

Identificação de mamonas em plantações de cana-de-açúcar utilizando aprendizado ativo

Juan Carlos Conceição de Lima Sales¹, Wesley Nunes Gonçalves²

¹ Faculty of Computer Science, Federal University of Mato Grosso do Sul, UFMS

² Faculty of Engineering, Architecture and Urbanism and Geography,
Federal University of Mato Grosso do Sul, UFMS

Abstract. *Castor bean plants have applications in multiple industries but can become a weed in several crops. Manual identification and control are unfeasible in large plantations. Using machine learning models can make the identification process efficient and accurate. Still, it is necessary to label a large amount of data, which is laborious and inefficient since many of the images are similar and do not improve model training. Active learning allows labeling only data that positively impacts model performance, which can match or surpass the performance of a model trained on the entire set, with significantly less data labeling effort. In this work, we test an active learning method that uses self-supervised pretext tasks to separate the data that must be labeled. The models trained with the pretext tasks presented a decrease in the recall relative to the reference model, which has 68%. We also trained a pseudo task that creates batches with the ideal distribution. The pseudo task's classifier obtained a recall of 86%, while being trained in less than 1% of the dataset.*

Resumo. *A mamona tem aplicações em múltiplas indústrias, mas pode se tornar uma planta daninha em diversas culturas. A identificação e controle manual são inviáveis em grandes plantações. O emprego de modelos de aprendizado de máquina pode tornar o processo de identificação eficiente e preciso, porém é necessário rotular uma grande quantidade de dados, o que é trabalhoso e ineficiente, visto que muitas das imagens são similares e não melhoram o treinamento dos modelos. Active learning permite a rotulação apenas de dados que impactam positivamente o desempenho do modelo, que pode se igualar ou superar o desempenho de um modelo treinado no conjunto inteiro, com um esforço significativamente menor na rotulação dos dados. Neste trabalho, testamos um método de active learning que utiliza tarefas auxiliares auto supervisionadas para separar os dados que devem ser rotulados. Os modelos treinados com as tarefas auxiliares apresentaram diminuição na revocação em relação ao modelo de referência, que obteve 68%. Também treinamos uma pseudo tarefa, que gera os batches com a distribuição ideal. O classificador da pseudo tarefa obteve uma revocação de 86%, sendo treinado em menos de 1% do conjunto de dados.*

1. Introdução

Ricinus communis, conhecida no Brasil como mamona, é uma planta tropical com aplicações em diversos setores da indústria. O principal subproduto da mamona é o óleo extraído de suas sementes, sendo algumas de suas aplicações a produção de biocombustível e utilização pelas indústrias de farmacêutica, cosmética, química, alimentícia e têxtil [Cândido et al. 2008] [Saadaoui et al. 2017]. Alguns dos subprodutos

do processo de extração do óleo de mamona podem ser utilizados como adubo orgânico [LIMA et al. 2008].

Apesar de seus usos, a mamona pode se tornar uma planta daninha em diversas culturas agrícolas. A mamona pode crescer agressivamente em uma variedade de ambientes, competindo com as culturas por recursos essenciais como água, nutrientes, luz e espaço. Seu rápido crescimento e folhagem densa podem privar a cultura de luz solar, levando à redução da fotossíntese e subsequentemente ao declínio na produção da mesma. Os frutos da mamona eclodem quando expostos ao sol, lançando sementes a longas distância. Uma das principais culturas afetadas pela presença de mamonas é a cana-de-açúcar [Zera et al. 2011].

O impacto econômico da infestação de mamonas é considerável e não se reduz apenas à produtividade das culturas, mas também aumenta o custo da produção agrícola devido à necessidade do emprego de práticas adicionais para seu controle, como a utilização de herbicidas. Além disso, a mamona apresenta princípios tóxicos para humanos e diversas espécies de animais em suas folhas e sementes, que podem ser letais quando consumidas [da Silva Fonseca and Soto-Blanco 2014][de Oliveira and Felizari 2011].

A detecção de plantas daninhas é fundamental para manter uma produção agrícola sustentável. Com a detecção de daninhas é possível determinar o melhor curso de ação para seu controle, prevenindo que elas se estabeleçam e se espalhem. Os métodos tradicionais de detecção de ervas daninhas, como a identificação visual feita por indivíduos no campo, são normalmente demorados, trabalhosos, e muitas vezes ineficazes em áreas extensas [Costa et al. 2018].

A incorporação de sistemas de detecção automatizados pode melhorar significativamente a eficiência e a precisão da identificação de infestações de mamonas. Sistemas de detecção podem ser embarcados em tratores, e quando pareados à ferramentas para remoção das daninhas, possibilitam o manejo em tempo real [Costa et al. 2018]. Com o emprego de drones equipados com câmeras de alta resolução é possível cobrir grandes áreas, aplicando algoritmos de visão computacional para realizar a detecção eficientemente.

Algoritmos de visão computacional pura podem não apresentar bom desempenho na detecção de mamonas em certos cenários, como regiões com alta concentração de plantas, onde a cultura e a planta daninha estão muito próximas, ou até mesmo diferentes tamanhos de plantas. Esses algoritmos podem também variar em desempenho em diferentes condições de iluminação. O emprego de algoritmos de aprendizado de máquina abre novos caminhos para enfrentar os desafios da tarefa [Ferreira 2017] e [Oda and Castro Neto 2023]. Esses algoritmos são treinados em conjuntos de dados rotulados para reconhecer as características únicas das mamonas, como formato da folha, padrão de crescimento e coloração.

