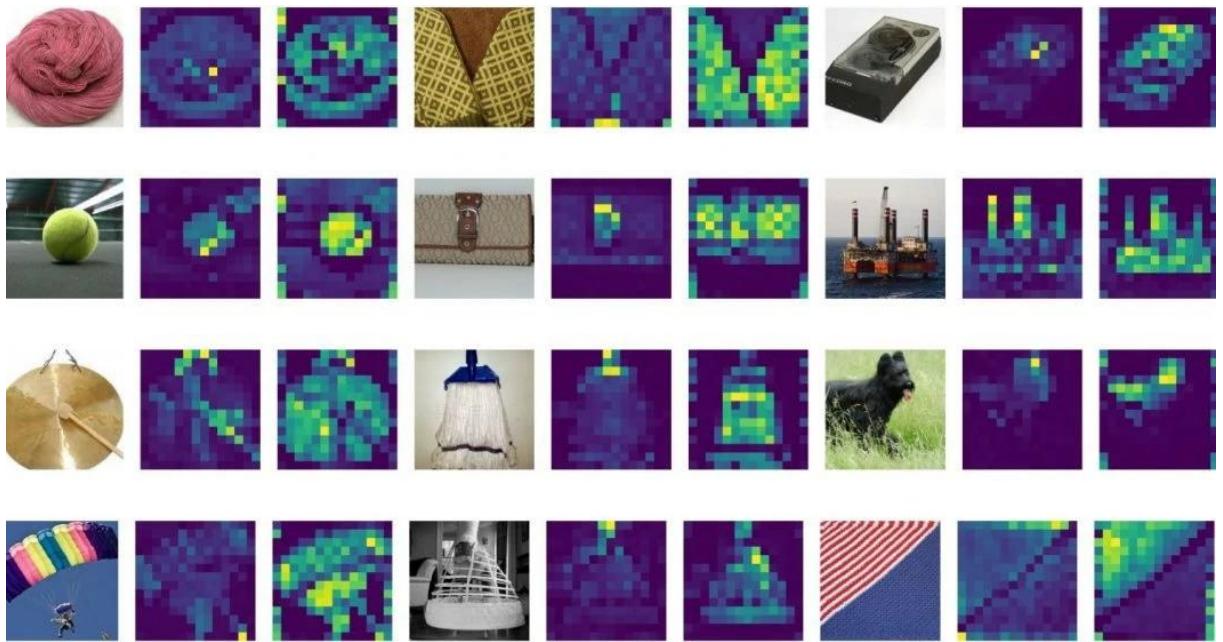


Figura 13 – O modelo lida com as regiões que são semanticamente relevantes para classificação. (CHEN; HSIEH; GONG, 2021)

2.4 Trabalhos anteriores

2.4.1 Problema de detecção de mudanças em minas de menor escala

Em (CAMALAN et al., 2022) foi gerado um *dataset*⁹ para detecção de mudanças e de mineração de ouro em menor escala. Com os dados rotulados, foram testadas abordagens supervisionadas e semi-supervisionadas e extratores de características para encontrar um melhor modelo detector de mudanças. Obtiveram o melhor modelo baseado no método E-ReCNN, utilizando seis canais e treino supervisionado, com pontuação f1 de 0.88 ± 0.05 .

2.4.2 Classificação de minas e represas

O problema de classificação para detecção em sensoriamento remoto, utilizando aprendizado profundo, foi explorado em (BALANIUK; ISUPOVA; REECE, 2020), utilizando imagens multiespectrais do satélite *Sentinel-2*. Consistiu em treinar um modelo baseado em CNN para identificar em recortes da imagem qual classe pertence.

⁹ Conjunto de dados

2.4.3 Pré-treino de *transformers* visuais para sensoriamento remoto

Para a aplicação de sensoriamento remoto temos trabalhos (WANG et al., 2022) que demonstram a aplicação de transferência de aprendizado e redes pré-treinadas utilizando ViT estado-da-arte. Utilizou-se o *dataset* MillionAID, o maior *dataset* datado até agora para sensoriamento remoto. Contem mais de um milhão de imagens sem sobreposição, com múltiplas visões temporais para a mesma cena, de canais apenas RGB. Possui uma árvore de classificação com 51 folhas, de cenas de terras de: agricultura, comercial, industrial, serviço público, industrial, transporte, regiões com água e regiões inutilizadas. Cada Folha possui 2.000 45,000 imagens. Foram obtidas pelo *software* *Google Earth*, que possui uma diversidade de sensores, resultando em diferentes resoluções, desde 50 cm à 150 m. E tamanhos de imagem de 10k pixeis até 900 mega pixeis. O autor pode concluir que a transferência de aprendizado foi eficaz para posteriores finetune¹⁰.

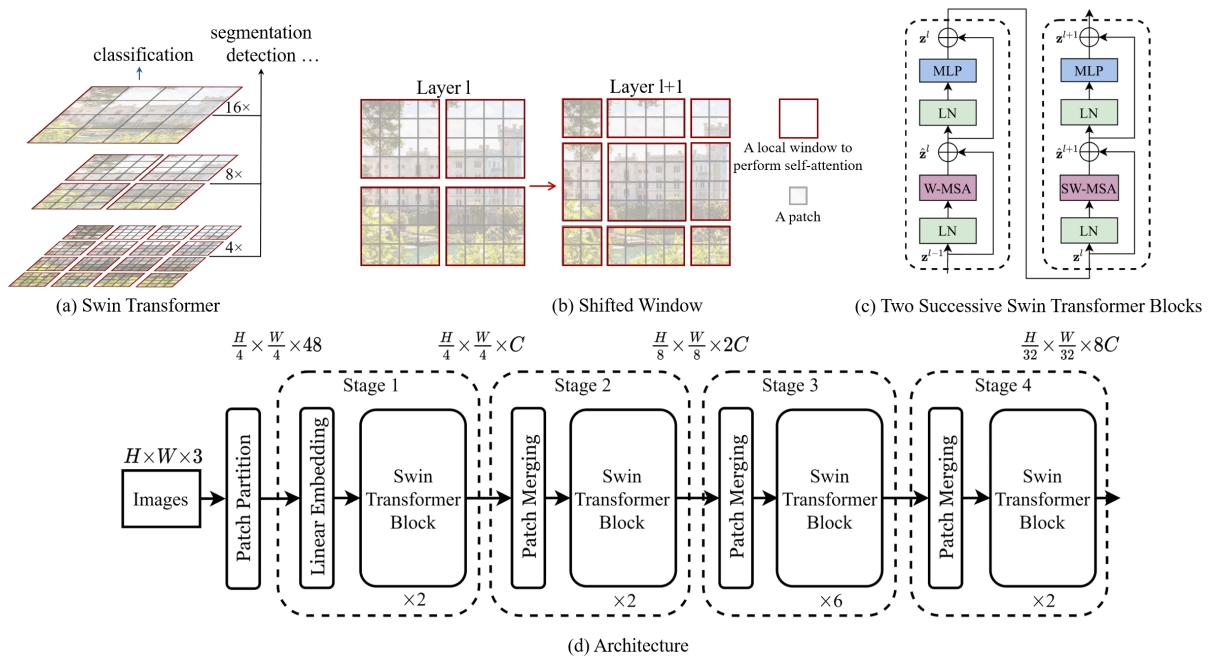
2.4.4 Transformer Swin

Em (LIU et al., 2022) foi proposto uma arquitetura de transformer visual com melhorias capazes de reduzir o custo computacional do cálculo de atenção entre cada recorte da imagem de entrada. Este trabalho aponta que a ordem de complexidade do ViT clássico é $O(n^2)$, sendo n o número de retalhos, já que é calculado a métrica de atenção entre cada retalho com todos os outros demais. Com isso, torna caro o treino para maiores resoluções ou granularidade de detalhes. A arquitetura Swin, ou *Shifted Window* se baseia em uma disposição hierárquica de Transformers, como ilustra a figura 14, sendo que nas camadas mais baixas, coletam atenção apenas localmente e não com todo restante da imagem. Dessa forma o custo se torna aproximadamente linear.

2.4.5 ForestViT Modelo de classificação de desmatamento

Em (KASELIMI et al., 2022) foi proposto um classificador multiclassas baseado na arquitetura de *transformers* visual. Os resultados também foram comparados com modelos baseados em CNN estabelecidos, como Resnet e VGG.

¹⁰ Ajuste fino.

Figura 14 – Arquitetura swin ([LIU et al., 2022](#))

3 Metodologia

Neste capítulo serão apresentados em detalhes as premissas do problema de classificação visual, bem como a metodologia para sua solução, que será particionada em seções conforme proposto a seguir: A seção 3.1 apresenta o conjunto de dados utilizado para o experimento. A seção 3.2 apresenta as limitações e condições de contorno do problema. A seção 3.3 apresenta a proposta de solução para o problema. Ambiente e ferramentas em 3.4. Um sumário dos experimentos é apresentado em 3.5. Por fim, a seção 3.6 apresenta os procedimentos do experimento para treino e teste do modelo. Dessa forma, a metodologia será seccionada nas seguintes partes:

- Datasets.
- Premissas do problema.
- Proposta de solução.
- Experimento para a solução do problema.

3.1 *Conjunto de dados*

Foram utilizados dois conjuntos de dados, escolhidos por razões de disponibilidade e representatividade do problema.

3.1.1 Dataset Amazônia do Espaço

Este conjunto de dados é de imagens coletadas dos satélites Planet Flock entre 2016 e 2017. Todas as imagens são da bacia amazônica. Este data set concerne ao desmatamento e cobrindo condições atmosféricas, coberturas de terreno e fenômenos raros. Cada amostra é um recorte de 256×256 pixels RGB, pertencente a uma ou mais das 17 classes distintas e possui resoluções espaciais variáveis, por exemplo, de 3m por pixel. Este data set foi publicado pela empresa Planet¹ na plataforma Kaggle, de concursos de aprendizado de máquina. Também são disponibilizadas o mesmo conjunto de dados com o canal de infravermelho-próximo e sem compressão.

3.1.2 Dataset poças de garimpo

Este *dataset* foi utilizado em (CAMALAN et al., 2022), mencionado no capítulo anterior. Concerne a tarefa de identificação de mudanças de imagem. Aplicadas a identificação

¹ <https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/data>



Figura 15 – Amostras de classes do dataset Amazônia do Espaço Fonte:([PLANETCO, 2016](#))

de garimpo artesanal de ouro de pequena escala, pode ser desafiador de se identificar, dado a variabilidade de condições atmosféricas e baixa resolução. Foram utilizadas imagens de Madre de Dios, região do Peru. Bem como amostras de outros países: Venezuela, Indonésia e Myanmar.

