

# VULCAN - Uma Camada Visual para Jupyter Notebook para o Ensino de Classificação de Imagem

Ramon Mayor Martins  
Departamento de Informática e Estatística  
Universidade Federal de Santa Catarina  
Florianópolis - SC - Brasil  
ramon.mayor@posgrad.ufsc.br

Christiane Gresse von Wangenheim  
Departamento de Informática e Estatística  
Universidade Federal de Santa Catarina  
Florianópolis - SC - Brasil  
c.wangenheim@ufsc.br

Augusto Cesar Medeiros Franz  
Departamento de Informática e Estatística  
Universidade Federal de Santa Catarina  
Florianópolis - SC - Brasil  
gutofranz@gmail.com

Jean Carlo Rossa Hauck  
Departamento de Informática e Estatística  
Universidade Federal de Santa Catarina  
Florianópolis - SC - Brasil  
jean.hauck@ufsc.br

## ABSTRACT

As Machine Learning (ML) is becoming omnipresent, it is important to popularize ML knowledge starting in high school. At this educational stage ML is typically taught by teaching students to develop ML models for image classification adopting visual environments. For this purpose a user-friendly visual layer of Jupyter Notebook was developed, that provides comprehensive support for all process steps involved in developing image classification models. Results of a preliminary evaluation with teachers and students provide a first indication that it can be a useful, functional, performance-efficient, and usable support for teaching the development of image classification models to novices and ease the transition to conventional environments later on.

## KEYWORDS

Image classification, High school, Machine Learning, Visual tool, Jupyter Notebook, Google Colab.

## 1 Introdução

Devido à crescente importância do Aprendizado de Máquina (AM), é importante permitir que os jovens se tornem usuários responsáveis e criadores de soluções inteligentes desde a educação básica [38,28]. Seguindo as diretrizes curriculares, isso deve abordar uma compreensão dos conceitos básicos de AM, tais como algoritmos de aprendizagem e fundamentos de redes neurais, como também a aprendizagem profunda [38,26]. Os estudantes também devem se tornar capazes de aplicar um processo de AM centrado no ser humano, incluindo a preparação de dados, treinamento, avaliação de desempenho e previsão [17]. Adotando principalmente estratégias ativas de aprendizagem, os estudantes são tipicamente instruídos

primeiro a desenvolver um modelo pré-definido de AM e depois orientados a desenvolver suas próprias aplicações de AM [26,38]. O emprego de um processo iterativo de desenvolvimento de AM tem demonstrado vários benefícios, pois permite aos usuários (mesmo aqueles sem conhecimento em programação ou experiência em AM) construir e refinar rapidamente modelos de AM. Este processo envolve os estudantes em um ciclo de modificação iterativa de dados, treinamento de modelos e/ou parâmetros para melhorar o desempenho do modelo desenvolvido [1, 2, 17]. Geralmente são utilizadas linguagens de programação baseadas em texto, tal como Python, executadas no ambiente Jupyter Notebook [22] rodando no Google Collaboratory (Google Colab) [14,13]. Fornecendo um ambiente livre, interativo e colaborativo para desenvolver modelos de AM na nuvem, o ambiente Jupyter/Colab já é amplamente aplicado na educação STEM<sup>1</sup> e no ensino de AM no ensino médio [4,16,28], bem como na prática profissional. O ambiente Jupyter/Colab oferece uma ferramenta para orientar os estudantes, desde a visualização passiva do conteúdo do curso até a exploração, análise e criação de artefatos de maneira ativa. Além disso, o Google Colab, suporta tanto o uso de GPU (unidade de processamento gráfico) como TPU (unidade de processamento de *Tensor*) e permite aos usuários colaborar e executar código inteiramente na nuvem usando os recursos do Google (Google Cloud). Isto permite treinar modelos de AM mesmo sem acesso a uma máquina de alto desempenho ou acesso à internet de alta velocidade, o que torna esse ambiente adequado para fins educacionais, além de permitir que os estudantes trabalhem colaborativamente em um projeto [23]. No entanto, o desenvolvimento de modelos de AM usando uma linguagem de programação textual requer a compreensão

<sup>1</sup> Ciência, Tecnologia, Engenharia e Matemática

dos conceitos de programação e de sua sintaxe [29]. Isto pode dificultar o aprendizado dos conceitos de AM, especialmente para iniciantes na educação básica [36]. Uma alternativa pode ser a adoção de ferramentas visuais a fim de auxiliar os estudantes a reduzir a carga cognitiva e evitar erros [12]. Devido a estas vantagens, os ambientes visuais estão sendo amplamente adotados na introdução de programação ao longo das etapas educacionais [6], bem como para o ensino de AM [16]. Portanto, este artigo apresenta uma camada visual para o Jupyter Notebook usando a extensão Ipywidgets [19] para o ensino de AM, especificamente para a classificação de imagens para iniciantes no ensino médio. Também apresenta uma avaliação preliminar da qualidade da camada visual por meio de um painel de representantes do público alvo.

## 2 Trabalhos Relacionados

Com base no resultado de um mapeamento sistemático da literatura [16], foram identificadas várias ferramentas visuais para suporte de desenvolvimento interativo de modelos de AM como parte da educação de AM na educação básica (K-12), voltado principalmente a iniciantes. A maioria dessas ferramentas é direcionada principalmente para classificação de imagem, texto, fala ou movimento.