A rotulação desses dados entretanto apresenta novos desafios, que afetam a vantagem desse método em detrimento dos demais, até mesmo da detecção manual. O primeiro deles está relacionado a área coberta pela cultura. Em um único voo é possível capturar dezenas milhares de imagens, o que torna o processo de rotulação demorado. O segundo está no custo computacional e tempo necessário para o treinamento de um modelo de aprendizado de máquina em um grande conjunto de dados. É necessário encontrar um

balanço entre a quantidade de imagens e a precisão desejada, o que torna esse o problema de selecionar as imagens que agregam o maior aumento de desempenho ao modelo.

O primeiro desafio pode ser parcialmente solucionado com *model-assisted labeling*, que consiste em treinar modelos de aprendizado de máquina para automaticamente rotular dados. Assim, é possível rotular apenas o subconjunto do conjunto de dados usado para treinar o modelo de rotulação. [Pham et al. 2024] aplica *model-assisted labeling* para a criação de um extenso conjunto de dados para detecção de águas vivas. O segundo desafio no entanto demanda uma solução mais sofisticada, visto que em um paradigma tradicional de aprendizado de máquina não é possível saber se um subconjunto de dados agrega ao desempenho do modelo sem antes rotulá-lo e alimentá-lo ao modelo, criando assim uma dependência circular. Esse problema pode ser contornado através da utilização de *active learning* (AL).

Active learning é uma ramificação do aprendizado de máquina em que o algoritmo seleciona iterativamente as amostras mais informativas do conjunto de dados para rotulagem [Settles 2009]. Com o emprego de AL, é possível obter-se um desempenho similar ou até mesmo superior ao do modelo treinado no conjunto de dados inteiro, em um subconjunto substancialmente menor, assim reduzindo o tempo e os recursos computacionais necessário para o treinamento. [Yi et al. 2022] propõe uma abordagem auto supervisionada, que utiliza tarefas auxiliares para determinar o "grau de dificuldade" de uma imagem, e por conseguinte seu impacto no desempenho do modelo.

Apesar dos benefícios da utilização de AL, a literatura sobre sua aplicação à identificação de plantas daninhas é escassa. Assim, o presente trabalho ajuda a preencher essa lacuna e aplica AL à detecção de mamonas em plantações de cana-de-açúcar.

O método utilizado foi proposto por [Yi et al. 2022] e aplica tarefas auxiliares a separação de dados relevantes para o treinamento dos modelos. O método foi escolhido pelos resultados obtidos pelos autores no trabalho original, que superam os resultados dos demais métodos de AL disponíveis na literatura. Testamos múltiplos 4 tarefas auxiliares para a amostragem dos dados, e uma pseudo tarefa que realiza uma amostragem ideal. Os resultados dos classificadores treinados com AL foram comparados a um classificador treinado sem AL no conjunto de dados inteiro.

As tarefas testadas não se adaptaram ao conjunto de dados e seus respectivos classificadores tiveram uma revocação inferior a do classificador base, que teve a revocação média de 68%. Isso indica que os classificadores baseados nas tarefas auxiliares falharam em identificar um número maior de mamonas do conjunto de dados. A pseudo tarefa, no entanto, teve revocação média de 86%, sinalizando a importância de uma boa distribuição de mamonas nos dados amostrados pelas tarefas. O classificador baseado na pseudo tarefa obteve o maior desempenho dos testes realizados, sendo treinado em menos de 1% das imagens do conjunto de dados.

2. Revisão de Literatura

2.1. Detecção de Plantas Daninhas

A detecção automática de plantas daninhas consiste em duas tarefas principais: extrair as características que descrevem uma planta e determinar se o objeto identificado é uma

planta daninha ou não. Efetivamente, os métodos diferem na forma como essas tarefas são executadas.

As características utilizadas na identificação de plantas podem ser visuais, como cor e formato, bem como espaciais, como a altura e densidade da planta.

Na etapa de classificação da planta, os métodos variam em complexidade e eficiência. Algoritmos de visão computacional e algoritmos tradicionais de aprendizado de máquina criaram a fundação para detecção automática de objetos. No entanto, os avanços em hardware, especialmente em GPUs, na última década possibilitaram o treinamento de modelos de aprendizado profundo cada vez maiores e mais precisos. Esses modelos atualmente compõem o estado da arte em detecção automática de objetos, embora algoritmos tradicionais ainda sejam amplamente utilizados.

O método empregado no trabalho de [Molina-Villa et al. 2016] envolve o uso de técnicas de filtragem de imagens RGB para a detecção de plantas daninhas em meio à cultura. Essas operações reduzem as características visuais das imagens em um subconjunto de características mais significativas. Inicialmente, o algoritmo filtra aos objetos na imagem por cor e formato, removendo quaisquer objetos que não façam parte da vegetação. A imagem resultante é então convertida em uma imagem binária, em que os pixels brancos correspondem à vegetação. Por fim, os objetos são classificados como parte da cultura ou plantas daninhas de acordo com sua área em pixels.

Similarmente, [Perez et al. 2000] utiliza análise de cores e formato de objetos para detecção de plantas daninhas a partir de imagens RGB. O método proposto utiliza histogramas de imagem e um processo de refinamento baseado em imagens binárias para segmentar as plantas. O resultado da segmentação é alimentado ao processo de análise do formato das plantas, que utiliza uma regra de Bayes e *K-Nearest Neighbors* (KNN) para a identificação das plantas daninhas.

Os experimentos realizados por [Chen et al. 2020] demonstram a eficiência de um método baseado fusão características visuais e *Support Vector Machines* (SVMs) na detecção de mudas de milho e plantas daninhas. O trabalho de [Aparna and Radha 2019] também emprega SVMs na classificação de plantas daninhas. O método descrito pelos autores utiliza um modelo de atenção visual para identificar regiões salientes nas imagens, candidatas a presença de daninhas. As regiões são alimentadas para uma SVM que determina se a região contém ou não daninhas.