3.2 *Premissas*

Temos que o problema de identificação em sensoriamento remoto impõe a dificuldade de alta similaridade extra-classes e divergências intra-classes, da qual surge uma dificuldade de generalização e de viés indutivo para identificação de amostras fora das classes treinadas. Temos ainda que o treinamento completo de tais modelos geralmente requer um volume substancial de amostras (>1 milhão de amostras) rotuladas e de recursos computacionais.

3.3 *Proposta de solução*

Como solução para o problema, foi proposta a utilização de um modelo de *Transformer visual* pré-treinado em um conjunto de dados expressivo e realizar o *fine-tune*² para o conjunto de dados de interesse. Dessa forma, obtendo um modelo com boa capacidade de generalização e melhor viés indutivo, assim demandando menor quantidade de amostras para ser treinado.

Para solucionar o problema de generalização e conjunto de dados limitado da região de interesse, propomos a utilização de um modelo pré-treinado explicado em 2.2.10 um dataset extensivo de imagens aéreas e de satélites, para aproveitar seu extrator de características, conforme realizado em ([WANG et al., 2022](#)). Assim o modelo será re-treinado (*fine-tunning*³) para o conjunto de dados de interesse.

² Ajuste fino

³ Trad. do inglês: Ajuste fino.

Dessa forma, a arquitetura proposta é o Swin-T apresentado em (LIU et al., 2022) composta por camadas hierárquicas de Encoders⁴ Transformers que funcionam como extratores de características. As camadas seguintes são camadas conectadas totalmente seguidas por uma camada *sigmoide* que realiza a classificação.

3.4 Ambiente e ferramentas

O Ambiente dos experimentos será em cadernos *Jupyter*, para ser facilmente replicável e ser executável em nuvem, com a possibilidade de alugar recursos computacionais da plataforma em nuvem *Google Colab*. Também será utilizado o *framework PyTorch*, por razões de disponibilidade de métodos e conhecimentos do autor.

3.4.1 Ambiente

Foram dispostos 100h de orçamento de recursos computacionais, em uma instância da plataforma com GPU NVidia Tesla T4 de 12 GB de RAM. CPU Intel(R) Xeon(R) CPU @ 2.20GHz, 12 GB de memória. Para o treinamento de redes neurais profundas, aceleradores como placas gráficas podem resultar em ganhos de tempo de até 40x em comparação com treinamentos em CPU, bem como o uso da RAM para placa gráfica para realizar *caching*¹ obtendo melhores tempos de acesso durante carregamentos. Todos resultados de experimentos serão executados neste [Caderno Jupyter](#). Os experimentos foram armazenados no seguinte repositório [GitHub](#).

3.4.2 Bibliotecas

Para o experimento também contou-se com bibliotecas de cálculo numérico e algébrico como o Numpy. Também Pandas para manipulação de dados tabulares. Já a ferramenta SciKitLearn contém diversos módulos para aprendizado estatístico, como funções de métricas, de perdas. Para visualização dos resultados utilizado serão Matplotlib, Plotly e Seaborn.

3.4.3 PyTorch

Pytorch é um *framework*⁵ de processamento de tensores e acelerados por GPU³ ou TPUs⁶ para aprendizado de máquina profundo. É código-aberto, possuindo um front-end de fácil utilização em Python, implementado em linguagens C++ e CUDA para otimizações

⁴ Trad. do inglês: Codificador

¹ Caching é o processo de utilizar um espaço de memória para o armazenamento temporário e de acesso rápido de dados que possuem uma grande probabilidade de serem utilizados novamente

⁵ Ferramental

³ Unidade de Processamento Gráfico, Trad.: *Graphic Process Unit*

⁶ Unidade de Processamento de Tensores. Trad. de *Tensor Processing Unit*

de computações numéricas, matriciais e de diferenciação, extensivas para esta aplicação. O projeto é originalmente criado pelo *Facebook Research* e é atualmente amplamente adotado pela academia e mercado em aplicações de aprendizado profundo. Possui muitas implementações das ferramentas mais utilizadas, especificamente aplicadas a *Deep Learning* e visão computacional. Tem ampla adoção devido à intenção de ser um *framework* de fácil uso e alto nível, com muitas abstrações e técnicas já implementadas.

3.5 Experiments

Para a construção da solução desejada, experimentaremos combinar um modelo pré-treinado com camadas internas, correspondente ao extrator de características e de saída treinadas para a região de interesse. Consiste em fazer experimentos em uma complexidade crescente, e replicando resultados para garantir corretude. Será simplificado a reprodução para apenas um *dataset*.

Para obter o modelo proposto e de melhor desempenho, inicialmente definem-se os seguintes passos de experimentos para construir o modelo final:

1. Análise exploratória dos dados, visando a melhor conhecer o problema e amostras.
2. Disponibilizar pre-processamento do conjunto de dados e de carregadores de amostras.
3. Elaborar e treinar um modelo ResNet-18 pré-treinado, fazendo ajuste de hiperparâmetros a fim de obter o melhor valor de métrica no modelo base
4. Utilizar pesos do trabalho 2.4.3 com *checkpoints*⁷ disponibilizados, a fim de replicar a metodologia de fine-tune.
5. Realizar o mesmo para o modelo proposto baseado em transformers visuais
6. Fixar os hiperparâmetros e metodologia de treino para modelos base e proposto e treiná-los novamente a fim de realizar comparações
7. Realizar análise dos resultados de ambos e comparar desempenho em várias métricas relevantes ao problema
8. Comparar modelos e com trabalhos-base 2.4.5

3.6 Procedimentos

Os procedimentos do experimento principal consiste principalmente nas etapas de configuração do ambiente, pré-processamento do conjunto de dados, análise exploratória, definição do modelo e iterações de treino-validação. Explicadas adiante.

⁷ Captura dos pesos de uma rede a partir de certo ponto do processo de treino

3.6.1 Configuração de Ambiente

A plataforma GoogleColab consiste em uma instância alocada temporariamente com custo por tempo de uso, porém com desconexão por tempo ocioso. Durante o treinamento a interface se torna ociosa e ocorrem desconexões. Para mitigar isto foram criadas funções de gerenciar contexto do experimento, para implementar o armazenamento e carregamento se estatísticas do treinamento a cada época.

Os arquivos compactados de são obtidos de hospedagem em nuvem e descompactado na instância, seja em disco ou em partição na memória RAM. Esta segunda opção permitiu importantes melhorias de tempo de carregamento de batches¹. O arquivo contendo os pesos do modelo para criação do modelo são obtidos da mesma forma. Com cada instância é volátil, sendo possível armazenar conteúdos somente no serviço de hospedagem GoogleDrive, os resultados, pesos do modelo e contexto do experimento precisam ser salvos a cada execução, em caso de desconexão.

3.6.2 Análise de dados exploratória

Para melhor entendimento das características do conjunto de dados, se faz necessário uma exploração dos dados. Consiste em entender os diferentes tipos de amostras e rótulos, por meio de estatísticas de distribuição de classes e visualizações de agrupamento.

Pela distribuição de classes em 1, podemos observar que se trata de um dataset com acentuado desbalanço. Contem certos eventos raros como o chamado de Roça de ventos, que consiste em um evento climático gerador de ventos de mais de 160 km/h que devasta uma vasta área de floresta. Na figura 16 podemos visualizar amostras de cada tipo de classe. Cada amostra pode a mais de um tipo de classe.

A partir da matriz de coocorrência, podemos averiguar sobreposições de classes mais frequentes. É possível constatar que as amostras relacionadas a vegetação primária, agricultura e águas são as predominantes. Quanto ao clima, também há sobreposição de classes. Contudo, para as classes raras, são em sua maior parte independentes entre si.

Outra técnica de agrupamento e semelhança entre amostras é o t-SNE⁵, ou Representação de vizinhos estocástica T-Distribuida, que é um método de redução de dimensionalidade para dados de alta dimensionalidade e projeção em espaço de representação de baixa dimensão⁶. Assim é possível realizar agrupamentos de amostras próximas num espaço de representação. Na figura 18 podemos verificar grupos de amostras visualmente próximos que serão desafiadores de realizar uma classificação precisa. (MAATEN; HINTON, 2008)

¹ Lotes ou número de amostras processadas entre atualizações de modelo

⁵ T-distributed Stochastic Neighbor Embedding

⁶ Embedding

Tabela 1 – Proporção de Classes do conjunto de dados *Planet*

Classe	Rótulo	Amostras	Proporção (%)
Mina Convencional	conventional mine	100	0,247
Roça de Ventos	blow down	101	0,250
Queimada	slash burn	209	0,516
Florescimento	blooming	332	0,820
Garimpo	artisinal mine	339	0,837
Desmatamento Seletivo	selective logging	340	0,840
Área Descoberta	bare ground	862	2,129
Nublado	cloudy	2089	5,161
Névoa	haze	2697	6,663
Habitação	habitation	3660	9,042
Cultivação	cultivation	4547	11,233
Parcialmente Nublado	partly cloudy	7261	17,938
Águas	water	7411	18,308
Estrada	road	8071	19,939
Agricultura	agriculture	12315	30,423
Clima Limpo	clear	28431	70,236
Vegetação Primária	primary	37513	92,673

3.6.3 Pré-processamento

O pré-processamento consiste em preparar as amostras do conjunto de dados de interesse para treino e validação utilizando as bibliotecas mencionadas anteriormente. Para o conjunto de dados floresta amazônica temos 40479 amostras. Foram separadas em uma divisão de treino-validação na proporção 80%-20%. Cada recorte de resolução 256×256 px.

O pré-processamento em cada amostra é feito em tempo de execução por instâncias de transformação composta de:

1. Carregamento dos canais RGB e descartando o de infravermelho-próximo de cada imagem.
2. Redimensionamento usando interpolação linear de 256×256 px para 224×224 px.
Isto se deve ao fato dos modelos já terem sido pré-treinados e configurados para essa dimensão de entrada.
3. Conversão da imagem para estrutura de dados numérica de Tensor
4. Aplicar espelho vertical ou horizontal, cada um com probabilidade de 25%
5. Normalizar cada canal de cor RGB usando normalização Gaussiana com médias e desvio padrão do dataset ImageNet.

