UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

HENRIQUE SOARES GOETZ

CVLabel: Uma plataforma online para rotulação de imagens

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Prof. Dr. Marcelo Pimenta

Porto Alegre
2022

RESUMO

Esse trabalho visa descrever e apresentar o desenvolvimento da plataforma CVLabel para promover a tarefa de rotulação de imagens, tipicamente realizada para identificar informações sobre conjuntos das imagens visando futuro treinamento de algoritmos de Machine Learning. Primeiramente, alguns conceitos fundamentais serão apresentados, como a própria tarefa de rotulação de imagens, algumas de suas características principais e as ferramentas disponíveis no mercado que serviram como inspiração, assim alguns de seus pontos positivos e negativos serão discutidos. A seguir serão apresentadas as tecnologias utilizadas no trabalho e a argumentação para serem escolhidas, a metodologia de desenvolvimento adotada, uma descrição geral da ferramenta CVLabel, do processo de implementação e um resumo das funcionalidades definidas e implementadas na aplicação. Por fim, serão apresentados os resultados obtidos até agora, as principais limitações e sugestões de trabalhos futuros.

Palavras-chave: Ferramenta web para rotulação de imagens. Anotação de imagens. CVLabel.

CVLabel: An online platform for image labeling

ABSTRACT

This work aims to describe and present the development of the CVLabel platform to promote the image labeling task, typically performed to identify information about sets of images for future training of Machine Learning algorithms. First, some fundamental concepts will be presented, such as the image labeling task itself, some of its main features and the tools available in the market that served as inspiration, so some of its positive and negative points will be discussed. Next, the technologies used in the work and the arguments to be chosen will be presented, the development methodology adopted, a general description of the CVLabel tool, the implementation process and a summary of the functionalities defined and implemented in the application. Finally, the results obtained so far, the main limitations and suggestions for future work will be presented.

Keywords: Web system tool for image labeling. Labeling tool. Image labeling.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

CVAT	Computer Vision Annotation Tool
MIT	Massachusetts Institute of Technology
UX	User Experience
HTTP	Hypertext Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
AJAX	Asynchronous JavaScript e XML
API	Application Programming Interface
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
SASS	Syntactically Awesome Style Sheets
JS	JavaScript
SQL	Structured Query Language
AWS	Amazon Web Services
S3	Simple Storage Service
RDS	Relational Database Service
SGBD	Sistema Gerenciador de Banco de Dados
DNS	Domain Name System
URL	Uniform Resource Locator
COO	Chief Operating Officer

SUMÁRIO

1 INTRODUÇÃO

Com o avanço da utilização de técnicas de Aprendizado de Máquina e Visão Computacional para solucionar problemas em diversos contextos, é cada vez mais comum se deparar com uma ferramenta que realiza algum tipo de processamento sobre imagens. Como exemplo de aplicações que utilizam essas técnicas, pode-se pensar em um leitor de QRCode, um aplicativo para reconhecimento de vinhos e até os sistemas utilizados em carros autônomos.

Se tratando de técnicas de Inteligência Artificial, é natural que uma parte do desenvolvimento desses sistemas seja realizar o treinamento do algoritmo. Para isso, é necessário possuir um conjunto de dados para dar segmento no processo. É visando facilitar a geração desses dados que esse projeto teve início.

O início do sistema se deu como um trabalho para uma cadeira da faculdade, onde a proposta era implementar um sistema que tivesse clientes reais para promover uma interação entre os alunos e os clientes. Formou-se uma dupla e como um dos integrantes trabalhava em uma empresa de soluções com inteligência artificial e o sistema em uso na empresa não atendia complemente a necessidade e as opções no mercado possuíam um custo elevado, foi proposto o desenvolvimento da plataforma CVLabel e disponibilização para testes e sugestões em parceria com a empresa. Por questões de confidencialidade, o nome da empresa não será citado nesse trabalho.

Embora as decisões tenham sido tomadas em equipe e o desenvolvimento de código dividido conforme a especialidade dos integrantes em tarefas de front-end e back-end. Ao decorrer desse texto, será introduzido o contexto da ferramenta e apresentadas as tecnologias, soluções, ideias e funcionalidades do sistema CVLabel como trabalho exclusivo do autor.

1.1 Objetivo

Esse trabalho pretende descrever e apresentar o desenvolvimento da plataforma CVLabel para promover a tarefa de rotulação de imagens, tipicamente realizada para identificar informações sobre conjuntos das imagens visando futuro treinamento de algoritmos de Machine Learning.