É possível ainda o emprego de sensores multiespectrais no processo de captura dos dados. Esses sensores capturam imagens em diferentes comprimentos de onda do espectro eletromagnético. Esses sensores constituem imagens com múltiplos canais de cor, além dos tradicionais canais RGB. Isso permite estimar uma assinatura única de cor para uma espécie de planta.

[Wang et al. 2001] projetou um sensor espectral, capaz de detectar diferentes variedades de plantas daninhas em campos de trigo. O sensor apresentou uma alta precisão na detecção de daninhas, embora a presença de sombras impacte significativamente os resultados.

Em sua pesquisa, [Lin et al. 2017] utiliza imagens múlti espectrais de alta resolução para analisar características de plantas de milho e diferentes espécies de plantas

daninhas. Após a captura dos dados e segmentação das características visuais, os autores utilizam árvores de decisão para identificar as plantas daninhas.

Características espaciais das plantas permitem distinguí-las pela forma como elas ocupam o espaço. [Andújar et al. 2012] utiliza sensores ultrassônicos de distância para detecção de plantas daninhas a partir da diferença de altura entre as plantas. De acordo com os autores, o sistema tem um bom desempenho na detecção de daninhas no estágio inicial de crescimento da cultura. Após esse estágio, as daninhas podem ser ocluídas pela cultura. [Shahbazi et al. 2021] propõe a aplicação de LiDAR na detecção. O processo também utiliza a diferença de altura entre as daninhas e a cultura para a detecção.

Técnicas de aprendizado profundo, como redes neurais convolucionais (CNN), tem se mostrado uma solução viável para a detecção de plantas daninhas. CNNs tendem a menos sensíveis a fatores como iluminação e oclusão, tornando-as a melhor alternativa em diversos casos.

Modelos altamente especializados podem ser construídos usando aprendizado de máquina profundo, permitindo a adaptação da arquitetura às necessidades do domínio do problema. [Punithavathi et al. 2023] propõe um método especializado que utiliza uma Faster RCNN para a detecção de plantas e uma Extreme Learning Machine (ELM) para classificar entre daninha e cultura.

Algoritmos de aprendizado profundo tem um alto custo computacional e de tempo para serem treinados, mas são eficientes durante a etapa de inferência, o que permite sua integração a sistemas embarcados, para atuação em tempo real no manejo de plantas daninhas. O trabalho desenvolvido por [Tummapudi et al. 2023] integra a classificação de plantas a um braço robótico, que atua na remoção das daninhas detectadas.

2.2. Active Learning

Active learning permite amostrar dados em conjuntos de dados não rotulados de maneira informada, selecionando dados mais significativos para o processo de tempo. Isso permite que aplicações com grandes quantidades de dados e/ou baixa disponibilidade de rotuladores obtenham o maior desempenho com o menor número de dados e logo, o mais baixo custo de rotulação. [Zhao et al. 2020] aplica AL ao treinamento de algoritmos de *Neural Machine Translation* (NMT), que traduzem diretamente texto de um idioma para outro, com base em palavras comuns em sentenças. O algoritmo demanda uma grande quantidade de dados textuais, e conseqüentemente muito esforço humano. Ao aplicar AL ao treinamento do modelo, os autores superaram o desempenho do modelo usado como referência, com apenas 20

Quando amostrando dados para treinar um modelo de aprendizado de máquina, as amostras podem acabar sendo atrelados ao modelo em si. Isso torna os dados amostrados enviesados e dificulta a utilização dos mesmos para treinar modelos no futuro. O trabalho desenvolvido por [Katragadda et al. 2022] tenta resolver esse problema propondo métodos de amostragem com foco nos dados, ao invés do modelo final. Além disso, os autores argumentam sobre a necessidade de se amostrar dados não apenas informativos para o modelo utilizado atualmente, mas que possam agregar ao desempenho de modelos utilizados futuramente.

O trabalho de [Cacciarelli and Kulahci 2023] revisa a literatura de AL aplicado

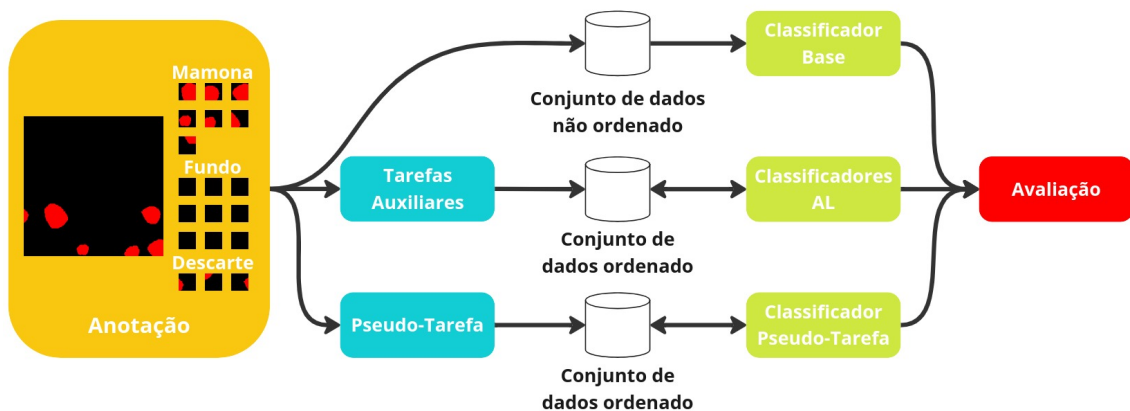


Figura 1. Fluxo dos experimentos

a problemas que envolvem fluxo contínuo de dados. Os autores separam algoritmos de AL em duas categorias: baseado em conjuntos estáticos e baseados em fluxo de dados, sendo conjuntos estáticos amostrados uma vez de um conjunto de dados não rotulados fixos, enquanto métodos baseados em fluxo recebem dados constantemente de um fluxo de dados em tempo real. O trabalho apresenta diversos exemplos de sistemas em tempo real, treinados com AL.