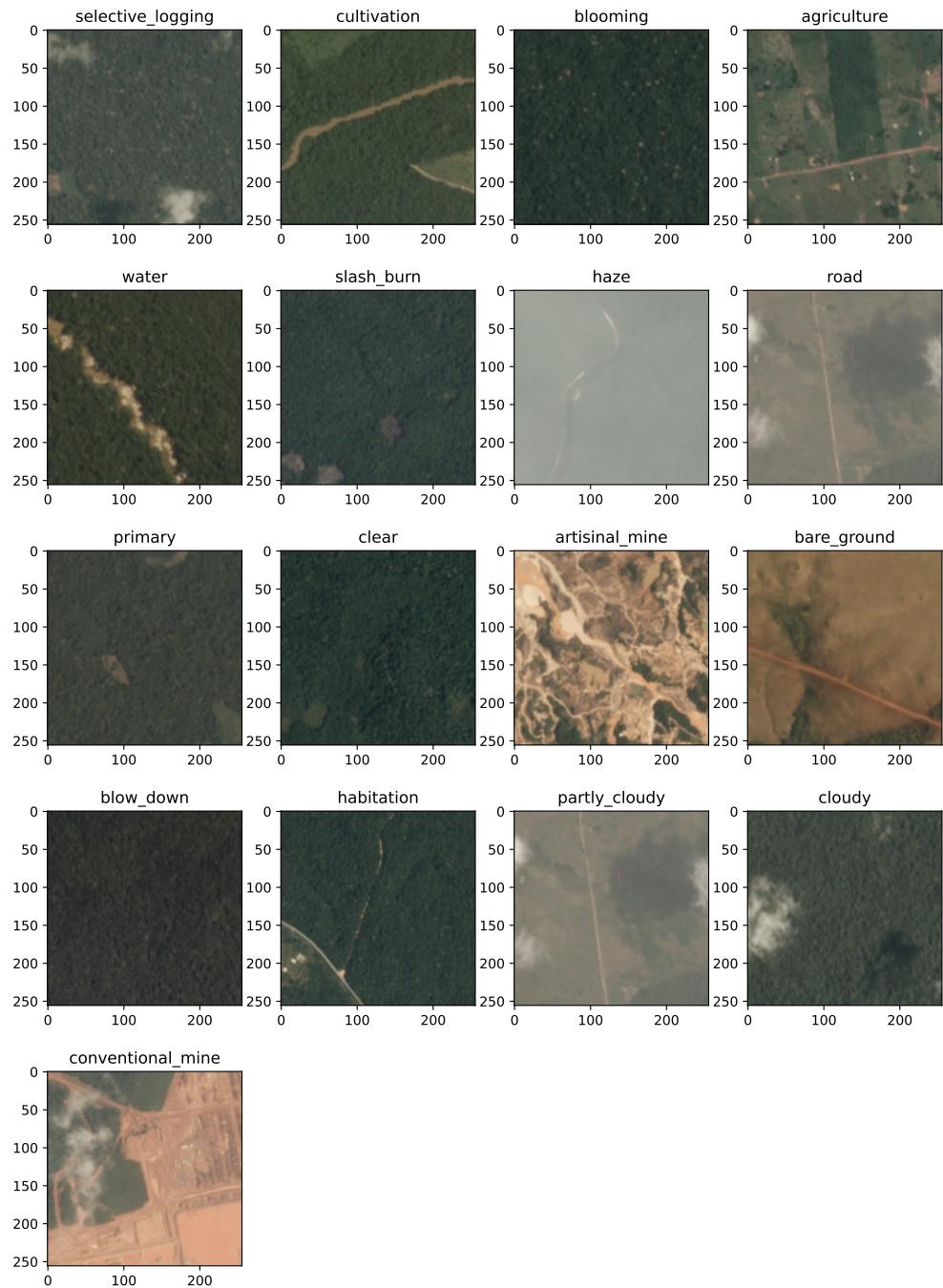


Figura 16 – Amostragem de cada classe de rótulo. Fonte: Autor

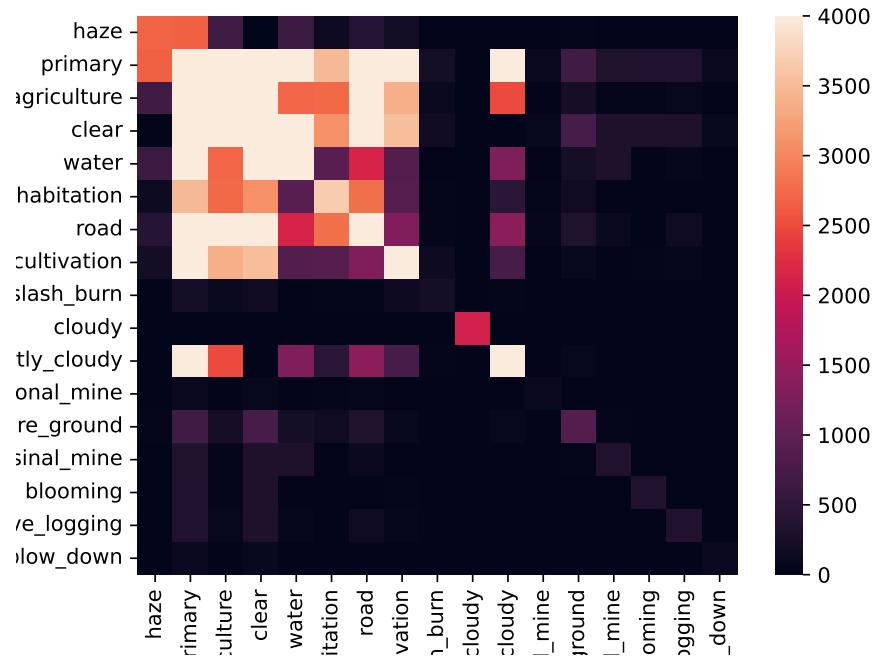


Figura 17 – Matriz de co-ocorrência. Fonte: Autor



Figura 18 – Agrupamento de amostras via técnica TSNE. Fonte: Autor

3.6.4 Definição do modelo

Para a definição do modelo, é interessante começar com modelos de menor capacidade, e ao longo da sintonização poder trocar por um de maior capacidade. Por isso, para o experimento base primeiro utilizou-se o modelo ResNet-18 e após validado utilizou-se o ResNet-50, correspondente a figura 19. Da mesma forma, a implementação do modelo Swin disponibilizado pelo autor de menor capacidade é o *Swin-Tiny*, de número de parâmetros e desempenho próximo ao ResNet50 em *benchmarks* no dataset Imagenet-1k.

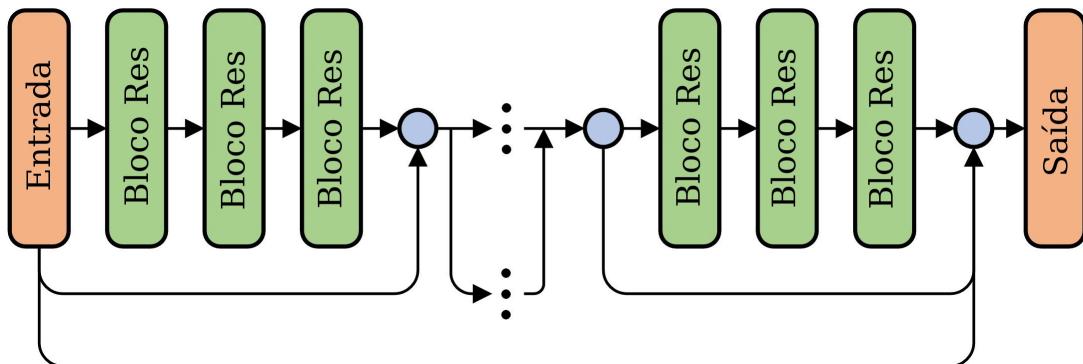


Figura 19 – Arquitetura de modelo base. Fonte:WikiCommons

Tabela 2 – Características do modelo fornecida pela biblioteca

Nome	Pré-Treino	Resolution	Acurácia@1	Acurácia@5	Parâmetros	FLOPs
Swin-T	ImageNet-1K	224x224	81.474	95.5	28.3M	4.5G
Resnet50	ImageNet-1K	224x224	80.858	92.9	25.5M	4.1G

O modelo Swin proposto corresponde a figura 20. Foi elaborado a partir de uma instanciação de um classificador de mesmo *backbone* da biblioteca de modelos *timm*. Em seguida foram a camada de saída, chamada de cabaça do transformer, por uma camada completamente conectada, de tamanho de entrada o número de dimensões do espaço de estados e a dimensão de saída o número de atributos/classes.

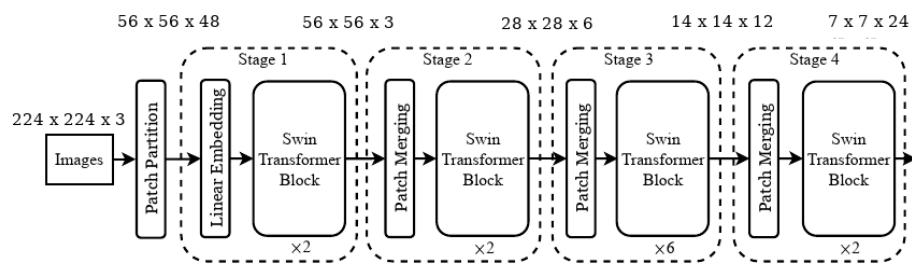


Figura 20 – Arquitetura de modelo proposto. Fonte: ([LIU et al., 2022](#))

3.6.5 Treino

Para a etapa de treino, utilizamos a arquitetura proposta em 3.3, que consiste em um modelo pré-treinado, e retreinando suas últimas camadas de classificação por camadas conectadas fortemente seguidas de camadas de *sigmoide*, como ilustra a imagem a seguir

3.6.6 Validação

A etapa de validação será realizada a medida do desempenho do modelo no conjunto de testes para classificação.

3.6.7 Seleção de modelos

Esta etapa concerne em gerar vários modelos e selecionar conforme o desempenho. É um processo experimental e de vasculhar diferentes hiperparâmetros e componentes do treino.

Critérios de seleção de modelo: Melhor métrica f2 entre diferentes modelos. Com parada antecipada no menor valor de perda de validação em um treinamento, até no máximo à quinta época, devido a limitações de tempo e orçamento de horas de instâncias de GPU. O segundo item foi possível de ser realizado. Realizando *fine-tune* do modelo o modelo Resnet-16 inicialmente. Foram utilizados os pesos dessa mesma rede treinada para o dataset IMAGENET-1k, realizando a troca das camadas de saída, originalmente para 1000 classes. Foram Removidas e adicionados uma camada completamente conectada de entrada igual ao número de neurônios da penúltima camada. Após a última camada foi adicionado uma camada de Síntese, que realizará a conversão de valores lineares para probabilidade de cada classe.

O modelo obteve um desempenho inicialmente satisfatório de métrica f2 de 0,88.