1.2 Estrutura do trabalho

O texto está dividido em 7 capítulos. O primeiro capítulo é a introdução que procura contextualizar o que será apresentado nos demais capítulos. O segundo capítulo apresenta fundamentos, conceitos e os sistemas de rotulação de imagens que tiveram influência no desenvolvimento do CVLabel. O terceiro capítulo cita e explica as ferramentas e tecnologias utilizadas no trabalho. O quarto capítulo aborda os requisitos, o planejamento e a metodologia empregada. O quinto capítulo explica como o sistema foi desenvolvido, aprofundando o uso das ferramentas e tecnologias citadas no capítulo 3. O sexto capítulo busca apresentar as funcionalidades e a usabilidade da plataforma atual. Por fim, o sétimo capítulo é a conclusão desse trabalho onde será analisado o resultado obtido, as limitações e o que se pensa como continuidade da plataforma.

2 FUNDAMENTOS E SISTEMAS RELACIONADOS

Nesse capítulo, serão apresentados alguns fundamentos sobre sistemas de análise de imagens, a tarefa de rotulação de imagens e os pontos negativos e positivos de sistemas desenvolvidos para esse propósito que serviram de inspiração para o CVLabel.

2.1 Tarefa de Rotulação

Normalmente, uma ferramenta que realiza alguma computação sobre imagens tem como objetivo classificar e/ou detectar elementos nas imagens e esses objetivos dão origem às tarefas classificação e detecção de imagens. A tarefa de classificação de imagens consiste em determinar a partir dos píxeis de uma imagem a qual classe certo elemento pertence. Por exemplo, a partir da imagem de um animal de estimação, classificá-lo com um cachorro. Já a tarefa de detecção de imagens, consiste em encontrar a posição dos elementos na imagem em questão. Por exemplo, detectar a posição de uma pessoa atravessando uma rua.

Para desenvolver uma aplicação que realize algum tipo de análise sobre imagens, é necessário construir uma base de dados para que os algoritmos possam ser treinados e validados. Portanto, se faz necessário uma análise humana sobre as imagens para gerar esse conjunto de dados para que o desenvolvimento possa continuar, essa é chamada a tarefa de rotulação. A rotulação de imagens trata-se da atividade de analisar e gerar marcações nas imagens, mapeadas de acordo com as posições da imagem e, dessa forma, podem ser utilizadas para o desenvolvimento da aplicação final.

Como pode-se imaginar, para um conjunto extenso de imagens esse trabalho se torna algo que requer tempo e equipes. Para facilitar essa operação, surgiram sistemas de software para rotulação de imagens e o CVLabel é um deles. A seguir, serão apresentadas as principais ferramentas que serviram como modelos para implementação da ferramenta CVLabel.

2.2 Labelme

O Labelme é um projeto open source desenvolvido em Python inspirado na ferramenta de mesmo nome feita pelo MIT. Esse projeto foi desenvolvido apenas para desktop

offline. A ferramenta original é uma aplicação online desenvolvida para fins de pesquisa, que permite anotação de polígonos, retângulos, círculos, linhas e pontos.

Essa ferramenta foi bastante considerada porque era a ferramenta em uso na empresa parceira. Entretanto, a ferramenta usada já utilizava uma versão otimizada pela equipe de desenvolvedores da empresa. Entre as principais melhorias em relação à ferramenta original pode-se citar:

- Noção de dataset e criação de classes globais
- Cor nas classes
- Opção para desativar a interpolação
- Adição de novos formatos de importação e exportação
- Auto save
- Melhorias de UX