Além disso, as técnicas de AL foram aplicadas com sucesso em diversos domínios, como o treinamento de modelos de classificação para modulação e classificação de sinais [Bhargava et al. 2022], processamento de texto da área da saúde para determinar a condição médica de paciente [Ferreira et al. 2021] e criação de dados sintéticos para testar algoritmos de direção autônoma [Meinke 2021].

3. Materiais e Metodologia

O fluxo de realização deste trabalho está representado na figura 1.

3.1. Conjunto de Dados

O conjunto de dados utilizado neste trabalho é composto por imagens aéreas de plantações de cana-de-açúcar com a presença de mamonas. Esses dados foram capturados em várias plantações em todo o Brasil utilizando diferentes modelos de drones, garantindo uma ampla cobertura e diversidade de condições de plantio. As ortoimagens das plantações foram geradas com uma GSD (Ground Sampling Distance) de aproximadamente 3 cm, o que proporciona um alto nível de detalhe e precisão nas imagens capturadas. Essa alta resolução é essencial para a identificação precisa das mamonas e outras características relevantes das plantações.

Inicialmente as imagens foram capturadas e as mamonas foram rotuladas manualmente por um especialista desenhando um polígono que contorna seu formato aproximado. Para a tarefa de classificação, dividimos as ortoimagens em patches de dimensão 32x32 pixels. Estas são as mesmas dimensões usadas no trabalho original. Dessa forma, para cada imagem no conjunto de dados, checamos se a imagem de anotação correspondente contém pixels de mamonas, e em caso positivo, associamos essa classe à imagem. A Figura 2 mostra um par de imagem RGB e imagem de anotação. Tendo em vista que as

imagens são recortes de uma imagem maior, é possível que apenas uma parte pequena de uma mamona esteja contida na imagem. Para evitar que essa imagem seja rotulada como mamona, atribuímos o rótulo apenas à imagens em que o objeto ocupe mais de 10% dos pixels. As imagens com menos de 10% de pixels de mamonas são descartadas.



Figura 2. Imagem resultante da segmentação de mamonas (esquerda) e imagem RGB correspondente (direita)

O conjunto de dados resultante do processo de divisão mencionado acima contém um total 1.633.744 imagens, sendo 48.156 imagens rotuladas como mamonas e 1.585.588 rotuladas com fundo. Os dados foram divididos de forma estratificada em conjuntos de treino e validação, com uma proporção de 90% dos dados de cada classe alocados para treino e 10% para validação.

3.2. Método Utilizado

O método utilizado como base para esse trabalho foi proposto por [Yi et al. 2022]. O objetivo do trabalho é maximizar o desempenho das tarefas de classificação e segmentação semântica aplicando AL. Essas tarefas são referidas como tarefa principal e são treinadas em ciclos com AL. A cada ciclo, a tarefa principal amostra sem reposição as imagens mais desafiadoras do conjunto de dados não rotulados e as encaminha para o *oráculo* para rotulagem. O oráculo é uma abstração de um rotulador de imagens, podendo ser um modelo para rotulagem automática ou um rotulador humano. As imagens são adicionadas ao conjunto de dados rotulados e usadas para treinar a tarefa principal. Ao final do treinamento, as métricas da tarefa principal são calculadas e o ciclo é repetido. O fluxo de funcionamento do algoritmo é descrito na imagem 3.

Para determinar quais imagens são desafiadoras e devem ser escolhidas, os autores utilizam tarefas auxiliares, que são modelos auto supervisionados treinados no conjunto de dados não rotulado, antes dos ciclos de AL. Na etapa de teste da tarefa auxiliar, o

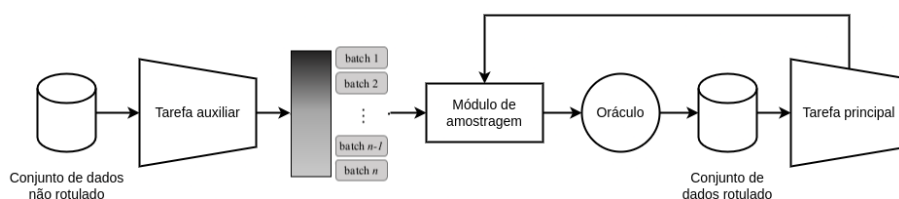


Figura 3. Fluxo de funcionamento do algoritmo. Adaptado do artigo original

Processador	RAM (GB)	GPU	SO	CUDA
AMD Ryzen 9 5940X	128	2x RTX 3080	Ubuntu 22.04.3 LTS	12.0
AMD EPYC 7551P	64	1x RTX 3070	Ubuntu 22.04.3 LTS	12.1
AMD EPYC 7453	32	1x RTX 3080	Ubuntu 22.04.3 LTS	12.1

Tabela 1. Hardware utilizado nos experimentos

conjunto de dados não rotulados inteiro é usado novamente e a tarefa calcula a *loss* de cada imagem, denotando seu grau de dificuldade.