A partir deste modelo inicial, foram feitos vários ajustes de hiperparâmetros e de componentes a fim de aperfeiçoar o desempenho da rede. Dentre os ajustes para seleção de modelo, constam:

1. Pesos pré-treinados
2. Aumentar capacidade da rede
3. Adicionar regularização
4. Amostrador
5. Aumento de dados aleatória
6. Função de perda

7. Otimizador
8. Taxas de aprendizado
9. Transferência de aprendizado vs Fine Tune
10. Experimentar os mesmos passos para o modelo Swin-T

Tendo inicialmente o modelo ResNet-18, os ajustes de capacidade de rede foram inicialmente se aumentar o número de camadas completamente conectadas, contudo, aumentar o número de camadas de blocos básicos, utilizando o modelo ResNet50 demonstrou melhor métrica F_2 e perda BCE. O mesmo foi testado utilizando com pesos do conjunto de dados ImageNet1k, igualmente com melhor resultados. Com pesos do trabalho [2.4.3](#) também de sensoriamento remoto, também houve aumento expressivo de métrica F-2 em comparação com o modelo pre-treinado no conjunto de dados ImageNet-1k

Os modelos pré-treinados apresentavam convergir próximos da primeira época, sugerindo um rápido sobre-ajuste, para isto, testou-se utilizar um otimizador com decaimento de pesos o AdamW, porém leva a perda de desempenho, supostamente por afetar os pesos já treinados. O uso de regularização se limitou ao drop-out presente nos modelos e camadas de normalização de lote e camada.

Para mitigar o desbalanço de quantidade de amostras nas classes, investigou-se a utilização de outros amostradores, que escolhem amostras baseados em probabilidades especificadas pelo inverso da frequência de cada classe. Contudo, não foi possível de ser realizado. O aumento de dados realizado foi aplicar espelho horizontal ou vertical com probabilidade de 25%, dessa forma, o número de amostras é virtualmente aumentado para 4x o tamanho original.

Outra alternativa a mitigar o desbalanço alterou o valor da função de perda por classe, especificando que as classes mais raras realizariam um maior peso na função de perda. Esta alternativa gerou um aumento de desempenho, porém com rápido sobre-ajuste. Uma terceira alternativa foi a função de perda focal, capaz de melhorar a métrica F_2 substancialmente e mitigar sobreajuste da classe dominante.

Para o otimizador, há trabalhos como em([STEINER et al., 2021](#)), que recomendam o uso de SGD para o ajuste-fino do modelo. Contudo, apresentou convergência muito lenta, incompatível com o orçamento de horas de GPUs. O otimizador escolhido foi o Adam sem decaimento de peso. A convergência do modelo demonstrou-se sensível à taxa de aprendizado. Por isso, recorreu-se ao uso de taxas de aprendizado variáveis, utilizando um agendador de taxa de aprendizado, que reajusta para um décimo da anterior cada vez que a função de perda ou de métrica de validação atinge um platô.

Ainda na seleção de modelos, experimentou-se utilizar transferência de aprendizado, definida por congelar os pesos das camadas extratoras de características e treinar

apenas ultimas camadas. Foi experimentado o ajuste fino, definido por retreinar o modelo inteiro, com as últimas camadas inicializadas com pesos 0 ou aleatórios. A segunda opção demonstrou a melhor desempenho.

Os mesmos passos foram realizados para o modelo Swin-T, chegando em uma iteração final de ambos modelos-base e proposto.

3.6.8 Treinamento ResNet

Foram registradas as métricas durante o treino no modelo dos modelos. Pelos gráficos 21 e 22, observamos a evolução da perda.

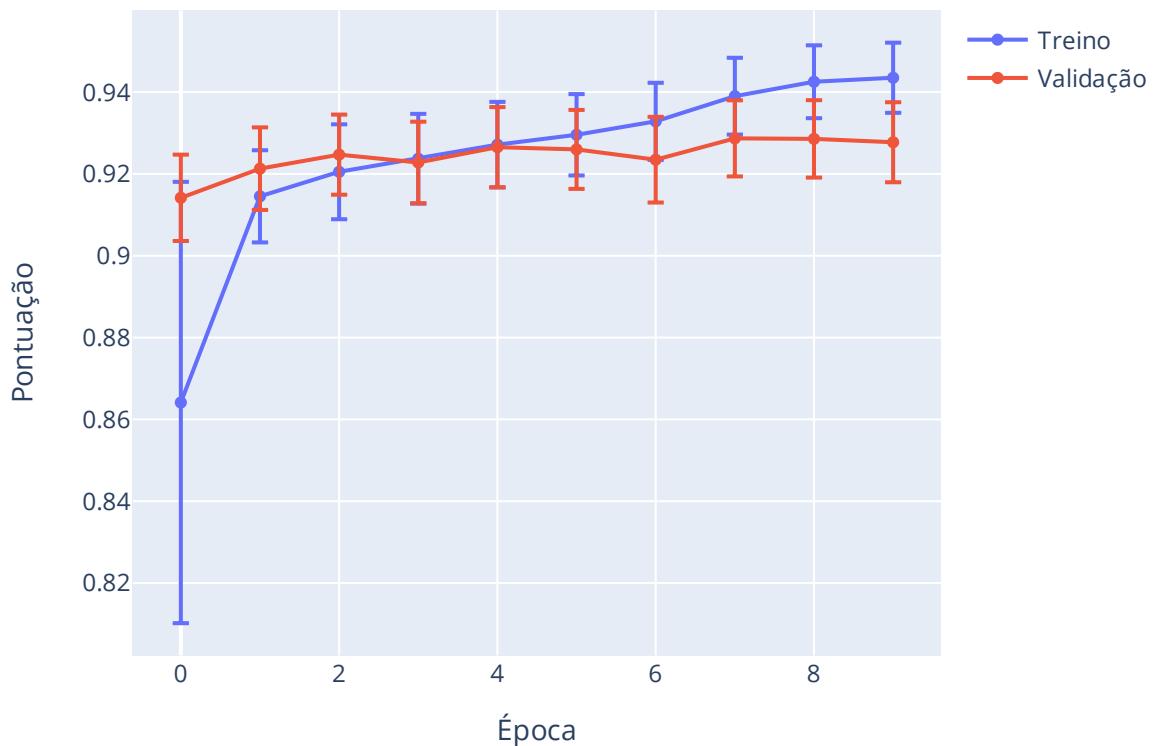


Figura 21 – Métrica F2 em treino e validação por época para a rede ResNet50. Fonte: Autor

3.6.9 Treinamento Swin-T

Foram registradas as métricas durante o treino no modelo dos modelos. Pelos gráficos 23 e 24, observamos a evolução da perda.

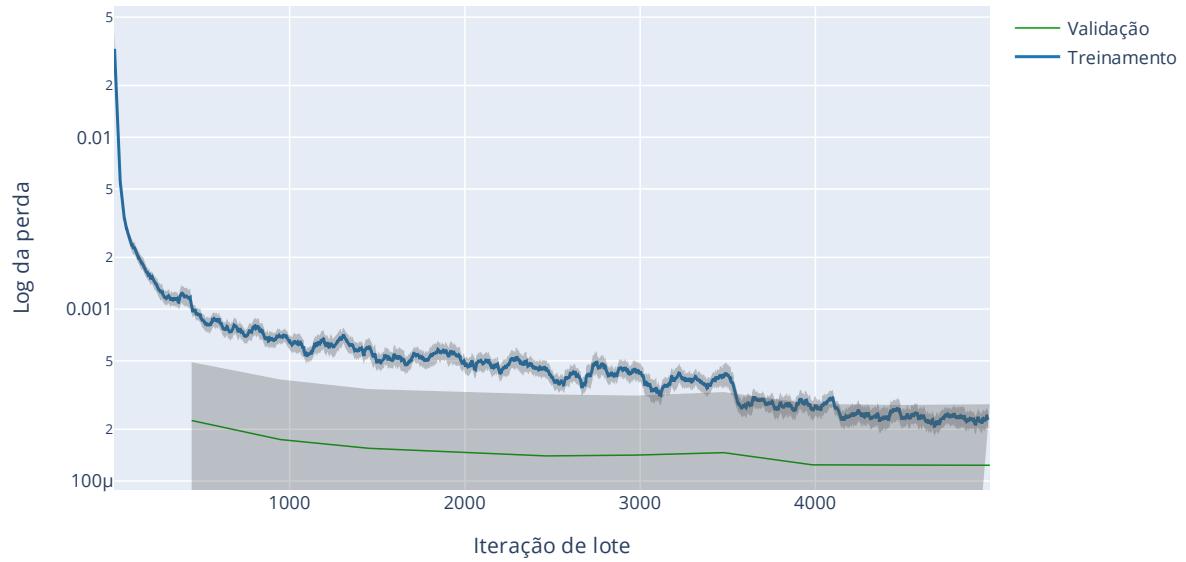


Figura 22 – Métrica de perda em treino e validação por época para a rede ResNet50.
Fonte: Autor

Pelos gráficos de perda e de métrica F_2 podemos determinar quando o modelo atinge o ponto de sobre-ajuste: quando a perda de validação começa a crescer enquanto a de treino continua a decair. Por meio das evoluções de perda também pode-se ser feitos ajustes durante a etapa de varredura de hiperparâmetros. Como exemplo foi a varreduras de taxas de aprendizado.