Figura 2.1: Dashboard de Anotação da ferramenta Labelme

Captura de tela retirada do Github do projeto

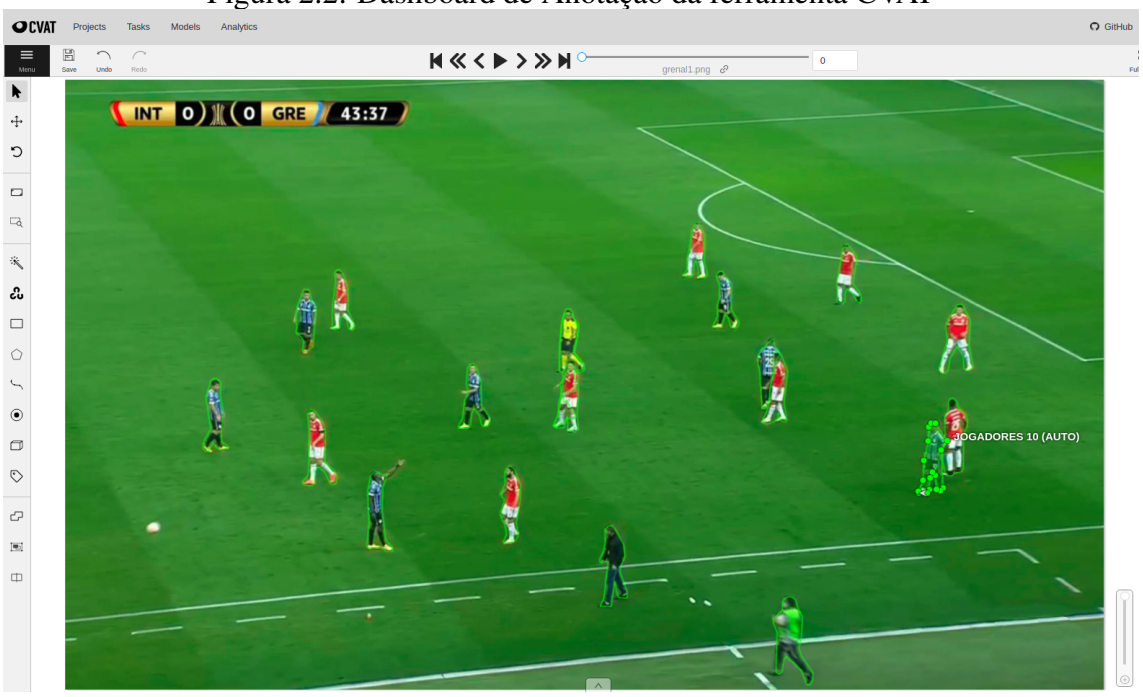
O principal ponto negativo da ferramenta Labelme é o fato de ser uma ferramenta offline. Essa característica afeta muito dois requisitos: segurança do dataset e sincronização de tarefas. Como as imagens precisam estar armazenadas nos dispositivos, é necessário realizar cópias, diminuindo o controle sobre os dados. Além disso, a sincronização das anotações pela equipe é difícil de ser realizada em sistemas offline.

2.3 CVAT

O CVAT é um projeto open source desenvolvido utilizando o framework Django com plataforma grátis e online. Possui como limitação 500Mb para upload e até 10 tarefas por usuário. Além disso, não permite adicionar ou remover imagens de uma tarefa já criada. O conjunto, uma vez criado, não pode ser modificado e o gerenciamento dos projetos não é muito atrativo.

Entretanto, o CVAT disponibiliza uma funcionalidade interessante, a anotação automatizada para grupos predefinidos. A plataforma permite selecionar um tipo de elemento, por exemplo, “Pessoa” e solicitar uma anotação automática. A seguir um exemplo do resultado, utilizando o tipo “Pessoa” em um jogo de futebol.