As tarefas auxiliares escolhidas pelos autores são a estimação de rotação de uma imagem, solução de quebra cabeça com pedaços de uma imagem, colorização de imagens preto-e-branco e uma rede SimSiam. As tarefas auxiliares não são necessariamente relacionadas a tarefa principal em seu escopo, visto que nenhuma das tarefas mencionadas é intrinsecamente relacionada com classificação e segmentação. No entanto, os autores estimam uma alta correlação entre as *losses* das tarefas auxiliares e da tarefa principal, em diferentes conjuntos de dados, demonstrando assim que o desempenho da tarefa auxiliar está relacionada ao da tarefa principal e a dificuldade de uma imagem indica seu potencial ganho de informação.

As imagens são ordenadas em ordem decrescente de acordo com sua *loss* e divididas em batches. A ordenação dos dados auxilia na seleção de dados desafiadores, enquanto a separação em batches distribui os dados de diferentes classes, auxiliando na representatividade da amostragem.

Assim, a cada ciclo de AL, a tarefa principal escolhe as primeiras imagens de um batch para utilizar em seu treinamento e o processo de criação dos batches garante que as imagens escolhidas foram as mais desafiadoras de acordo com a tarefa auxiliar. O número máximo de ciclos de treinamento corresponde a quantidade de batches e a tarefa principal pode ser treinada até todos os ciclos serem realizados ou até as curvas de treinamento se estabilizarem.

3.3. Protocolo Experimental

Para a elaboração do presente trabalho realizamos 5 experimentos. Todo o código foi escrito em Python, com a utilização da biblioteca PyTorch para o treinamento dos modelos. Os experimentos foram realizados em três computadores diferentes, com as configurações contidas na tabela 1.

Os experimentos que utilizam AL foram iniciados com 250 imagens, amostradas do primeiro batch, e tiveram um incremento de 250 imagens dos demais batches ao conjunto de dados rotulados a cada ciclo de treinamento. Um ciclo corresponde ao treinamento da tarefa principal por 200 épocas. As tarefas auxiliares geram batches de 500 imagens, para serem usadas no treinamento da tarefa principal, totalizando 2940 batches. Devido ao tempo necessário por ciclo de treinamento e à estabilização das curvas, o treinamento da tarefa principal foi interrompido antes do último batch, em todos os experimentos com AL.

Para o primeiro experimento, foi treinada uma rede de classificação de imagens no conjunto de dados inteiro. O intuito desse experimento foi definir um desempenho base para determinar se os experimentos com AL apresentam algum ganho. O treinamento foi

realizado por 200 épocas e utilizou uma ResNet18 pré treinada como backbone. Todos os modelos de classificação de imagens utilizados nos experimentos têm como backbone uma ResNet18.

No segundo experimento, treinamos a tarefa principal em ciclos, similarmente aos experimentos com AL, com dados amostrados de maneira aleatória. Tanto as imagens iniciais quanto o incremento do conjunto de dados rotulados a cada ciclo foram amostradas aleatoriamente sem reposição do conjunto de dados não rotulados. O intuito desse experimento foi definir um desempenho base para a tarefa principal, quando treinada com dados da tarefa auxiliar.

O terceiro experimento faz uso de AL com uma rede de predição de rotação como tarefa auxiliar. Essa tarefa é auto supervisionada e consiste na aplicação de uma rotação aleatória a uma imagem. As possíveis rotações são 0 graus (nenhuma rotação), 90, 180 e 270 graus. A imagem e o rótulo da rotação são então alimentados à uma rede de classificação, que tenta prever a rotação da imagem. Todo o conjunto de dados não rotulados é usado no treinamento desse modelo, que é testado nesses mesmos dados, armazenando a loss de cada uma das imagens. Como mencionado na seção anterior, as imagens são ordenadas pela loss e divididas em batches, antes de serem usadas no treinamento da tarefa principal.

No primeiro ciclo de AL, a tarefa principal amostra os dados do primeiro batch uniformemente. Nos ciclos subsequentes, as primeiras 250 imagens do batch são amostradas e usadas no treinamento da tarefa principal.

O quarto experimento consiste de uma tarefa auto supervisionada de colorização de imagens. A imagem de entrada é convertida para grayscale e passada para a rede de colorização, juntamente com a imagem original como rótulo. A rede estima a cor de cada pixel e compara a imagem resultante do processo com a label, usando *Mean Squared Error* (MSE). Diferentemente dos modelos de classificação, o modelo de colorização utiliza uma arquitetura específica para o problema.

Além das tarefas mencionadas, testamos a resolução de quebra-cabeças como um tarefa auxiliar. A tarefa consiste em dividir cada imagem do conjunto de dados não rotulados em quatro peças, embaralhar as peças de acordo com uma de suas possíveis permutações e alimentá-las para uma rede de classificação de imagens, que deve tentar determinar a ordem correta. No entanto, a rede de classificação não convergiu, tendo o desempenho equivalente a uma escolha aleatória. Acreditamos que isso se deva a perda de informação. As imagens com dimensão 32x32 pixels são divididas em quatro imagens com dimensão 16x16. Para serem passadas pela rede, as imagens precisam ser redimensionadas de volta para 32x32, o tamanho mínimo da entrada de uma ResNet18. Visto que as mamonas não têm um contorno distinto como os objetos de interesse em conjuntos de dados sintéticos como o CIFAR10, pode ser que a informação na imagem após redimensionamento e convoluções não seja suficiente para que a rede aprenda.

Para o último experimento, utilizamos uma pseudo tarefa, que simula uma tarefa auxiliar ideal, que se sobressaísse para imagens de mamonas e criasse batches com alta concentração de imagens da classe. Para isso, criamos manualmente batches com 100 imagens de mamonas e 400 imagens de fundo. Criamos batches apenas com imagens do fundo após esgotar as imagens de mamonas. Os batches foram então utilizados normal-

$$\text{Acurácia} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}} \quad (1)$$

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}} \quad (2)$$

$$\text{Revocação} = \frac{\text{VP}}{\text{VP} + \text{FN}} \quad (3)$$

$$\text{F1 score} = 2 \times \frac{\text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}} \quad (4)$$

Figura 4. Equações da acurácia, precisão, revocação e F1 score

mente no treino do classificador com AL.