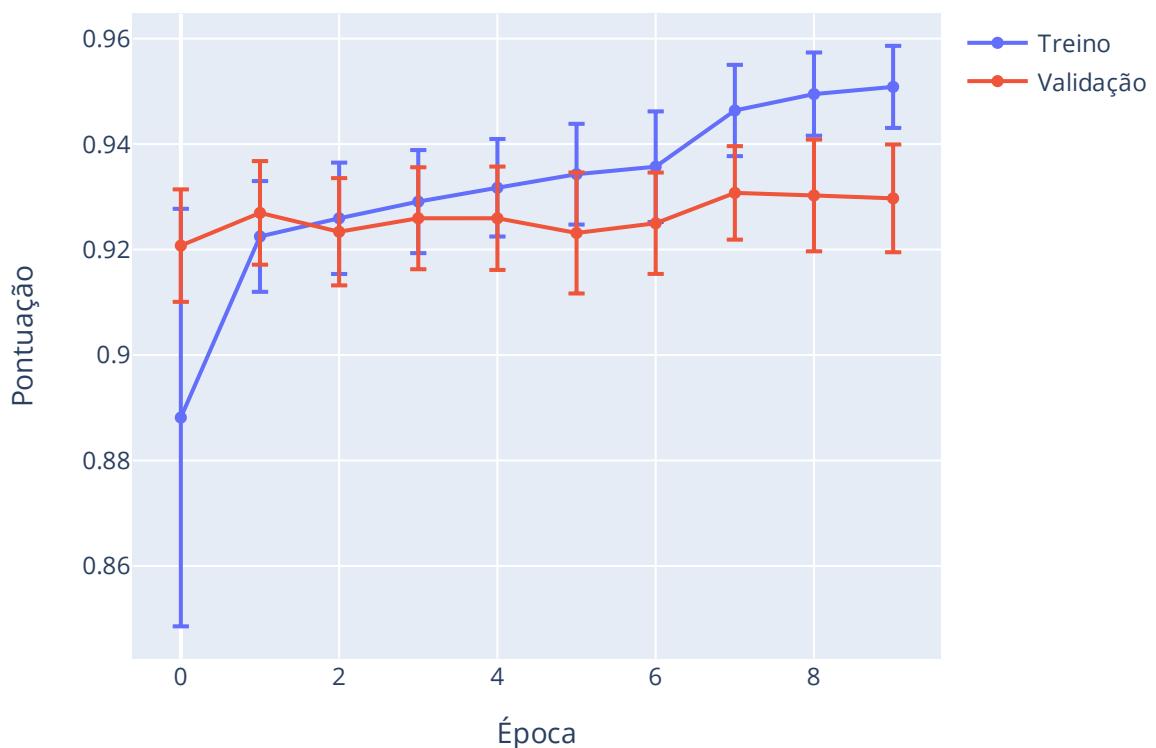


Figura 23 – Métrica F2 em treino e validação por época para a rede Swin-T. Fonte: Autor

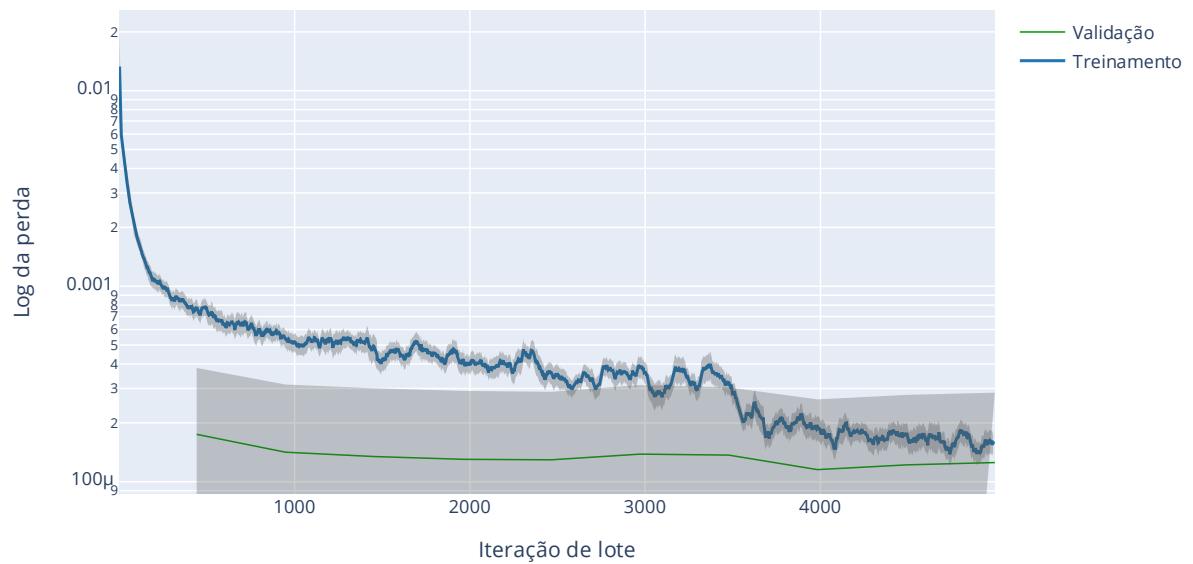


Figura 24 – Métrica de perda em treino e validação por época para a rede Swin-T. Fonte: Autor

4 Resultados

Neste capítulo será apresentado em detalhes os resultados obtidos para o classificador base e o classificador proposto.

4.1 *Classificador Base Resultados Iniciais*

O classificador base pode ser mensurado pelas métricas F_2 e áreas das curvas ROC e PR. A partir da matriz confusão podemos obter com granularidade o comportamento do classificador. Também é possível notar experimentalmente o fato da acurácia em classes raras não refletir a qualidade do modelo: mesmo classificando apenas um verdadeiro positivo corretamente, obteve acurácia de 99,46% para a classe Queimada, observando sua matriz confusão.

O desempenho do modelo base para cada classe foi sumarizado a seguir, em ordem crescente de métrica F_2 . O desempenho do modelo para as diferentes classes foi muito discrepante, expressivamente abaixo para as classes mais raras. O classificador atinge resultados satisfatórios para classes mais abundantes, como o esperado. Contudo, o que os modelos podem se distinguir são nas classes raras. Para isso serão comparados os resultados globais e das classes raras, definidas como possuindo menos de 5% do total de amostras, expostas na tabela 4. Dessa forma é possível de comparar a capacidade de generalização dos modelos e do viés indutivo.

Temos ainda que a métrica AUC-ROC apresentou-se pouco capaz de distinguir a atuação do modelo, como demonstra a figura 26: Classes com baixos escores tiveram áreas similares de classes performáticas. Portanto, a análise a seguir se delimitará às classes raras e com escores precisão-revolução.

4.2 *Classificador Base Análise Para classes Escassas*

Sintetizando os mesmos resultados anteriores para as classes escassas da tabela 4, temos a tabela 5. Com exceção da classe de Garimpo, o modelo teve baixa desempenho, em relação as demais classes removidas. Podemos constatar o mesmo observando a curva PR, da figura 27. Tais classes também obtiveram a curva característica de um classificador pobre.

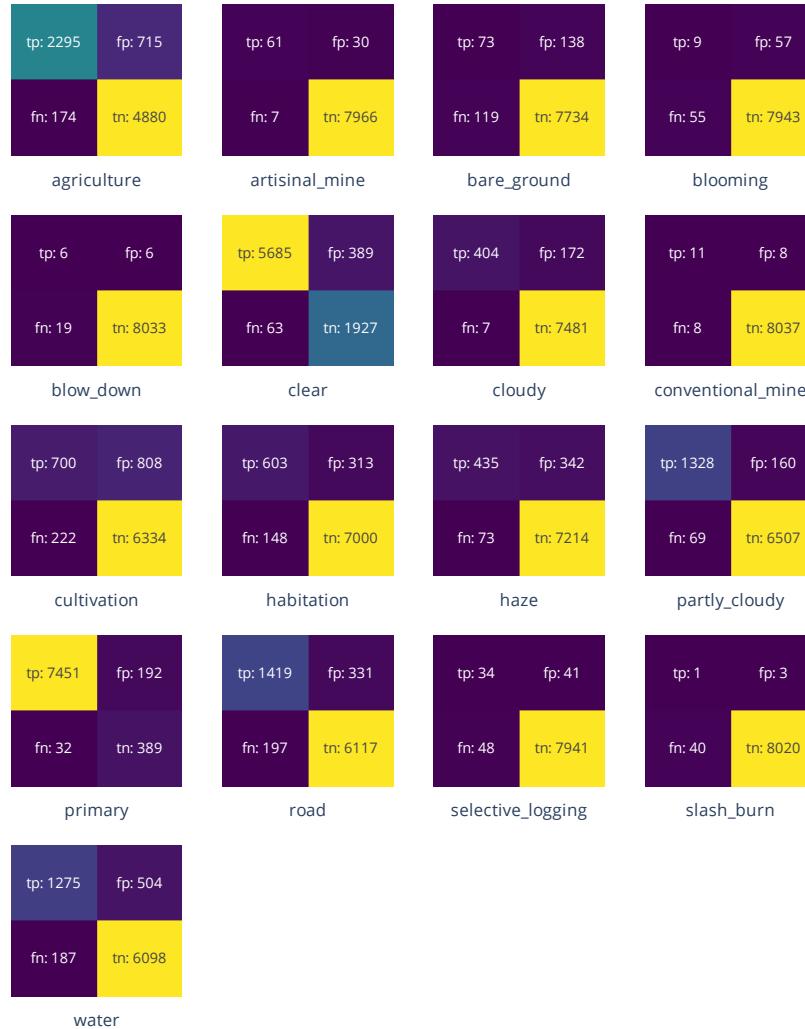


Figura 25 – Matriz Confusão para o modelo base Resnet-50. Fonte: Autor

4.3 Classificador Proposto

Reproduzindo os mesmos procedimentos anteriores para o modelo Swin-T os mesmos resultados anteriores para as classes escassas da tabela 4, temos a tabela 6.

4.4 Comparação de resultados

Nesta sessão compararemos a desempenho dos modelos. Podemos observar pela curva PR 28 que as classes mais desafiadoras permanecem distantes de um classificador ideal, contudo pode-se notar um aumento relevante nas métricas F_2 , PR_{AUC} , portanto qualificando este classificador superior ao modelo base. Temos ainda que embora a métrica

Tabela 3 – Resultados do Modelo ResNet50

	Rótulo	F2	Limiar	PR AUC	AUC Ingênuo
15	slash burn	0,030	0,230	0,145	0,005
3	blooming	0,140	0,130	0,096	0,008
4	blow down	0,268	0,190	0,245	0,003
2	bare ground	0,373	0,210	0,316	0,024
14	selective logging	0,422	0,090	0,401	0,010
7	conventional mine	0,579	0,170	0,536	0,002
8	cultivation	0,674	0,130	0,650	0,114
9	habitation	0,769	0,130	0,802	0,093
10	haze	0,774	0,210	0,784	0,063
16	water	0,836	0,210	0,892	0,181
1	artisinal mine	0,840	0,190	0,880	0,008
13	road	0,864	0,210	0,916	0,200
0	agriculture	0,890	0,250	0,929	0,306
6	cloudy	0,910	0,230	0,946	0,051
11	partly cloudy	0,938	0,210	0,972	0,173
5	clear	0,978	0,210	0,996	0,713
12	primary	0,992	0,190	0,999	0,928
17	global	0,928	0,188	0,677	0,170

Tabela 4 – Classes Raras Dataset *Planet* — Valores do conjunto de dados inteiro

Classe	Rótulo	Amostras	Proporção (%)
Mina Convencional	conventional mine	100	0,247
Roça de Ventos	blow down	101	0,250
Queimada	slash burn	209	0,516
Florescimento	blooming	332	0,820
Garimpo	artisinal mine	339	0,837
Desmatamento Seletivo	selective logging	340	0,840
Área Descoberta	bare ground	862	2,129
Todo conjunto de dados	global	40479	100.0