Figura 2.2: Dashboard de Anotação da ferramenta CVAT

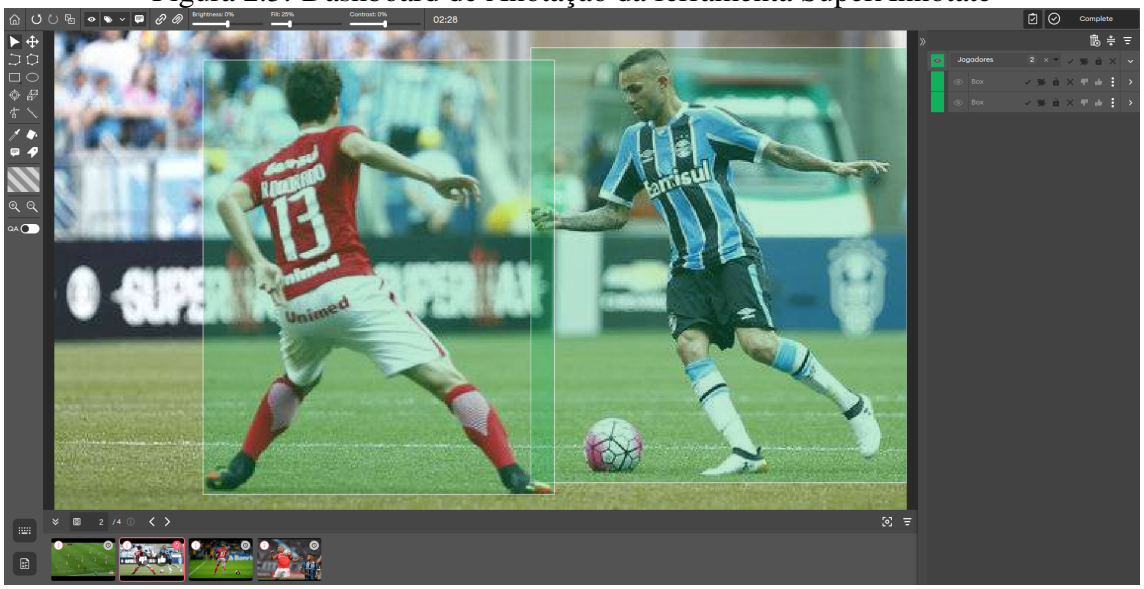


Exemplo do resultado obtido utilizando anotação automática

2.4 SuperAnnotate

SuperAnnotate é uma plataforma paga e online. Os custos não estão mais disponíveis publicamente (é preciso entrar em contato). Entretanto, os valores são normalmente cobrados por usuário. Assim como o CVAT, também disponibiliza tecnologias para anotação automatizada.

Figura 2.3: Dashboard de Anotação da ferramenta SuperAnnotate



Captura de tela realizada utilizando a plataforma Superannotate

Pelos testes realizados utilizando o plano de teste de 14 dias, pode-se perceber que está é uma das ferramentas mais completas do mercado. Por isso, se tornou um modelo de qualidade a ser seguido.

3 TECNOLOGIAS ENVOLVIDAS E FERRAMENTAS AWS

Nesse capítulo serão resumidos alguns fundamentos de uma aplicação web e explicadas as tecnologias e ferramentas utilizadas no desenvolvimento da plataforma CVLabel.

3.1 HTTP e HTTPs

HTTP é um protocolo de comunicação desenvolvido pelo britânico Tim Berners-Lee, que ficou conhecido como o criador da World Wide Web, a internet. Quando o navegador de um dispositivo realiza requisições para a internet para carregar arquivos de multimídia, é esse protocolo que está sendo utilizado.

O HTTPs é uma versão melhorada do protocolo HTTP, onde se utiliza criptografia nas mensagens a fim de proteger o conteúdo das mesmas enquanto trafegam pela rede.

3.2 Requisições AJAX

Normalmente, quando se navega por um determinado site, conforme o usuário troca de conteúdo, o navegador segue realizando requisições HTTP e carregando as novas páginas. Entretanto, em alguns cenários, é interessante realizar uma requisição sem gerar um novo carregamento de página. Para esses casos, os desenvolvedores utilizam requisições AJAX, realizadas via Javascript utilizando o objeto XMLHttpRequest e são assíncronas. Ou seja, são feitas em background, permitindo que a aplicação continue disponível enquanto a resposta ainda não foi recebida.

3.3 API

Uma API é uma interface de comunicação de programas que disponibiliza chamadas do sistema. No caso de uma API web, são disponibilizadas rotas que podem ser requisitadas para que uma determinada operação seja realizada. Além das rotas, o tipo de requisição é comumente utilizado para definir qual método deve ser executado quando uma requisição é recebida.

3.4 HTML

O HTML é o bloco web cuja função é montar a estrutura e a semântica dos elementos que serão colocados exibidos em tela.

3.5 CSS

O CSS é o bloco web que personaliza o conteúdo HTML adicionando estilos como posicionamento, cores e tamanhos.

3.6 SASS/SCSS

SASS é uma linguagem de extensão do CSS. Utilizando-a, o desenvolvimento do estilo da aplicação se torna mais fácil e o código mais limpo. Isso porque essa linguagem estende o CSS com mais recursos. Então, a partir do código SASS, um compilador gera o respectivo código CSS final. SCSS é a sintaxe SASS utilizada.

3.7 Javascript

Javascript é uma linguagem de programação que, no contexto desse trabalho, executa no navegador do cliente e permite que a aplicação reaja conforme eventos acontecem na página. É através dessa tecnologia que os desenvolvedores conseguem tornar as páginas web interativas.