4. Resultados e Discussão

4.1. Análise Quantitativa

As métricas de teste para o classificador de referência e os classificadores treinados com base nas tarefas auxiliares são apresentadas nas Tabelas 2 e 3. Alguns dos modelos apresentaram instabilidade durante o treinamento, então dividimos as métricas em duas tabelas, com o intuito de obter uma análise mais detalhada. A tabela 2 contém as métricas ao fim do treinamento, enquanto a tabela 3 contém a média das métricas de todos os ciclos de treino. Na tabela 3, a média das métricas da referência correspondem as épocas, e não aos ciclos, visto que a referência foi treinada apenas uma vez.

As equações 1, 2, 3 e 4 formulam as métricas discutidas neste trabalho. VP representa a quantidade de imagens de mamonas classificadas corretamente, ou verdadeiro positivos, VN representa a quantidade de imagens de fundo classificadas corretamente, ou verdadeiro negativos, FP representa a quantidade de imagens de fundo classificadas como mamonas, ou falso positivos e FN representa a quantidade de imagens de mamonas classificadas como fundo, ou falso negativos.

Dado o desbalanceamento de classes, se o modelo sempre predizer a classe das imagens como sendo plano de fundo, a acurácia (equação 1), que define a taxa de acerto em relação a todas as predições feitas, é de aproximadamente 97%. Dessa forma, uma métrica que combine ambas as predições para mamonas e plano de fundo não fornece informações relevantes sobre as mamonas. Por essa razão, as tabelas contêm apenas as métricas da classe mamona. Para analisar os resultados, utilizamos as métricas precisão (equação 2) e a revocação (equação 3). Calculamos também o F1 score (equação 4), utilizado apenas como um meio de estimar o balanço entre a precisão e a revocação.

Idealmente, a tarefa auxiliar utilizada na criação dos batches para treinamento com active learning deveria ter um desempenho superior para imagens de uma das classes. A

	Precisão	Revocação	F1 score
Referência	0.80	0.93	0.86
Amostragem Aleatória	0.92	0.79	0.85
Rotação	0.93	0.70	0.80
Colorização	0.93	0.48	0.63
Tarefa Genérica	0.79	0.92	0.85

Tabela 2. Precisão, revocação e F1 score ao fim do treinamento

	Precisão	Revocação	F1 score	No. de ciclos
Referência	0.81 ± 0.18	0.68 ± 0.19	0.70 ± 0.15	*200 épocas
Amostragem Aleatória	0.89 ± 0.13	0.60 ± 0.18	0.71 ± 0.17	74
Rotação	0.88 ± 0.09	0.59 ± 0.10	0.70 ± 0.09	74
Colorização	0.71 ± 0.23	0.39 ± 0.07	0.49 ± 0.11	67
Tarefa Genérica	0.66 ± 0.15	0.86 ± 0.07	0.74 ± 0.13	42

Tabela 3. Precisão, revocação e F1 score médios ao longo dos ciclos de treino

tarefa de rotação se sobressai em imagens com linhas e formas que pareçam deslocadas quando a imagem é rotacionada. No conjunto de dados utilizado nesse trabalho, as rotações testadas podem ocorrer naturalmente, visto que as plantas não são objetos orientados. Isso faz com que a loss das imagens seja similar para todo o conjunto de dados, de forma que os batches são efetivamente criados com imagens aleatórias. Isso pode ser observado na tabela 4.

A colorização teve o pior desempenho geral, embora seja a tarefa não artificial com a maior concentração de mamonas por batch, indicando que as características das imagens selecionadas não ajudam o modelo a aprender a detectar as mamonas.

A amostragem aleatória e a tarefa de rotação apresentaram uma precisão de aproximadamente 90%, superando a referência e as demais tarefas. Isso indica que, das imagens classificadas como mamona, 90% de fato contém mamonas.

Entretanto, a revocação de ambas as tarefas é superior apenas à da colorização. A revocação indica quantas das mamonas no conjunto de dados a tarefa principal foi capaz de identificar. Os frutos de mamonas podem lançar sementes a uma longa distância, portanto, uma única mamona não removida pode causar uma nova infestação, tornando crucial a remoção do maior número possível de mamonas na plantação. Por esse motivo, a revocação é mais significativa que a precisão na detecção de mamonas. A referência e a tarefa genérica tiveram uma revocação superior às demais tarefas.

O classificador baseado na tarefa genérica teve o melhor desempenho com base

	Mamonas	Background
Amostragem Aleatória	7.68 ± 2.64	242.32 ± 2.64
Rotação	8.49 ± 3.55	241.51 ± 3.55
Colorização	17.34 ± 7.01	232.66 ± 7.01
Tarefa Genérica	53.05 ± 6.56	196.95 ± 6.56

Tabela 4. Distribuição de classes por amostra de 250 imagens

na revocação, sendo treinado por apenas 42 ciclos. A quantidade de imagens utilizadas no treinamento desse classificador corresponde a menos de 1% do total de imagens no conjunto de dados, reduzindo consideravelmente o custo de rotulação dos dados.

4.2. Análise Qualitativa

Analisando visualmente os erros cometidos por cada classificador, não foi possível encontrar um padrão de erro intrinsecamente atrelado a uma tarefa auxiliar específica. Por essa razão, analisamos erros recorrentes em todos os classificadores.