Os tipos de ferramentas visuais para suporte de desenvolvimento de modelos de AM vão desde o suporte baseado em blocos, ampliando o respectivo ambiente de programação por meio de blocos de AM específicos para preparação, treinamento e avaliação de dados até ferramentas baseadas em fluxo de dados que permitem ao usuário construir a arquitetura da rede neural. Contudo, a maioria, como por exemplo, Google Teachable Machine [15], PIC [37] e LearningML [34], adotam uma abordagem orientada ao fluxo de trabalho que suporta o desenvolvimento do modelo de AM, guiando o usuário passo a passo por meio de um navegador web ou interface visual. Desta forma, ferramentas orientadas ao fluxo de trabalho também facilitam a compreensão conceitual do processo de AM e podem reduzir a carga cognitiva para os iniciantes, ao mesmo tempo que lhes permite experimentar o desenvolvimento de seus modelos de AM para transferir competências aprendidas com tutoriais guiados para seus próprios problemas.

Geralmente as ferramentas suportam todas as etapas básicas de um processo de desenvolvimento interativo de AM. Eles normalmente incentivam a coleta de pequenas quantidades de dados e sua rotulação. Então, esses dados são usados para treinar um modelo de AM por meio do

aprendizado por transferência, o que permite construir modelos precisos com base em pequenos conjuntos de dados e com pouco tempo para o treinamento, usando diversas plataformas, como Pytorch, Keras, ou Tensorflow. Uma vez que o modelo é treinado, seu desempenho pode ser avaliado. Isto é realizado principalmente permitindo ao usuário testar o modelo com novos dados para os quais o modelo fornece como saída o rótulo previsto [9, 34]. Algumas ferramentas também permitem visualizar métricas de desempenho, tais como precisão e função de perda [9, 37]. A maioria das ferramentas também suporta a exportação do modelo criado para sua implantação diretamente para um ambiente de programação baseado em blocos (Scratch ou App Inventor) ou em diversos formatos (como Tensorflow.js, Python, etc.) permitindo a implantação dos modelos de AM criados como parte de jogos ou aplicativos móveis. No entanto, nenhuma das ferramentas suporta a análise de requisitos.

Estas ferramentas orientadas a fluxo de trabalho que guiam o desenvolvimento de modelos de AM são atualmente projetados para ambientes específicos de programação na educação básica, como LearningML, Cognimates, ou mblocks para Scratch; PIC, PAC, ou ML4K para App Inventor; ou Google Teachable Machine permitindo uma implantação via JavaScript. No entanto, o suporte na forma de interface visual para ambientes mais convencionais como o Jupyter Notebook é escasso. E embora o Jupyter Notebook também tenha sido usado para suportar fluxos de trabalho em outras áreas, tais como processos científicos reproduzíveis [7], não foi encontrada nenhuma camada visual para este ambiente a ser usado na educação de AM na educação básica.

## 3 Metodologia de Pesquisa

Visando o desenvolvimento de uma camada visual para facilitar o aprendizado da aplicação dos conceitos de AM, foi adotada uma estratégia de pesquisa multimétodo.

**Análise do estado da arte.** A fim de fornecer uma visão geral do atual estado da arte sobre os ambientes visuais de AM, realizamos um estudo sistemático de mapeamento seguindo o procedimento proposto por Petersen et al. [30]. Na fase de definição, foram definidas as questões de pesquisa e o protocolo de revisão, assim como as fontes de dados, a estratégia de busca, as *strings* de busca e os critérios de inclusão/exclusão. A fase de execução foi realizada seguindo o protocolo de revisão realizando as buscas nos repositórios especificados. Os resultados iniciais da busca foram analisados quanto à sua relevância, aplicando os critérios de inclusão/exclusão. Uma vez

identificados os estudos relevantes, foram extraídos dados com relação às questões de pesquisa. Com base nos dados extraídos, os estudos encontrados foram analisados e os resultados foram sintetizados. Os resultados detalhados desta revisão são apresentados em [16] e resumidos na Seção 2. **Design da camada visual do Jupyter Notebook centrado no ser humano.** Seguindo uma abordagem de design centrado no ser humano [20], foram realizadas as seguintes atividades: (i) *Caracterização do contexto de uso.* Como parte da análise do contexto, foram analisados o público-alvo e as necessidades em relação ao respectivo nível educacional. O processo de desenvolvimento de AM foi analisado e ajustado para fins educacionais nesta etapa educacional, identificando o objetivo e as tarefas. Também foi analisado o ambiente em termos das características da plataforma alvo (Jupyter Notebook e Google Colab). (ii) *Especificação dos requisitos do usuário.* Os requisitos funcionais e não funcionais da camada visual foram especificados com base na análise do contexto. (iii) *Desenvolvimento da camada visual.* Analisando alternativas tecnicamente viáveis para o desenvolvimento de uma camada visual para o Jupyter Notebook no Google Colab, foi adotada uma abordagem iterativa e incremental de desenvolvimento de software [24]. Inicialmente foi desenvolvido o suporte de software especificando o conteúdo do projeto da interface do usuário e, em seguida, desenvolvendo protótipos da interface do usuário que foram concluídos incrementalmente com base no *feedback* informal de especialistas de AM até contemplar todo o processo de desenvolvimento de AM. Em seguida, foram realizados testes do sistema e, com base nos resultados, foram implementadas e testadas as correções e melhorias necessárias. **Avaliação da camada visual.** Visando uma avaliação preliminar da camada visual, foi realizado um estudo de usuário. De acordo com Wohlin et al. [39] usando GQM (*Goal/Question/Metric*) [5], o objetivo da avaliação, questões e métricas foram definidas, e um questionário para operacionalizar a coleta de dados foi projetado. Durante a execução, os participantes testaram a camada visual proposta e responderam o questionário de pós-teste. Os dados coletados foram analisados utilizando estatística descritiva e os resultados foram interpretados e discutidos.