3.8 Vue.js

É um framework Javascript utilizado para desenvolver aplicações front-end. Com ele é possível gerar HTML, CSS e JS a partir de um modelo próprio de programação. Dessa forma, permite simplificar o desenvolvimento de funcionalidades mais complexas e aumentar a compatibilidade entre os navegadores.

3.9 vue-chartjs

É uma biblioteca Javascript para geração de gráficos dinâmicos, desenvolvida sobre a biblioteca Chart.js e o framework Vue.js.

3.10 Python

É uma linguagem de programação interpretada e de código aberto utilizada para diversos fins como: data science, machine learning, desenvolvimento web, automação, etc. É reconhecida pelos desenvolvedores como de fácil aprendizado, eficiente e flexível.

3.11 SQLAlchemy

Desenvolvido para a linguagem Python, o SQLAlchemy é um framework de mapeamento objeto-relacional SQL. O objetivo dessa ferramenta é permitir uma codificação mais robusta, enxuta e simples quando se realiza a conversão dos dados com o banco de dados.

3.12 Amazon S3

É um serviço de armazenamento em nuvem escalável que permite armazenar arquivos com acesso protegido. O serviço promete altos níveis de escalabilidade, disponibilidade, segurança e performance. Foi projetado para fornecer 99,999999999% de durabilidade e 99,99% de disponibilidade dos arquivos em determinado ano.

3.13 Amazon RDS

É um conjunto de serviços da Amazon que facilita o gerenciamento de bancos de dados na nuvem. O RDS é compatível com diversos sistemas de gerenciamento de bancos de dados como: Amazon Aurora/MySQL, Amazon Aurora/PostgreSQL, MySQL, MariaDB, PostgreSQL, Oracle e SQL Server.

3.14 Amazon Aurora PostgreSQL

O Amazon Aurora é um sistema de gerenciamento de banco de dados criado para ser um serviço em nuvem. O sistema promete performance e disponibilidade por um décimo do custo tradicional. O Amazon Aurora PostgreSQL é a versão do AWS Aurora compatível com o SGBD PostgreSQL.

3.15 DNS

DNS é o protocolo responsável por traduzir os endereços web em endereços IP.

3.16 Amazon Route 53

É um serviço da Amazon para gerenciamento de DNS.

3.17 Amazon Gateway

É um serviço da Amazon para criação, gerenciamento e manutenção de API's.

3.18 Amazon Lambda

É um serviço escalável da Amazon orientado a eventos. Por meio desse mecanismo, é possível executar funções sem gerenciar servidores. Para o contexto desse trabalho, se utilizou para tratar as requisições da API.

3.19 Amazon Cognito

É um serviço da Amazon para gerenciamento de usuários. Ele permite o cadastro, autenticação e controle de acesso de usuários a aplicações. Pode ser escalado para milhões de usuário e oferece suporte a login com os principais provedores de identidade social, como: Apple, Facebook, Google.

3.20 Lazyload

É um método utilizado em alguns componentes HTML (imagens, iframes, vídeos) para otimização do carregamento de uma aplicação web. Originalmente, quando uma página continha uma imagem, o navegador já realizava a requisição dessa imagem para renderizá-la em tela.

O problema é que, dessa forma, o carregamento da página fica pendente esperando o carregamento das imagens e, em alguns casos, a imagem nem apareceria no início da página, portanto, poderia carregar depois que a página já estivesse disponível.

Para solucionar esse problema, surgiram bibliotecas para realizar o carregamento da imagem de forma assíncrona à construção da página, permitindo um tempo de carregamento melhor.

Atualmente, alguns navegadores já utilizam esse método automaticamente, enquanto outros fornecem suporte. Todavia, ainda é comum utilizar uma biblioteca para tratar o problema independente do navegador.

3.21 vue3-lazyload

É uma biblioteca Javascript desenvolvida para o framework Vue.js na versão 3 que inclui a possibilidade de adicionar um carregamento lazy aos componentes desejados.

4 CVLABEL: REQUISITOS E ORGANIZAÇÃO DO PROJETO

Nesse capítulo serão descritos os requisitos iniciais e a organização do projeto. Detalhes sobre a implementação serão dissertados no capítulo 5 e, no capítulo 6, serão demonstradas as funcionalidades disponíveis na aplicação.