Entre as predições incorretas, um dos cenários recorrentes ocorre quando as mamonas estão próximas a borda e a maior parte do objeto está fora da imagem. Exemplos disso podem ser vistos na figura 5. Isso provavelmente se deve a falta de informação, visto que as imagens são pequenas e passam por uma série de convoluções.



Figura 5. Predições incorretas próximas a borda

Os classificadores também tiveram dificuldade em identificar mamonas em meio a outras plantas (figura 6). Nesse cenário, a mamona e o plano de fundo tem a mesma cor, e suas bordas não são facilmente identificáveis. Similarmente, quando uma mamona ocupa a imagem inteira, a tarefa de identificação normalmente falha, uma vez que a maior parte da borda está fora da imagem (figura 7).

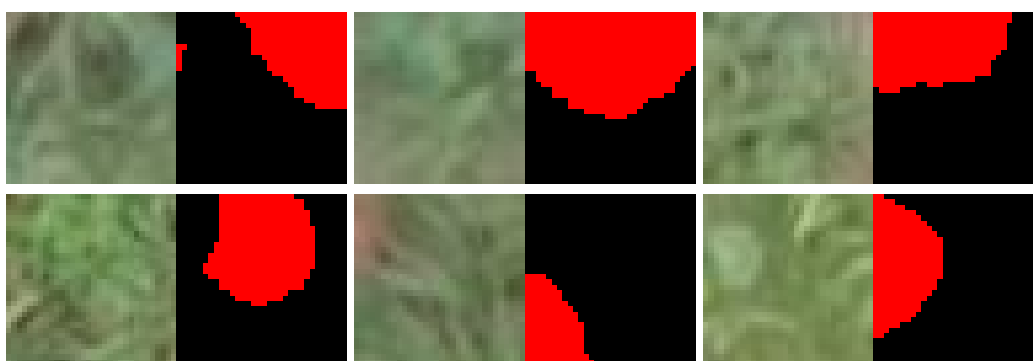


Figura 6. Predições incorretas no meio da cultura

5. Conclusão

Dada a natureza do conjunto de dados e a baixa quantidade de informação contida nas imagens, tanto o modelo de referência quanto os modelos treinados com active learning

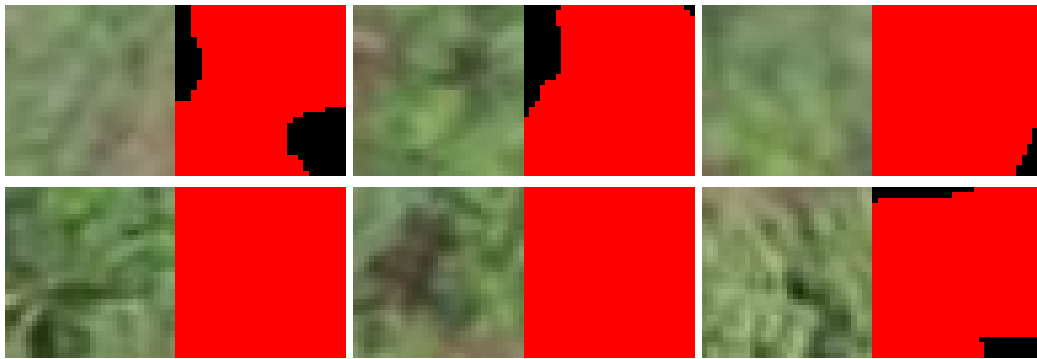


Figura 7. Predições incorretas ocupando a imagem inteira

tiveram uma boa performance. Os resultados obtidos apontam para a eficácia do active learning aplicado à identificação de plantas daninhas em meio a cultura, reduzindo o custo de rotulagem dos dados e aumentando o desempenho dos modelos.

O desbalanceamento extremo entre as classes dificulta o processo de escolha de tarefas auxiliares, uma vez que a tarefa deve forçar o aumento da concentração da classe sub representada nos batches para manter um número significativo de exemplos.

A colorização apresentou o pior desempenho entre a referência e as demais tarefas. Isso indica que as cores extraídas de imagens RGB podem não ser uma característica adequada para a detecção de mamonas, visto que a mamona pode facilmente ser confundida com a cultura, como foi observado na análise dos resultados.

A amostragem aleatória e a rotação apresentaram uma alta precisão, ao custo da redução da revocação em relação ao classificador de referência. Isso não é ideal em um cenário real, uma vez que os classificadores identificam menos mamonas. A revocação mais alta foi obtida pela tarefa genérica e, baseado nesses resultados, a extensão mais evidente deste trabalho é a avaliação de novas tarefas auxiliares, que gerem batches com uma maior concentração de imagens de mamonas.

Observamos que as dimensões das imagens utilizadas no trabalho limitam severamente a quantidade de informação disponível para o treino dos modelos. Além disso, o trabalho utiliza uma Resnet18 como backbone, que possui uma arquitetura rasa e pode impactar negativamente o desempenho da classificação. Com essas alterações, esperamos, esperamos uma melhora nos resultados obtidos, tanto nos classificadores como nas tarefas auxiliares.

Além disso, o trabalho tem foco na apenas na detecção de mamonas, embora diferentes variedades de daninhas possam estar presentes na plantação. Com o aumento do número de classes, esperamos ter um modelo mais generalizável e robusto.

Por fim, outra abordagem possível para trabalhos futuros é, a cada ciclo, realizar a inferência dos dados não rotulados utilizando o classificador treinado no ciclo anterior, separando os dados não rotulados em classes. Após a separação, executamos as tarefas auxiliares para cada classe, garantindo assim que uma quantidade significativa de imagens de mamonas seja amostrada.