Tabela 5 – Resultados do Modelo Base

Rótulo	F2	PR AUC	AUC Ingênuo
slash burn	0,030	0,145	0,005
blooming	0,140	0,096	0,008
blow down	0,268	0,245	0,003
bare ground	0,373	0,316	0,024
selective logging	0,422	0,401	0,010
conventional mine	0,579	0,536	0,002
artisinal mine	0,840	0,880	0,008
global	0,928	0,677	0,170

F_2 esteja distante de um classificador ideal, de valor próximo a 1, temos que ela ainda

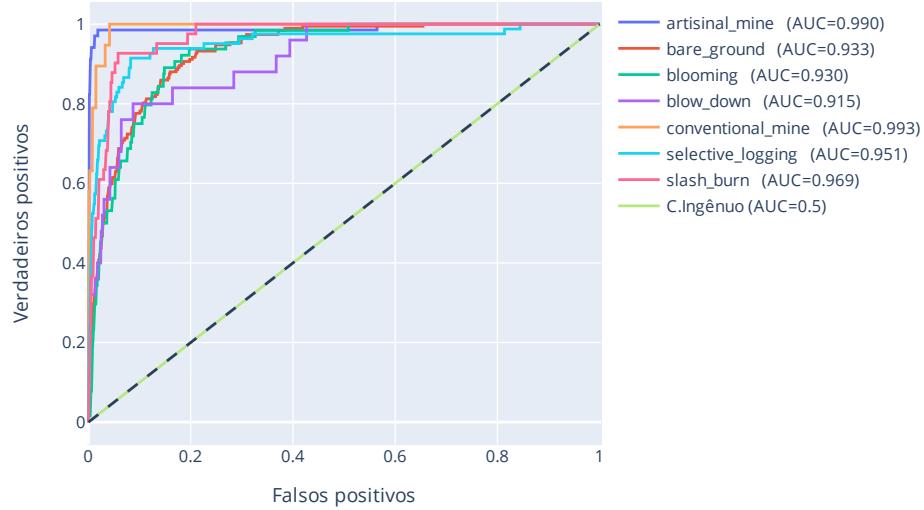


Figura 26 – Curva COR para o modelo base. Fonte: Autor

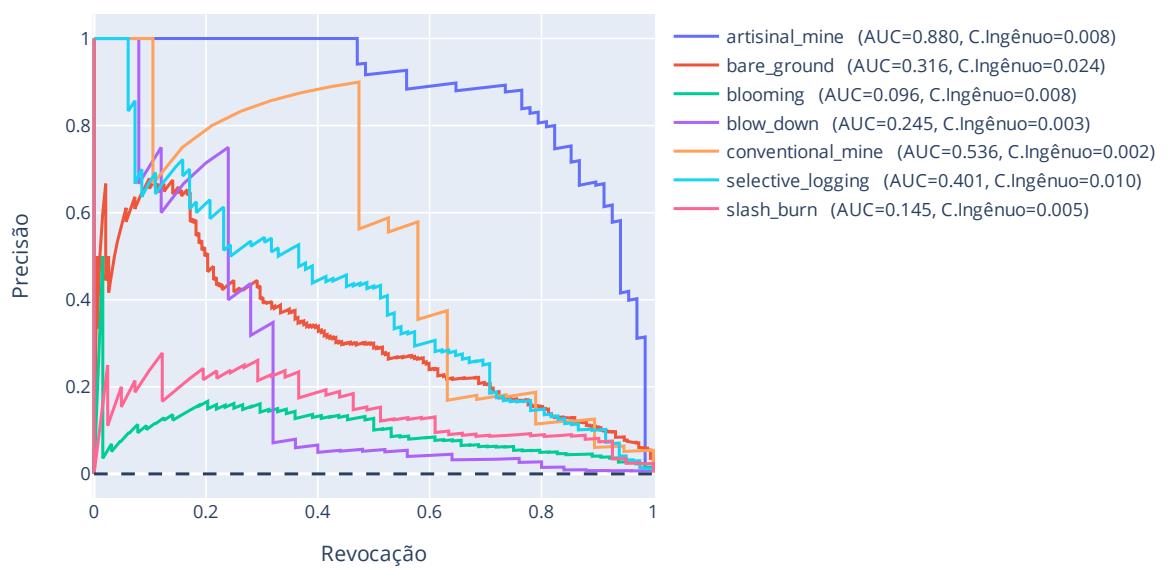


Figura 27 – Curva PR para o modelo base. Fonte: Autor

é bastante superior ao classificador ingênuo/aleatório. Comparando com trabalho de (KASELIMI et al., 2022), de métrica F_2 de 0,878, concluímos que este classificador possui desempenho satisfatório.

Tabela 6 – Resultados do Modelo proposto Swin-T

Rótulo	F2	PR AUC	PR-AUC Class.Ingênuo
blooming	0,230	0,129	0,008
slash burn	0,338	0,342	0,005
blow down	0,310	0,279	0,003
bare ground	0,447	0,344	0,024
selective logging	0,475	0,422	0,010
conventional mine	0,538	0,625	0,002
artisinal mine	0,872	0,880	0,008
global	0,930	0,704	0,170

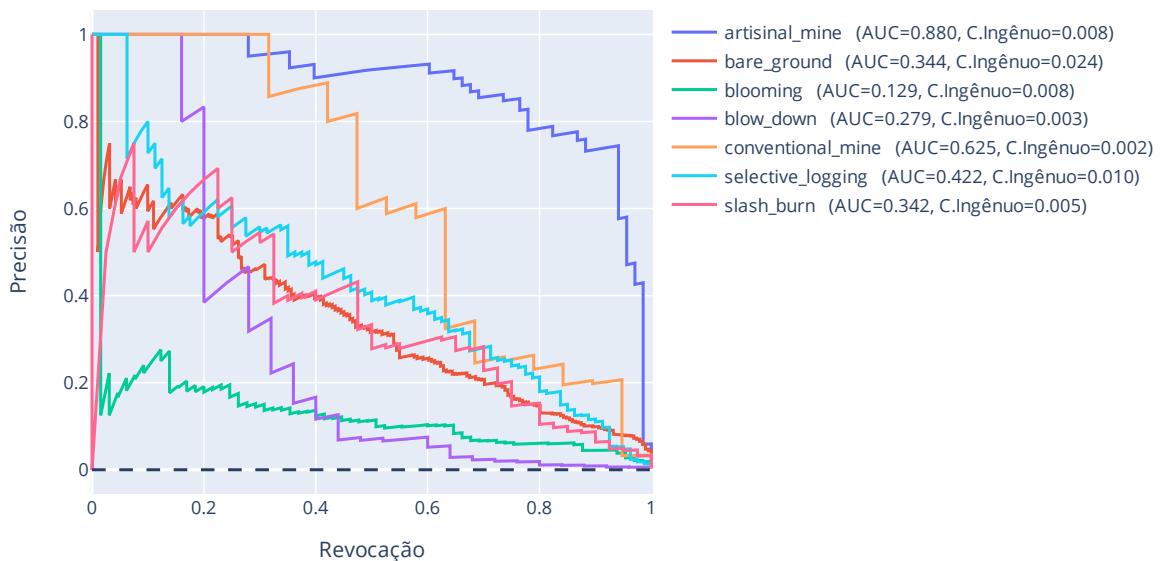


Figura 28 – Arquitetura de modelo Swin-T proposto. Fonte: Autor

Tabela 7 – Comparaçāo de resultados da métrica F2 entre Modelo base e proposto.

Rótulo	F2-Resnet	F2-SwinT
slash burn	0,030	0,338
blooming	0,140	0,230
blow down	0,268	0,310
bare ground	0,373	0,447
selective logging	0,422	0,475
conventional mine	0,579	0,538
artisinal mine	0,840	0,872
global	0,928	0,930

Tabela 8 – Comparação de resultados da métrica PR-AUC entre Modelo base e proposto.

Rótulo	PR-AUC-ResNet	PR-AUC-SwinT	PR-AUC Class.Ingênuo
slash burn	0,145	0,342	0,005
blooming	0,096	0,129	0,008
blow down	0,245	0,279	0,003
bare ground	0,316	0,344	0,024
selective logging	0,401	0,422	0,010
conventional mine	0,536	0,625	0,002
artisinal mine	0,880	0,880	0,008
global	0,677	0,704	0,170

Através das tabelas 8 e 7, podemos constatar que o modelo Swin apresentou melhor desempenho que o modelo base em todas as classes raras, bem como desempenho global, dado as métricas escolhidas. Dessa forma conseguiu melhor generalizar no contexto de poucas amostras.

Para uma validação final, podemos inspecionar visualmente as probabilidades marginais atribuídas a cada classe. É possível notar que para as classes raras, ainda são atribuídas probabilidades marginais inferiores, contudo também possuem menores limiares de classificação, como mostra a tabela de resultados e limiares presente nos anexos. 9