## 4 Visual Jupyter para Classificação de Imagens

O objetivo desta pesquisa é desenvolver uma camada visual baseada na web que permita aos estudantes de ensino médio desenvolver modelos de classificação de imagens em Jupyter Notebook por meio do Google Colab sem necessidade de codificação. Esta ferramenta deve ser

usada como parte introdutória do ensino de AM preparando e facilitando uma posterior transição para a programação baseada em texto em Python, de uma forma mais convencional.

### 4.1 Análise de Contexto

Os usuários alvo são estudantes de ensino médio entre 15 e 18 anos de idade. Nesta fase educacional, os estudantes já adquiriram competências básicas em matemática, linguagem, ciências, etc. No entanto, normalmente o conhecimento em línguas estrangeiras como o inglês pode ser limitado, exigindo uma opção de interface de usuário no idioma nativo (português brasileiro). Até o momento, como nem computação, nem IA são parte explícita do currículo escolar nacional, pode-se considerar que a maioria dos estudantes não têm conhecimento prévio sobre programação ou AM. A maioria das escolas que atendem essa etapa educacional, está equipada com laboratórios de informática, dispendo de computadores com conexão à Internet. Especialmente motivado pela situação pandêmica, muitos estudantes agora também têm acesso a um computador ou notebook com conexão à Internet em casa. Contudo, a configuração típica desses computadores não conta com uma GPU dedicada, necessário para realizar o treinamento de um modelo de AM na máquina local. Com base nas diretrizes curriculares AI4K12 [38, 26] e CSTA K-12 [10], os objetivos de aprendizagem visam tornar os estudantes a serem capazes de aplicar conceitos básicos de AM relacionados com o desenvolvimento de modelos de aprendizagem profunda. Concentrou-se no apoio à tarefa específica de classificação de imagem, normalmente ensinada como uma das primeiras aplicações [28]. Neste contexto, adotando o ciclo *use-modify-create* [27], os estudantes normalmente desenvolvem primeiro um modelo pré-definido de AM, que depois é modificado até que sejam direcionados a desenvolver seu próprio modelo de classificação de imagem em qualquer tipo de domínio de aplicação. A aplicação dos conceitos de AM é tipicamente ensinada pela adoção de um processo de AM interativo centrado no ser humano, derivado dos processos de AM interativo comumente propostos [1, 2] que foi ajustado e simplificado para manter a carga cognitiva apropriada para iniciantes em AM (Figura 1).

### 4.2 Requisitos

O principal requisito funcional da camada visual é fornecer suporte para o estudante desenvolver interativamente um modelo de AM para classificação de imagens no Jupyter Notebook/Google Colab. Nesse sentido, a camada visual precisa dar suporte a todo o processo de desenvolvimento

## XIV Computer on the Beach

30 de Março a 01 de Abril de 2023, Florianópolis, SC, Brasil

da AM, conforme ilustrado na Figura 1. Apenas a implantação do modelo, executado em plataformas alvo, está fora do escopo. A fim de permitir a utilização do suporte on-line e o uso gratuito de recursos computacionais

via Google Colab para treinamento do modelo de AM, a camada visual precisa ser totalmente integrada ao Jupyter Notebook e ser executável no Google Colab.



Figura 1: Processo iterativo de desenvolvimento de AM [17].

A ferramenta também deve permitir a conexão com o Google Drive para permitir o *upload* do conjunto de dados previamente separado em pastas para cada categoria. Além disso, durante a etapa de previsão, também deve ser possível fazer o *upload* de novas imagens para testar a classificação. Imagens de diferentes tamanhos e formatos devem ser aceitas e tratadas para serem uniformizadas. A camada visual deve estar disponível no idioma de Português do Brasil como também em inglês a fim de permitir aplicações para um público alvo mais amplo.

Visando principalmente os iniciantes, o desenvolvimento da camada visual também deve considerar os princípios de design propostos por Resnick et al. [33] e de aplicações de AM de forma flexível propostas por Sun et al. [35].

### 4.3 Arquitetura da Camada Visual VULCAN

A implementação da camada visual consiste em um ambiente que roda em um Jupyter Notebook dentro do Google Colab (Figura 2). Os dados são carregados do Google Drive para gerenciar, visualizar e compartilhar o conjunto de dados facilmente. O modelo também é treinado dentro do ambiente do Google Colab.