4.1 Análise dos Requisitos

Os primeiros requisitos do projeto surgiram em conjunto com a necessidade de uma empresa que implementava soluções utilizando processamento de imagens e precisava melhorar sua tarefa de anotação de imagens. Naquele momento, a empresa utilizava uma versão offline e otimizada da ferramenta Labelme e o projeto surgiu como um trabalho para faculdade e teve como meta suprir os pontos negativos do Labelme e adicionar novas funcionalidades. Em um primeiro momento, baseado na experiência de um dos integrantes da equipe do CVLabel e as ferramentas disponíveis no mercado, foram debatidas as funcionalidades e características importantes para um sistema de rotulação de imagens internamente.

Então, com a equipe familiarizada com o contexto, se realizou uma conversa com membros da empresa para discutir as possibilidades, funcionalidades e prioridades. Nessa reunião, estavam presentes os integrantes da dupla CVLabel e três integrantes da empresa parceira: dois rotuladores e o COO. A partir desse contato, foram originados os requisitos funcionais e gerais primordiais da aplicação.

4.2 Requisitos Funcionais

Foram definidos um conjunto essencial de funcionalidades para serem implementadas no sistema. São elas:

- Anotação de Retângulos

  - Para encurtar o escopo do primeiro modelo, se definiu que a primeira versão permitiria apenas realizar a anotação de shapes do tipo retângulo.

- Sistema Online

  - Para permitir compartilhamento e maior segurança, se optou por desenvolver a ferramenta como uma ferramenta web.

- Simples transição do Labelme
  - Para facilitar a transição para o CVLabel, se decidiu que os atalhos e interações seriam os mesmos que o Labelme.

4.3 Facilidade para compartilhar os datasets

Como uma tarefa de anotação é normalmente realizada por uma equipe, o trabalho precisa ser dividido entre os anotadores. Para uma plataforma offline, o conjunto de imagens precisaria ser distribuído entre a equipe. Então, a equipe teria de dividir o conjunto de imagens entre os integrantes para realizar a tarefa. Se algum integrante cometer um engano, a probabilidade da tarefa não ser concluída corretamente ou o trabalho ser duplicado é significativa.

Além disso, mesmo que a divisão seja correta, é comum que a velocidade de anotação dos integrantes varie. Dessa forma, para continuar trabalhando, os mais rápidos, quando terminassem, precisariam contatar os outros integrantes para redividir as imagens restantes.

Como o CVLabel foi modelado como uma ferramenta online, é possível que um único usuário faça a submissão do conjunto de imagens para anotação e, a partir disso, toda a equipe pode trabalhar pela plataforma sem manter cópias locais.

4.4 Segurança do conjunto de dados

Tratando-se de uma ferramenta online, a segurança dos datasets já é maior pelo simples fato de que não é necessário que os usuários mantenham cópias dos dados em seus dispositivos. Entretanto, como todos dados ficam disponíveis pela rede, alguns cuidados foram tomados para manter os dados acessíveis apenas para usuários permitidos.

Para que múltiplos grupos possam utilizar a plataforma, se criou a entidade workspace. Para que um usuário tenha acesso a qualquer informação de um dataset, o mesmo precisa fazer parte do workspace que o dataset pertence. Logo, duas empresas diferentes podem utilizar a ferramenta, cada uma cadastrando seus datasets em seu respectivo workspace e compartilhando o acesso a suas equipes.

Além disso, a url das imagens são disponibilizadas somente utilizando urls pré-assinados. Ou seja, só se adquire o caminho das imagens através uma requisição autenti-

cada para a API.

4.5 Sincronização das tarefas

Para que uma equipe seja produtiva, ele precisa poder trabalhar de forma paralela e sem conflitos. Apoiado novamente no modelo online, o CVLabel permite que todos os integrantes possam visualizar e acompanhar o trabalho dos demais para auxiliar na tarefa. Como diferencial, ainda conta com uma funcionalidade de salto para a próxima imagem não anotada do conjunto de imagens.

4.6 Metodologia de Desenvolvimento

No início do projeto, se fez necessário definir como o desenvolvimento seria organizado e as metodologias tradicional e ágil foram consideras.

A metodologia tradicional é normalmente utilizada para desenvolvimento de projetos com escopo bem definido. O desenvolvimento costuma ser planejado em etapas, mas com uma visão completa do trabalho a ser desenvolvido. Cada etapa tem seu prazo definido e o objetivo é manter os prazos para que no final das etapas o produto esteja finalizado. Como os custos e prazos são calculados no início do trabalho, é importante para essa metodologia que o planejamento não sofra muitas alterações. Do contrário, a entrega tem grande chance de atrasar e o custo do projeto pode ultrapassar o orçamento planejado.