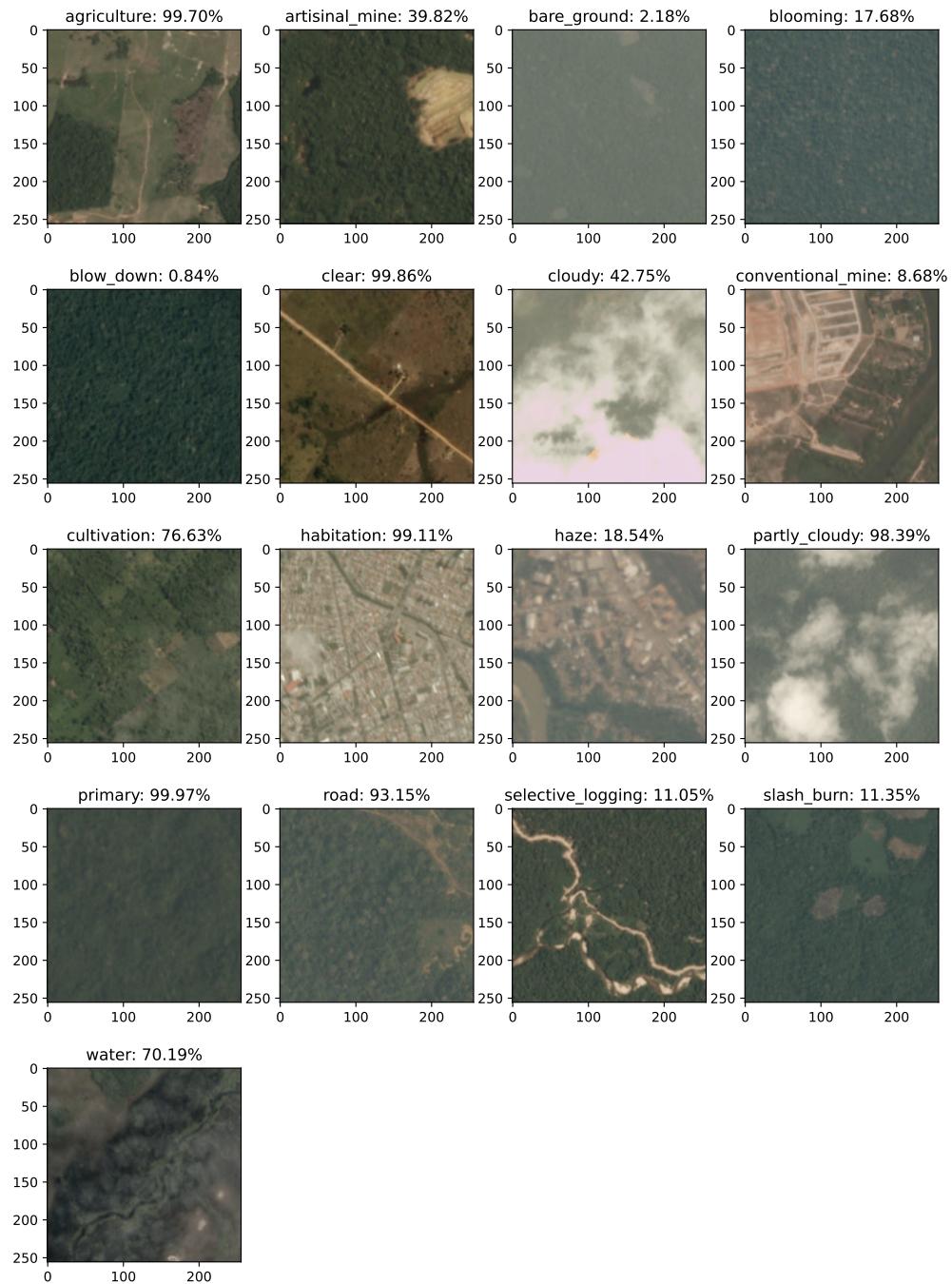


Figura 29 – Probabilidade de inferência para cada classe. Fonte: Autor

5 Conclusão

Este trabalho primeiro realizou um estudo teórico sobre o sensoriamento remoto e aprendizado de máquina profundo, citando tanto conceitos fundamentais, como pontos onde as redes CNNs não apresentam bom desempenho, visando a criar um modelo aplicável à detecção de áreas irregulares como de garimpo e queimadas. Apresentou novas arquiteturas de outro paradigma de redes neurais. Contribuiu fornecendo a metodologia e aspectos necessários para realizar experimentos análogos.

Também Foi possível realizar uma prova de conceito onde a arquitetura Swin é aplicável para o contexto de sensoreamento remoto para detecção de cenas. Teve como ponto de partida um dataset da bacia da floresta amazônica com cenas de eventos comuns e raros e uma busca de metodologia para problemas análogos. Obteve-se um modelo baseado na arquitetura SWIN de melhor pontuação global e de principalmente nas classes raras, que indicou melhor capacidade de generalização sobre classes com poucas amostras aos já estabelecidos redes convolucionais residuais. Embora não tenha sido possível realizar a mesma comparação com outros conjuntos de dados, como o já mencionado Amazon Ponds, pode-se demonstrar que o modelo obtido além de compatível para esse problema, consegue superar em métricas significativas arquiteturas já estabelecidas, sendo um forte arquitetura candidata para futuros sistemas de sensoriamento remoto.

5.1 Trabalhos futuros

Para trabalhos futuros, é possível aplicar a mesma metodologia para avaliar a comparação em conjunto de dados diferentes. Algumas opções interessantes de experimentos que não foram exploradas, podem ser citadas:

- Utilizar o extrator de características para obter o mapa de atenção e ter explicabilidade do modelo
- Realizar o experimento em outros datasets para validação das conclusões
- Embutir esta prova de conceito em uma aplicação real para monitoramento remoto.
- Utilizar função de perda Hamming, adequada para classificações multi-rótulos.
- Utilizar Aumento de dado, por meio de geração de dados sintéticos para classes raras utilizando ruido gaussiano.

- Utilizar o canal de infravermelho próximo, já que vários satélites provêm esse espectro, e várias técnicas de sensoriamento remoto utilizam desse espectro e de ondas mais longas.
- Normalizar cada amostra com a média e desvio do próprio dataset, em vez de utilizar os do conjunto de dados ImageNet1k.

Referências

- ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: *2017 International Conference on Engineering and Technology (ICET)*. [S.l.: s.n.], 2017. p. 1–6. Citado na página [24](#).
- ALOM, M. Z. et al. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*, 2018. Citado 2 vezes nas páginas [16](#) e [25](#).
- AMIGO, I. Scientists are trying to pin down how quickly climate change, deforestation and fires might ruin the world's largest tropical rainforest. *Nature*, v. 578, n. 505, 2020. Citado na página [15](#).
- BALANIUK, R.; ISUPOVA, O.; REECE, S. Mining and tailings dam detection in satellite imagery using deep learning. *Sensors*, v. 20, n. 23, 2020. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/20/23/6936>>. Citado na página [30](#).
- CAMALAN, S. et al. Change detection of amazonian alluvial gold mining using deep learning and sentinel-2 imagery. *Remote Sensing*, v. 14, n. 7, 2022. ISSN 2072-4292. Disponível em: <<https://www.mdpi.com/2072-4292/14/7/1746>>. Citado 2 vezes nas páginas [30](#) e [33](#).
- CHEN, X.; HSIEH, C.-J.; GONG, B. *When Vision Transformers Outperform ResNets without Pre-training or Strong Data Augmentations*. arXiv, 2021. Disponível em: <<https://arxiv.org/abs/2106.01548>>. Citado 2 vezes nas páginas [8](#) e [30](#).
- DOSOVITSKIY, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. Citado 6 vezes nas páginas [8](#), [17](#), [27](#), [28](#), [29](#) e [30](#).
- EMERY, B.; CAMPS, A.; RODRIGUEZ-CASSOLA, M. *Introduction to Satellite Remote Sensing: Atmosphere, Ocean, Land and Cryosphere Applications*. Elsevier Science, 2017. ISBN 9780128092590. Disponível em: <<https://books.google.com.br/books?id=sZLUDQAAQBAJ>>. Citado na página [18](#).
- GOODFELLOW, I. J.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. <<http://www.deeplearningbook.org>>. Citado na página [19](#).
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning*. New York, NY, USA: Springer New York Inc., 2001. (Springer Series in Statistics). Citado 3 vezes nas páginas [20](#), [23](#) e [24](#).
- INSTRUTORGIS. *QGIS: Satélite Amazonia-1 – Composição Colorida RGB*. 2022. Disponível em: <<https://www.instrutorgis.com.br/qgis-satelite-amazonia1-composicao-colorida-rgb>>. Citado 2 vezes nas páginas [8](#) e [18](#).
- KASELIMI, M. et al. A vision transformer model for convolution-free multilabel classification of satellite imagery in deforestation monitoring. *IEEE Transactions on*

- Neural Networks and Learning Systems*, p. 1–9, 2022. ISSN 2162-2388. Citado 2 vezes nas páginas 31 e 51.
- LIU, Z. et al. Swin transformer v2: Scaling up capacity and resolution. In: *CVPR 2022*. [s.n.], 2022. Disponível em: <<https://www.microsoft.com/en-us/research/publication/swin-transformer-v2-scaling-up-capacity-and-resolution/>>. Citado 5 vezes nas páginas 8, 31, 32, 35 e 41.
- LOVEJOY, T.; NOBRE, C. Amazon tipping point. *Science Advances*, v. 4, p. eaat2340, 02 2018. Citado na página 15.
- MAATEN, L. van der; HINTON, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, v. 9, n. 86, p. 2579–2605, 2008. Disponível em: <<http://jmlr.org/papers/v9/vandermaaten08a.html>>. Citado na página 37.
- MAPBIOMAS. *Nota Técnica sobre Garimpo no Rio Madeira*. 2021. Citado na página 15.
- MITCHELL, T. M. *Machine Learning*. New York: McGraw-Hill, 1997. ISBN 978-0-07-042807-2. Citado na página 19.
- PLANETCO, K. *Planet: Understanding the Amazon from Space - Use satellite data to track the human footprint in the Amazon rainforests*. Kaggle, 2016. Disponível em: <<https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/data>>. Citado 2 vezes nas páginas 8 e 34.
- ROSTAMI, M. *Learning Transferable Knowledge Through Embedding Spaces*. Tese (Doutorado) — University of Pennsylvania, 2019. Citado 2 vezes nas páginas 26 e 27.
- SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.*, v. 3, p. 210–229, 1959. Citado na página 19.
- SANCHES, M. K. *Aprendizado de Máquina Semi-supervisionado: Proposta de um Algoritmo para Rotular Exemplos a Partir de Poucos Exemplos Rotulados*. Dissertação (Mestrado) — Universidade de São Paulo, 2003. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-12092003-101358>>. Citado na página 27.
- SATYAMURTY, P.; COSTA, C. P. W. da; MANZI, A. O. Moisture source for the amazon basin: a study of contrasting years. *Theoretical and Applied Climatology*, Springer, v. 111, n. 1, p. 195–209, 2013. Citado na página 15.
- SEDAGHAT, A.; MOKHTARZADE, M.; EBADI, H. Uniform robust scale-invariant feature matching for optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, v. 49, n. 11, p. 4516–4527, 2011. Citado na página 16.
- STEINER, A. et al. How to train your vit? data, augmentation, and regularization in vision transformers. arXiv, 2021. Disponível em: <<https://arxiv.org/abs/2106.10270>>. Citado 2 vezes nas páginas 22 e 43.
- VALERIANO, D. de M.; NARVAES, I.; MAIA, J. Metodologia do sistema deter-b (sistema de detecção do desmatamento e alterações na cobertura florestal em tempo quase real) mapeamento de alertas com imagens dos sensores awifs-resourcesat-2 e wfi-cbers-4. *São José dos Campos*, 2016. Citado na página 15.

VASWANI, A. et al. *Attention Is All You Need*. arXiv, 2017. Disponível em: <<https://arxiv.org/abs/1706.03762>>. Citado na página 29.

WANG, D. et al. An empirical study of remote sensing pretraining. *arXiv preprint arXiv:2204.02825*, 2022. Citado 4 vezes nas páginas 16, 17, 31 e 34.