Para a implementação dos elementos visuais, é utilizada a biblioteca Jupyter Widgets (extensão Ipywidgets) [19]. Esta extensão fornece *widgets* HTML interativos para Jupyter Notebook e o núcleo IPython, o que permite uma interação gráfica semelhante à GUI com Jupyter Notebook por meio de menus de seleção, barras deslizantes, botões de rádio ou caixas de texto [31]. Esses *widgets* gráficos podem ser incorporados ao Jupyter Notebook e os usuários podem combinar a habitual análise de scripts com a ativação de *widgets*. Os *widgets* gráficos podem ser usados, por exemplo, para permitir a variação de valores dos parâmetros de entrada e explorar um conjunto de dados ou resultados computacionais. Tais elementos podem ocultar o código baseado em texto e permitir a criação de um *notebook* com uma interface visual. Para a tarefa de classificação de imagens, foi utilizado o *framework* de

aprendizagem profunda fast.ai, sendo um *framework* de alto nível com uma das curvas de aprendizagem mais rápidas em comparação com outros [18]. Toda a camada visual foi desenvolvida em Python, para rodar nativamente no Colab e ter compatibilidade com os *frameworks* selecionados.

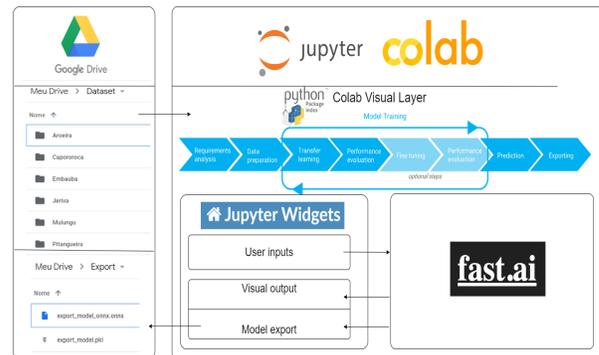


Figura 2: Arquitetura da camada visual.

De acordo com a análise de requisitos e a arquitetura definida seguindo um processo de AM interativo centrado no ser humano [17], as necessidades de entrada e saída de dados foram identificadas, especificando o suporte pela Camada Visual de Classificação (VULCAN) (Figura 3). A camada visual está disponível em português brasileiro e inglês. O pacote *vulcan-colab* é distribuído por meio do repositório Python *Package Index* (PyPI) [32] para torná-lo disponível também no Jupyter Notebook, ocultando os detalhes de implementação para os estudantes. O pacote está disponível em <https://pypi.org/project/vulcan-colab> e pode ser importado diretamente para qualquer *notebook* Colab. Todo o código desenvolvido está disponível no repositório de códigos da UFSC (<https://codigos.ufsc.br/gqs/colabvisuallayer>) sob a licença BSD 3-clause. Para orientar o uso da camada visual, também é compartilhado um exemplo de *notebook* em [bit.ly/3FWLcwe](https://bit.ly/3FWLcwe), assim como a documentação sobre cada

uma das etapas suportadas pela camada visual em <https://vulcan-colab.readthedocs.io/>.

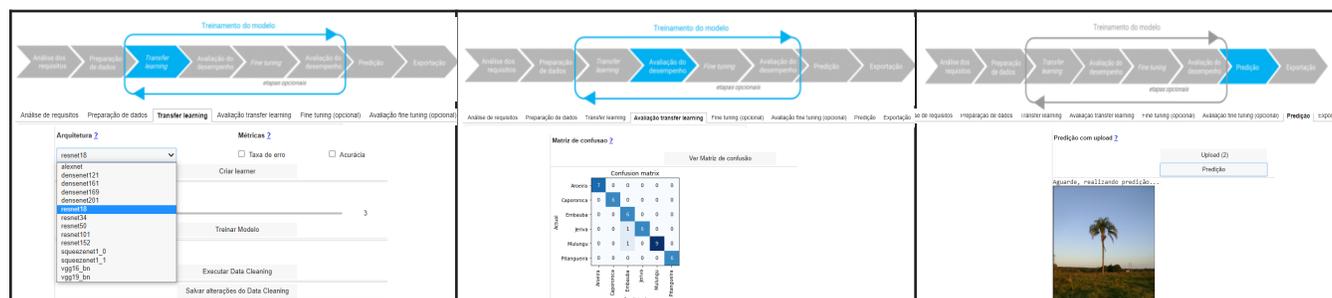


Figura 3: Exemplos de telas da camada visual VULCAN suportando o processo interativo de AM.

## 5 Avaliação Preliminar da Camada Visual

### 5.1 Definição e Execução da Avaliação

A fim de avaliar a qualidade da camada visual, foi realizada uma avaliação preliminar na forma de um estudo de usuário. O objetivo foi analisar a qualidade da camada visual em termos de utilidade, adequação funcional, eficiência de desempenho e usabilidade do ponto de vista dos professores e estudantes no contexto do ensino de computação. Com base em [20, 21, 11, 8], os fatores de qualidade a serem avaliados são decompostos e a coleta de dados é operacionalizada usando um questionário de pós-teste. Durante o estudo, os usuários recebem primeiro uma visão geral básica do objetivo e das características do VULCAN. Em seguida, eles desenvolvem um modelo de AM para a classificação de imagens de espécies de árvores, usando um conjunto de dados preparado, executando o processo completo de desenvolvimento de AM. A avaliação foi realizada por um total de 14 participantes, incluindo cinco professores do ensino médio, dez professores de computação no ensino superior e oito alunos do ensino superior. Considerando o foco principal no uso do VULCAN para o ensino médio, não foi possível incluir esses estudantes na avaliação devido à falta de

conhecimento de AM nesta etapa educacional atualmente. Portanto, foram convidados estudantes do ensino superior que, em sua maioria, tinham pelo menos alguma experiência prévia com AM. Os participantes foram selecionados usando uma abordagem de amostragem de conveniência dos membros da iniciativa Computação na Escola/INCoD/INE/UFSC com conhecimento prévio sobre o AM para classificação da imagem. Os participantes foram convidados via e-mail com explicação do objetivo da avaliação e assegurando sua confidencialidade. A participação foi voluntária. Instruções e formulários de coleta de dados foram disponibilizados on-line usando a ferramenta LimeSurvey. Foram convidados um total de 26 alunos e professores e obtivemos uma taxa de resposta de 54% (n=14).