Como contraponto, existe a metodologia ágil. Esse modelo é comumente utilizado quando é difícil ter uma definição da complexidade e do escopo do projeto. Dessa forma, a metodologia prega comunicação constante entre a equipe e particionamento das tarefas para serem resolvidas em prazos curtos. As equipes que utilizam essa tecnologia costumam definir reuniões periódicas para debater as tarefas em desenvolvimento, definir novas tarefas e manter o controle sobre o andamento.

Como o projeto não possui um escopo definido e novas funcionalidades seguem surgindo durante o trabalho, se optou por utilizar o modelo de desenvolvimento ágil. Para isso, a equipe, desde o início, realiza reuniões para organizar o projeto. Inicialmente, as reuniões aconteciam uma vez por semana e atualmente acontecem uma vez a cada duas semanas.

4.7 Ambientes: produção e desenvolvimento

Existindo a preocupação de gerenciar um ambiente de desenvolvimento produtivo e seguro, foram preparados dois ambientes: produção e desenvolvimento. Assim, a equipe de desenvolvimento pode dar continuidade na aplicação e testar em um ambiente já conectado na estrutura da AWS sem interferir no ambiente de produção.

Para isso, foi necessário configurar cópias das estruturas de produção na AWS: banco de dados, buckets do AWS S3, API no AWS Gateway.

4.8 Versionamento e Gitflow

Versionamento de código consiste na prática de manter o desenvolvimento e as versões de um programa em uma plataforma de versionamento. Dessa forma, se mantêm registradas todas as atualizações desenvolvidas desde o início do projeto. Como exemplo de plataformas de versionamento pode-se citar o Github, o Bitbucket e o Gitlab. No início do projeto, se utilizava como plataforma de versionamento o Bitbucket, todavia se optou por migrar o projeto para a plataforma Github. O motivo dessa migração foi o tempo de execução de pipelines gratuitos das plataformas.

Pipeline é uma funcionalidade disponível nas plataformas de versionamento que permite a automação de tarefas a fim de testar, validar e até lançar as atualizações adicionadas na plataforma de versionamento. Essas tarefas a serem executadas são configuradas conforme o objetivo dos desenvolvedores. No caso desse projeto, o objetivo era realizar tarefas de teste sobre o código atualizado. Para isso, é necessário levantar uma máquina virtual, instalar as dependências do projeto e então realizar os testes.

Como esse processamento é realizado pela plataforma de versionamento, elas cobram de acordo com tempo de execução que os pipelines exigem. No caso do Bitbucket o limite é de até 50min por mês no plano gratuito, até 2500min no plano Standard e até 3500min no plano Premium. Já o Github, é mais flexível para o plano gratuito. O plano gratuito permite até 2000min, o plano Team 3000min e o plano Enterprise 50000min. Como, já no primeiro mês, foi atingido o tempo máximo do plano gratuito do Bitbucket, a equipe optou por migrar para o Github para continuar utilizando os pipelines sem custo adicional.

Para aumentar o controle e a organização sobre o código do projeto, se optou por utilizar como metodologia de versionamento o GitFlow. GitFlow é um padrão de fluxo de

versionamento. Nesse modelo, existem sempre pelo menos duas branches ativas: master e develop. O código na branch master deve estar completamente validado e testado, pronto para a próxima atualização em produção. Já a branch develop, deve conter o estado atual do desenvolvimento. Toda nova funcionalidade ou correção que for requisitada, deve ser implementada em outra branch criada para essa tarefa. Quando a atualização estiver finalizada, um merge com a branch develop deverá ser realizado, novos testes e verificações devem ser realizados e, se tudo estiver conforme o esperado, um merge da branch develop com a master poderá ser realizado.