Anexos

A seguir é documentado resultados completos dos experimentos mencionados

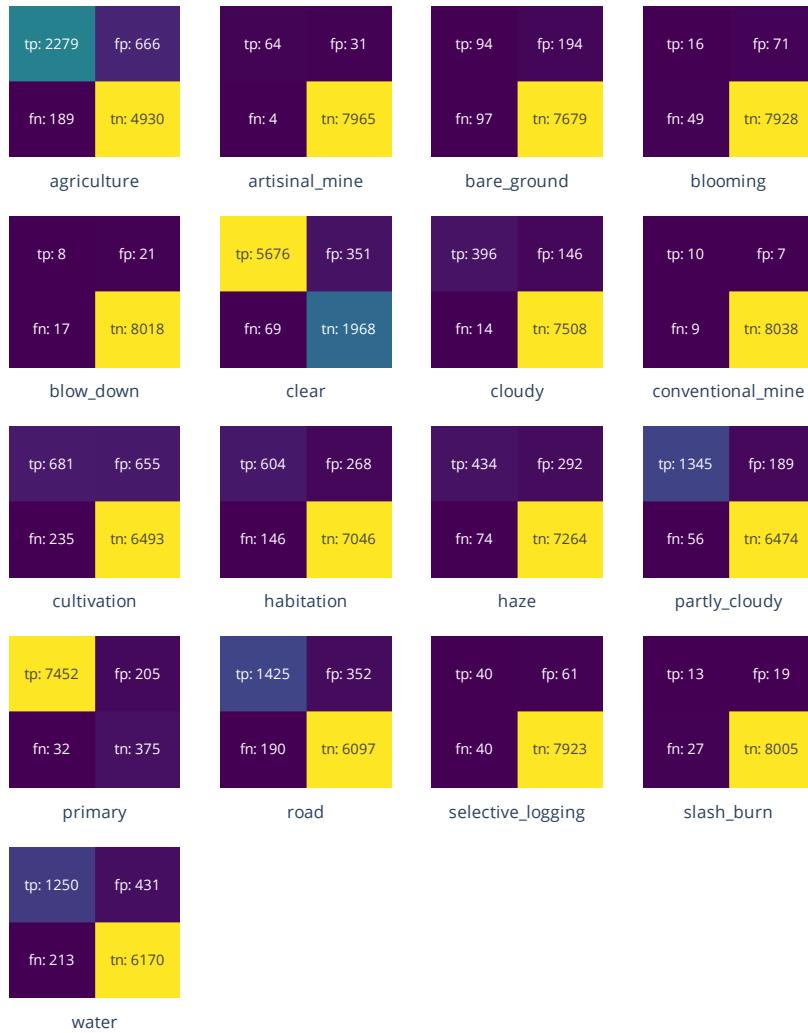


Figura 30 – Matriz Confusão SwinT. Fonte: Autor

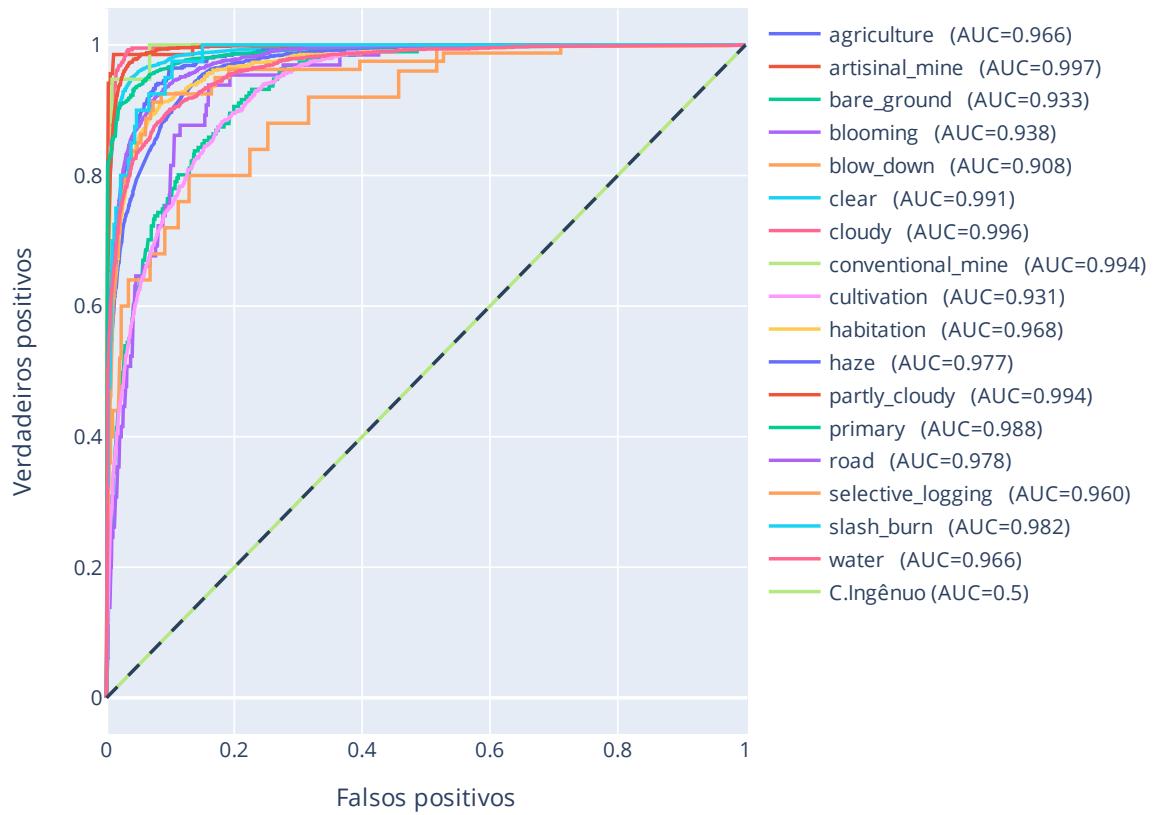


Figura 31 – Curva COR Swin-T. Fonte: Autor

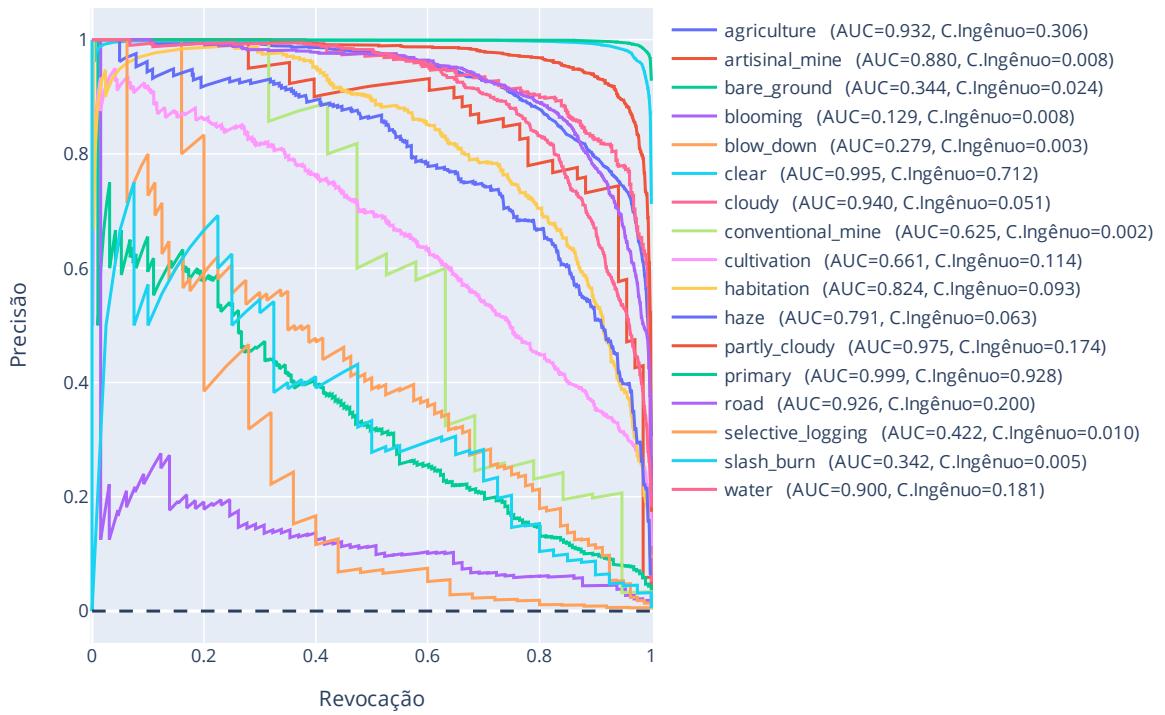


Figura 32 – ACurva PR Swin-T. Fonte: Autor

Tabela 9 – Resultados do Modelo Swin-T

	label	F2	threshold	PR AUC	PR-AUC	Class.Ingênuo
3	blooming	0,230	0,270	0,129		0,008
4	blow down	0,310	0,190	0,279		0,003
15	slash burn	0,338	0,130	0,342		0,005
2	bare ground	0,447	0,170	0,344		0,024
14	selective logging	0,475	0,110	0,422		0,010
7	conventional mine	0,538	0,210	0,625		0,002
8	cultivation	0,681	0,130	0,661		0,114
9	habitation	0,780	0,390	0,824		0,093
10	haze	0,787	0,230	0,791		0,063
16	water	0,830	0,250	0,900		0,181
13	road	0,865	0,190	0,926		0,200
1	artisinal mine	0,872	0,250	0,880		0,008
0	agriculture	0,889	0,290	0,932		0,306
6	cloudy	0,907	0,270	0,940		0,051
11	partly cloudy	0,942	0,170	0,975		0,174
5	clear	0,978	0,190	0,995		0,712
12	primary	0,991	0,230	0,999		0,928
17	global	0,930	0,216	0,704		0,170

Tabela 10 – Resultados do Modelo ResNet50

	label	F2	threshold	PR AUC	PR-AUC Class.Ingênuo
15	slash burn	0,030	0,230	0,145	0,005
3	blooming	0,140	0,130	0,096	0,008
4	blow down	0,268	0,190	0,245	0,003
2	bare ground	0,373	0,210	0,316	0,024
14	selective logging	0,422	0,090	0,401	0,010
7	conventional mine	0,579	0,170	0,536	0,002
8	cultivation	0,674	0,130	0,650	0,114
9	habitation	0,769	0,130	0,802	0,093
10	haze	0,774	0,210	0,784	0,063
16	water	0,836	0,210	0,892	0,181
1	artisinal mine	0,840	0,190	0,880	0,008
13	road	0,864	0,210	0,916	0,200
0	agriculture	0,890	0,250	0,929	0,306
6	cloudy	0,910	0,230	0,946	0,051
11	partly cloudy	0,938	0,210	0,972	0,173
5	clear	0,978	0,210	0,996	0,713
12	primary	0,992	0,190	0,999	0,928
17	global	0,928	0,188	0,677	0,170