### 5.2 Análise da Avaliação do Usuário

**Utilidade.** Todos os professores acharam a ferramenta útil para ensinar os iniciantes em cursos de graduação e 9 de 10 professores também acharam a ferramenta útil para ensinar AM no ensino médio. A maioria dos estudantes e professores também acham que a ferramenta pode ser mais útil para aprender a desenvolver um modelo de AM do que um ambiente convencional baseado em texto, exceto para um estudante que discorda (Tabela 1).

Tabela 1: Análise de utilidade.

Ponto de vista: Estudante/Professor (total de 14 respostas)					
	Discordo fortemente	Discordo	Nem concordo nem discordo	Concordo	Concordo fortemente
Eu preferiria começar a aprender o desenvolvimento de um AM	1	2	2	4	4
Acho que isso pode me ajudar a aprender conceitos de AM	1	4	5	4	0
VULCAN é útil para aprender AM	1	6	2	5	0
Acho que pode ser mais útil do que o convencional	2	3	5	4	0
Acho que pode ajudar os estudantes a aprender conceitos de AM	0	5	4	5	0
Acho VULCAN útil para ensinar AM para iniciantes na educação básica/graduação	0	5	5	4	0
Acho que VULCAN é útil para ensinar AM no ensino médio	1	0	5	4	0

**Adequação funcional.** A maioria dos participantes concorda que todas as etapas para o ensino do desenvolvimento de um modelo de classificação de imagem são suportadas pela ferramenta (Tabela 2). Dois participantes comentaram sobre a falta de suporte para

rotular manualmente o conjunto de dados, mas esta parte está fora do escopo do suporte fornecido pelo Jupyter Notebook. Poucos erros foram relatados, principalmente relacionados com a ordem de alguns *widgets* e a tradução dos termos de AM para o português brasileiro.

Tabela 2: Análise da adequação funcional.

Ponto de vista: Estudante/Professor (total de 14 respostas)	
Existem alguns passos/partes para o ensino do desenvolvimento da classificação de imagem para os iniciantes que não são contemplados pelo VULCAN?	<div style="display: flex; justify-content: space-between;"> <span>■ Sim (se sim, qual?)</span> <span>■ Não</span> </div> <div style="display: flex; align-items: center;"> <div style="width: 100px; height: 10px; background-color: #0070C0; margin-right: 5px;"></div> <span>2</span> <div style="width: 800px; height: 10px; background-color: #D9E1F2; margin-left: 5px;"></div> <span>12</span> </div>
Você notou algum erro em relação à funcionalidade do VULCAN?	<div style="display: flex; justify-content: space-between;"> <span>■ Sim (se sim, qual?)</span> <span>■ Não</span> </div> <div style="display: flex; align-items: center;"> <div style="width: 100px; height: 10px; background-color: #0070C0; margin-right: 5px;"></div> <span>3</span> <div style="width: 800px; height: 10px; background-color: #D9E1F2; margin-left: 5px;"></div> <span>11</span> </div>

**Eficiência do desempenho.** Todos participantes consideraram satisfatória a eficiência do desempenho, sendo capazes de executar todas as etapas rapidamente. Um participante, que aumentou o número de épocas de treinamento, observou que o treinamento levou mais tempo, mas também reconheceu que isto aconteceu como consequência dos parâmetros de treinamento escolhidos, independentemente da camada visual.

A maioria dos participantes utilizou o conjunto de dados preparado para a classificação, experimentando diferentes arquiteturas disponíveis, incluindo Alexnet, ResNet18, ResNet34, entre outras. (Tabela 3). A ferramenta também se mostrou eficiente, pois todos os usuários consideraram o tempo necessário para desenvolver o modelo como apropriado (Tabela 3). Onze participantes completaram a tarefa em 30 minutos, um tempo aceitável para um treinamento inicial de classificação de imagem como parte de um curso de AM. Além disso, 10 participantes consideraram que o tempo necessário para completar todo o processo é mais rápido do que utilizar um Jupyter *notebook* baseado em texto.

**Usabilidade.** Com exceção de um professor que não tinha experiência prévia no desenvolvimento de modelos de AM, todos os participantes foram capazes de completar a tarefa de desenvolvimento do modelo para a classificação de imagem de espécies de árvores.

Tabela 3: Análise de usabilidade.