Referências

- Andújar, D., Weis, M., and Gerhards, R. (2012). An ultrasonic system for weed detection in cereal crops. *Sensors*, 12(12):17343–17357.
- Aparna, M. and Radha, D. (2019). Detection of weed using visual attention model and svm classifier. In *Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)*, pages 243–253. Springer.
- Bhargava, B., Deshmukh, A., and Narasimhadhan, A. (2022). Modulation and signal class labelling with active learning and classification using machine learning. In *2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–5. IEEE.
- Cacciarelli, D. and Kulahci, M. (2023). Active learning for data streams: a survey. *Machine Learning*, 113(1):185–239.
- Cândido, M. J. D., Bomfim, M. A. D., Severino, L. S., and de Oliveira, S. Z. R. (2008). Utilização de coprodutos da mamona na alimentação animal. In: CONGRESSO BRASILEIRO DE MAMONA, 3., 2008, Salvador. Energia e
- Chen, Y., Wu, Z., Zhao, B., Fan, C., and Shi, S. (2020). Weed and corn seedling detection in field based on multi feature fusion and support vector machine. *Sensors*, 21(1):212.
- Costa, N. V., Rodrigues-Costa, A. C. P., Ferreira, S. D., de Araujo Barbosa, J., et al. (2018). Mã© todos de controle de plantas daninhas em sistemas orgânicos: breve revisã o. *Revista Brasileira de Herbicidas*, 17(1):25–44.
- da Silva Fonseca, N. B. and Soto-Blanco, B. (2014). Toxicidade da ricina presente nas sementes de mamona. *Semina: Ciências Agrárias*, 35(3):1415–1424.
- de Oliveira, R. C. and Felizari, J. (2011). Toxicidade de extratos de ricinus communis e euphorbia milii para caramujo aquático e tilápia. *Revista Thêma et Scientia*, 1(2).
- Ferreira, A. d. S. (2017). Redes neurais convolucionais profundas na detecção de plantas daninhas em lavoura de soja.
- Ferreira, M. D., Malyska, M., Sahar, N., Miotto, R., Paulovich, F., and Milios, E. (2021). Active learning for medical code assignment.
- Katragadda, S. P., Cody, T., Beling, P., and Freeman, L. (2022). Active learning with combinatorial coverage. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1129–1136. IEEE.
- LIMA, R., Severino, L. S., Albuquerque, R. C., BELTRÃO, N. d. M., and Sampaio, L. R. (2008). Casca e torta de mamona avaliados em vasos como fertilizantes orgânicos.
- Lin, F., Zhang, D., Huang, Y., Wang, X., and Chen, X. (2017). Detection of corn and weed species by the combination of spectral, shape and textural features. *Sustainability*, 9(8):1335.
- Meinke, K. (2021). Active machine learning to test autonomous driving. In *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 286–286. IEEE.

- Molina-Villa, M. A., Solaque-Guzmán, L. E., et al. (2016). Machine vision system for weed detection using image filtering in vegetables crops. *Revista Facultad de Ingeniería Universidad de Antioquia*, (80):124–130.
- Oda, Y. S. and Castro Neto, J. C. d. (2023). Detecção de plantas daninhas e plantas de soja utilizando imagens multiespectrais e visão computacional.
- Perez, A., Lopez, F., Benlloch, J., and Christensen, S. (2000). Colour and shape analysis techniques for weed detection in cereal fields. *Computers and electronics in agriculture*, 25(3):197–212.
- Pham, T.-N., Nguyen, V.-H., Kwon, K.-R., Kim, J.-H., and Huh, J.-H. (2024). Improved yolov5 based deep learning system for jellyfish detection. *IEEE Access*.
- Punithavathi, R., Rani, A. D. C., Sughashini, K., Kurangi, C., Nirmala, M., Ahmed, H. F. T., and Balamurugan, S. (2023). Computer vision and deep learning-enabled weed detection model for precision agriculture. *Comput. Syst. Sci. Eng.*, 44(3):2759–2774.
- Saadaoui, E., Martín, J. J., Tlili, N., and Cervantes, E. (2017). Castor bean (*ricinus communis* L.) diversity, seed oil and uses. *Oilseed Crops: Yield and Adaptations under Environmental Stress*, pages 19–33.
- Settles, B. (2009). Active learning literature survey.
- Shahbazi, N., Ashworth, M. B., Callow, J. N., Mian, A., Beckie, H. J., Speidel, S., Nicholls, E., and Flower, K. C. (2021). Assessing the capability and potential of lidar for weed detection. *Sensors*, 21(7):2328.
- Tummapudi, S., Sadhu, S. S., Simhadri, S. N., Damarla, S. N. T., and Bhukya, M. (2023). Deep learning based weed detection and elimination in agriculture. In *2023 International Conference on Inventive Computation Technologies (ICICT)*, pages 147–151. IEEE.
- Wang, N., Zhang, N., Dowell, F. E., Sun, Y., and Peterson, D. E. (2001). Design of an optical weed sensor using plant spectral characteristics. *Transactions of the ASAE*, 44(2):409.
- Yi, J. S. K., Seo, M., Park, J., and Choi, D. (2022). Using self-supervised pretext tasks for active learning. *CoRR*, abs/2201.07459.
- Zera, F. S., Azania, C. A. M., Schiavetto, A. R., Lorenzato, C. M., Freitas, G. B., and Azania, A. A. d. P. M. (2011). Tolerância de mamona (*ricinus communis*) a herbicidas utilizados na cultura da cana-de-açúcar. *Nucleus*, 8(1):1–10.
- Zhao, Y., Zhang, R. H., Zhou, S., and Zhang, Z. (2020). Active learning approaches to enhancing neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1796–1806.