Ponto de vista: Estudante/Professor (total de 14 respostas)	
Eficácia: Você foi capaz de desenvolver um modelo de AM para classificação de imagens com VULCAN?	<div style="display: flex; justify-content: space-between;"> <span>■ Sim</span> <span>■ Não</span> </div> <div style="display: flex; align-items: center;"> <div style="width: 100px; height: 10px; background-color: #0070C0; margin-right: 5px;"></div> <span>13</span> <div style="width: 10px; height: 10px; background-color: #D9E1F2; margin-left: 5px;"></div> <span>1</span> </div>
Eficiência: Você considera apropriado o tempo necessário para desenvolver o modelo de AM com VULCAN?	<div style="display: flex; justify-content: space-between;"> <span>■ Sim</span> <span>■ Não</span> </div> <div style="display: flex; align-items: center;"> <div style="width: 100px; height: 10px; background-color: #0070C0; margin-right: 5px;"></div> <span>14</span> </div>
Eficiência: Quanto tempo demorou para desenvolver um AM usando VULCAN?	<div style="display: flex; justify-content: space-between;"> <span>■ Menos de 30 minutos</span> <span>■ Mais de 30 minutos</span> </div> <div style="display: flex; align-items: center;"> <div style="width: 100px; height: 10px; background-color: #0070C0; margin-right: 5px;"></div> <span>12</span> <div style="width: 10px; height: 10px; background-color: #D9E1F2; margin-left: 5px;"></div> <span>2</span> </div>
Eficiência: Em comparação com uma versão de programação baseada em texto de um Jupyter <i>notebook</i> , você levou menos, igual ou mais tempo para desenvolver um AM para classificação de imagens?	<div style="display: flex; justify-content: space-between;"> <span>■ Sim</span> <span>■ Não</span> </div> <div style="display: flex; align-items: center;"> <div style="width: 100px; height: 10px; background-color: #0070C0; margin-right: 5px;"></div> <span>10</span> <div style="width: 10px; height: 10px; background-color: #D9E1F2; margin-left: 5px;"></div> <span>4</span> </div>
Operabilidade: Você acha que VULCAN tem elementos ambíguos ou difíceis de entender?	<div style="display: flex; justify-content: space-between;"> <span>■ Sim</span> <span>■ Não</span> </div> <div style="display: flex; align-items: center;"> <div style="width: 100px; height: 10px; background-color: #0070C0; margin-right: 5px;"></div> <span>5</span> <div style="width: 10px; height: 10px; background-color: #D9E1F2; margin-left: 5px;"></div> <span>9</span> </div>

Com relação à operabilidade, ficou evidente que, embora os participantes tenham sido capazes de executar o processo, o simples recebimento do *notebook* é insuficiente para orientá-los sobre como executar o processo (Tabela 3). Aplicando a Escala de Usabilidade do Sistema (SUS) [8] para medir a satisfação, foram obtidas altas pontuações com uma média de cerca de 80 pontos indicando um excelente nível de satisfação [3] (Tabela 4). O item do questionário SUS com a média mais baixa refere-se a se "é possível aprender a usar este sistema rapidamente". A pontuação mais alta foi relacionada à consistência do

sistema, sem nenhum participante indicando inconsistências.

Tabela 4: Pontuação no questionário SUS.

Professores			Estudantes			Média Total
Mínimo	Médio	Máximo	Mínimo	Médio	Máximo	
65	80.2	97.5	65	80.6	100	80.3

### 5.3 Discussão

Os resultados da avaliação fornecem uma primeira indicação de que a camada visual pode ser um suporte útil,

funcional, eficiente e utilizável para o ensino de AM no desenvolvimento de modelos de classificação de imagem para iniciantes no ensino médio e mesmo em atividades iniciais no ensino superior. Os principais pontos fortes baseados no *feedback* dos professores incluem a orientação passo a passo por meio da camada visual que orienta os iniciantes na execução do processo de desenvolvimento do modelo de AM. Os participantes também citaram a praticidade do fluxo de processamento, sendo capazes de executar etapas que normalmente são mais complexas ou opcionais, tais como o aumento de dados. Por sua vez, os estudantes enfatizaram como pontos fortes a forma compacta, concisa e focalizada com que a camada visual orienta o processo de AM. Também observou-se que a facilidade para reexecutar a rede neural os motivou a experimentar diferentes configurações. Entretanto, apontaram que a explicação da terminologia e do processo usado para desenvolver um modelo de AM com a camada visual precisa ser melhorada para se tornar mais compreensível, especialmente por iniciantes. A respeito desta sugestão, está sendo melhorado o suporte para o uso da camada visual de diferentes maneiras: incluindo informações de ajuda para cada etapa como parte da interface visual ligada à documentação externa on-line para fornecer explicações mais detalhadas sobre a terminologia e as etapas do processo, e criando um curso on-line para o ensino de conceitos de AM e o desenvolvimento de modelos de AM adotando a camada visual no modelo de exemplo para a classificação da imagem de espécies de árvores. Outras sugestões de melhoria relacionadas a questões de usabilidade incluíram, por exemplo, a adição de mais *feedback* visual sobre os resultados da aplicação das transformações, bem como a visualização do progresso das etapas de processamento. E, para evitar erros, os participantes também sugeriram a ativação condicional das etapas somente uma vez que as anteriores foram concluídas. Essas sugestões já foram consideradas na melhoria da versão final da camada visual. Um dos professores observou que embora o código Python subjacente para o desenvolvimento do modelo de AM esteja disponível, a camada visual não estimula o estudante a trabalhar com o código. De tal forma, pode distanciar o aluno dos conceitos básicos de programação dos modelos de AM. No entanto, como a adoção da camada visual pretende ser apenas um primeiro contato dos iniciantes com os conceitos de AM, ela visa reduzir o esforço cognitivo focalizado nos conceitos de AM. A fim de permitir que os estudantes, no final, possam desenvolver modelos de AM em um ambiente convencional usando uma linguagem de programação textual comum, como o Python, também é importante continuar o processo de aprendizado guiando os estudantes na transição de tal ambiente visual

para a programação textual. **Ameaças à validade.** Os resultados obtidos nesta avaliação precisam ser interpretados com cautela, levando em conta as ameaças potenciais à sua validade. Uma ameaça pode ser o projeto de pesquisa adotado ao realizar um teste de usuário. Devido à falta de medições em um contexto educacional real e um grupo de controle, os resultados são limitados para fornecer apenas uma primeira indicação da qualidade da camada visual proposta. Outra ameaça está relacionada com a diversidade e tamanho da amostra dos participantes. Considerando o foco principal no uso do VULCAN no ensino médio, não foi possível incluir os estudantes do ensino médio na avaliação devido à falta generalizada de conhecimento de AM nesta etapa educacional atualmente. Portanto, foram convidados estudantes do ensino superior que, em sua maioria, tinham pelo menos alguma experiência prévia com o AM. No entanto, como eles estão em um estágio educacional diferentes, podem não ser capazes de apontar problemas potenciais que os estudantes do ensino médio podem encontrar ao usar a camada visual. Nesse caso, avaliações adicionais precisam ser realizadas também com estudantes do ensino médio, por exemplo, como parte de cursos de AM. Por outro lado, em relação aos professores, foi possível incluir os professores do ensino médio e do ensino superior. Em termos de amostras, há também evidências de que 14 participantes são suficientes para obter resultados aceitáveis, especialmente quando considerada a natureza exploratória da pesquisa [25]. Outra possível ameaça é a definição da medida, pois a qualidade das ferramentas de software é difícil de ser medida. Para minimizar esta ameaça, os questionários foram desenvolvidos decompondo sistematicamente o objetivo da avaliação em itens de questionário adotando a abordagem GQM [5]. Devido a limitações práticas, os resultados da avaliação foram obtidos de um projeto de classificação de imagens pré-criado dentro de um contexto artificialmente controlado para fins de avaliação. Este tipo de avaliação que ocorre fora de um contexto educacional pode não ser suficiente e, portanto, são necessários mais estudos de avaliação em contextos educacionais para confirmar os resultados.

## 6 Conclusão

Neste artigo, apresentamos o VULCAN, uma camada visual desenvolvida a fim de facilitar o aprendizado dos conceitos de AM visando principalmente os alunos do ensino médio. Pretende-se que esta ferramenta seja adotada no início dos cursos de AM para estudantes iniciantes, a fim de reduzir a carga cognitiva. A camada visual fornece suporte para o processo completo de desenvolvimento de AM, incluindo análise de requisitos, preparação de dados, treinamento de

modelos, avaliação e exportação de uma forma simplificada para manter a carga cognitiva adequada para os iniciantes em AM. Desta forma, ela fornece uma interface visual, generalizada e flexível para fazer modelos de classificação de imagens que podem ser facilmente desenvolvidos sem experiência ou conhecimento prévio em AM ou codificação. A fim de facilitar sua aplicação em sala de aula e ilustrar seu uso, foi preparado também um exemplo de modelo de AM para a classificação de espécies de árvores, incluindo um conjunto de dados preparado. Os resultados de uma avaliação preliminar fornecem uma primeira indicação de que a camada visual pode ser um suporte útil, funcional, eficiente e utilizável para o ensino do desenvolvimento de modelos de classificação de imagens para iniciantes no ensino médio e superior. Um dos principais pontos fortes da camada visual é a indicação embutida dos passos e parâmetros necessários, bem como os valores padrões típicos que facilitam a experimentação do desenvolvimento de modelos de AM por iniciantes. Com base nas oportunidades de melhoria identificadas, atualmente está sendo desenvolvido um curso para fornecer mais explicações sobre como utilizar a camada visual, bem como a terminologia e os conceitos utilizados em AM.

## AGRADECIMENTOS

Agradecemos a todos os participantes da avaliação por seu valioso *feedback*. Este trabalho foi apoiado pelo CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), uma entidade do governo brasileiro voltada para o desenvolvimento científico e tecnológico.

## REFERÊNCIAS

[1] Amazon. Amazon Machine Learning, AWS Documentation. <https://docs.aws.amazon.com/machine-learning/latest/dg/building-machine-learning.html>.

[2] S. Amershi et al., Software Engineering for Machine Learning: A Case Study, Proc. of the 41st Int. Conference on Software Engineering, Montreal, Canada (2019).

[3] A. Bangor, P. Kortum e J. Miller, Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale, Journal of Usability Studies 4(3), 114–123 (2009).

[4] L. Barba et al., Teaching and Learning with Jupyter, GitHub, <https://jupyter4edu.github.io/jupyter-edu-book/>

[5] V. Basili, G. Caldiera e H. Rombach, Goal, Question Metric Paradigm, J.J. Marciniak (Ed.), Encyclopedia of Software Engineering, Wiley-Interscience, New York (1994).

[6] D. Bau, J. Gray, C. Kelleher, J. Sheldon, e F. Turbak, Learnable programming, Communications of the ACM, 60(6), 72–80 (2017).

[7] M. Beg et al., Using Jupyter for Reproducible Scientific Workflows. Computing in Science & Engineering, 23(2), 36–46 (2021).

[8] J. Brooke, SUS—A Quick and Dirty Usability Scale, P. W. Jordan, B. Thomas, B. A. Weerdmeester, A. L. McClelland, Eds., Usability Evaluation in Industry, Taylor and Francis, London (1996).

[9] M. Carney et al., Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification, Proc. of the Conference on Human Factors in Computing Systems, Honolulu, HI, USA (2020).

[10] CSTA K-12, Computer Science Standards, <http://www.csteachers.org/standards>

[11] D. Davis, Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology, MIS Quarterly, 13(3), 319–340 (1989).

[12] G. Dove, K. Halskov, J. Forlizzi e J. Zimmerman, UX Design Innovation, Proc. of the Conference on Human Factors in Computing Systems, 278–228, Denver, CO, USA (2017).

[13] A. C. M. Franz, Desenvolvimento de uma Ferramenta Visual de Classificação de Imagens para o Ensino de Machine Learning no ensino médio, Final Paper, Information System B.S., Universidade Federal de Santa Catarina <https://bit.ly/3wKGe0S>

[14] Google Colab, <https://colab.research.google.com/>

[15] Google Teachable Machine, <https://experiments.withgoogle.com/teachable-machine>

[16] C. Gresse von Wangenheim, et al. Visual tools for teaching machine learning in K-12: A ten-year systematic mapping". Educ Inf Technol 26, 5733–5778, 2021.

[17] C. Gresse von Wangenheim e A. von Wangenheim, Overview on a human-centric interactive ML process for teaching ML in K-12. Working Paper WP\_GQS\_01\_2021\_v10, GQS/INCoD/UFSC, Florianópolis, Brasil.

[18] J. Howard e S. Gugger, Deep learning for coders with fastai and PyTorch: AI applications without a PhD. Sebastopol: O'Reilly Media, Inc., (2020).

[19] Ipywidgets, Jupyter Widgets 8.0.0rc0 documentation, <https://ipywidgets.readthedocs.io/en/latest/>

[20] ISO/IEC 9241-210:2019, Ergonomics of human-system interaction — Part 210: Human-centered design for interactive systems, <https://www.iso.org/standard/77520.html>

[21] ISO/IEC 25010:2011, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuARE) — System and software quality models, <https://www.iso.org/standard/35733.html>

[22] Jupyter Notebook, <https://jupyter.org/>

[23] T. Kluyver et al., Jupyter Notebooks - a publishing format for reproducible computational workflows. IOS Press, 87–90 (2016).

[24] C. Larman e V. Basili, Iterative and incremental developments, Computer, 36(6), 47–56 (2003).

[25] C. Lawshe, A quantitative approach to content validity, Personnel Psychology, 28(4), 563–575 (1975).

[26] D. Long e B. Magerko, What is AI Literacy? Competencies and Design Considerations, Proc. of the Conference on Human Factors in Computing Systems, Honolulu, HI, USA 1–16 (2020).

[27] N. Lytle et al. Use, modify, create: Comparing computational thinking lesson progressions for stem classes. Proc. of the ACM Conference on Innovation and Technology in Computer Science Education, Aberdeen, Scotland, UK, 395–401 (2019).

[28] L. Marques, C. Gresse von Wangenheim e J. Hauck, Teaching Machine Learning in School: A Systematic Mapping of the State of the Art, Informatics in Education, 19(2), 283–321 (2020).

[29] M. McCracken et al., A multi-national, multi-institutional study of assessment of programming skills of first-year CS students, ACM SIGCSE Bulletin, 33(4), 125–180 (2001).

[30] K. Petersen et al. Systematic mapping studies in software engineering. Proc. of the 12th Int. Conference on Evaluation and Assessment in Software Engineering, Bari, Italy, 68–77 (2008).

[31] J. Piazzentin, O. Freire e J. Silva, Interactive Data Visualization in Jupyter Notebooks, Computing in Science & Engineering, 23(2), 99–106 (2021).

[32] PyPI, Python Package Index repository, <https://pypi.org/>

[33] M. Resnick, R. Berg e M. Eisenberg, Beyond black boxes: Bringing transparency and aesthetics back to scientific investigation. The Journal of the Learning Sciences, 9(1), 7–30 (2000).

[34] J. Rodríguez-García et al., Evaluation of an Online Intervention to Teach Artificial Intelligence with LearningML to 10-16-Year-Old Students, Proc. of the 52nd ACM Technical Symposium on Computer Science Education, New York, NY, USA 177–183 (2020).

[35] L. Sun et al., Developing a toolkit for prototyping machine learning-empowered products: The design and evaluation of ML-Rapid. International Journal of Design, 14(2), 35–50 (2020).

[36] S. Tamilselvan et al., A visual programming paradigm for abstract deep learning model development, Proc. of the 10th Indian Conference on Human-Computer Interaction, India, 1–11 (2019).

[37] D. Tang, Y. Utsumi e N. Lao, PIC: A Personal Image Classification Webtool for High School Students. Proc. of the IJCAI EduAI Workshop, Macao, China (2019).

[38] D. Touretzky et al., Envisioning AI for K-12: What Should Every Child Know about AI?, Proc. of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA 9795–9799 (2019).

[39] C. Wohlin et al., Experimentation in Software Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg (2012).