Figura 4.1: Fluxo Gitflow

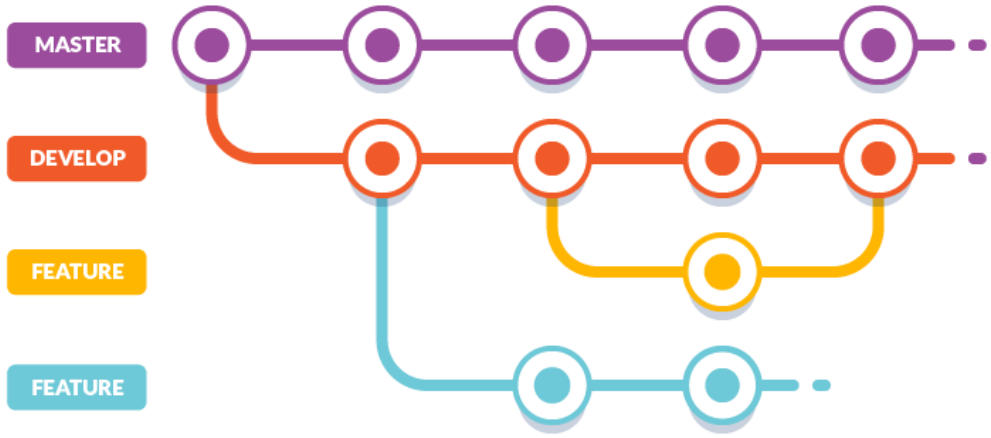
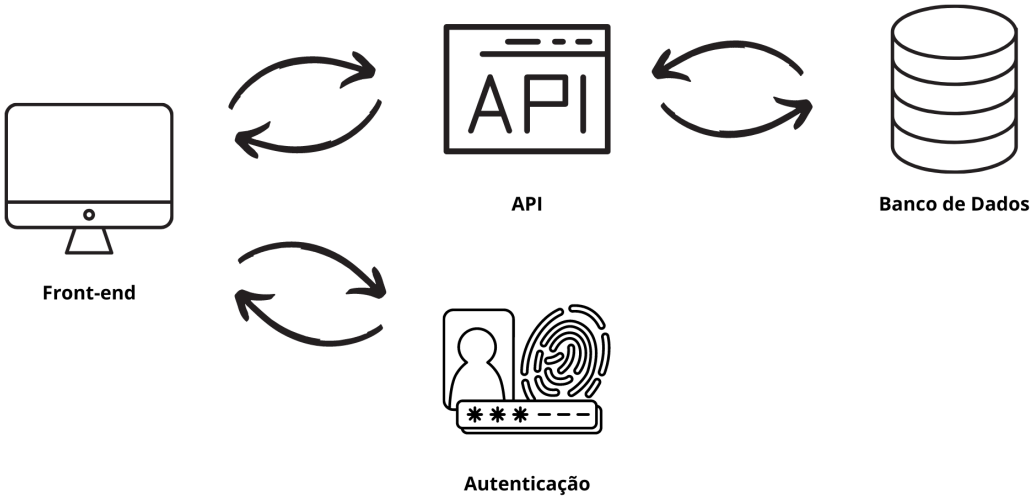


Ilustração explicativa do fluxo Gitflow

5 CVLABEL: IMPLEMENTAÇÃO

Esse capítulo tem com objetivo detalhar as características e as decisões referentes a implementação da plataforma CVlabel. Uma aplicação web é um sistema desenvolvido para ser acessado por meio de um navegador no modelo cliente-servidor. Para cada segmento desse processo, vamos analisar o fluxo de uma utilização da plataforma.

Figura 5.1: Arquitetura do Sistema



Front-end, Autenticação, API e Banco de Dados

5.1 Aplicação Front-end

Para acessar o sistema, o usuário deve acessar o site https://app.cvlabel.com. Esse endereço web foi configurado no AWS Route 53 para resolver em um bucket do AWS S3 que contém os arquivos estáticos essenciais para o front-end da aplicação. Então, por meio de uma requisição HTTP, ao acessar a aplicação, o navegador do cliente recebe os arquivos necessários para a geração do site na máquina local, que são compostos por arquivos HTML, CSS e JS. Esses arquivos foram gerados a partir do framework Vue.js e as bibliotecas citadas no capítulo 2. A partir desses arquivos o navegador pode renderizar a aplicação CVLabel.

Entretanto, antes de realizar qualquer requisição para a API, o usuário precisar estar identificado e autenticado. Para isso, se o usuário ainda não estiver nesse estado, a aplicação o redirecionará para o formulário de autenticação do AWS Cognito. Preen-

chendo o formulário com sucesso, o sistema do AWS Cognito retorna para o CVLabel por um redirecionamento com o token de autenticação na URL. Agora, o usuário está autenticado e o token será adicionado no header de todas requisições para a API. Quando o token expirar, um novo redirecionamento para o formulário do AWS Cognito é realizado e o processo se repete.

5.2 API

A API foi desenvolvida utilizando a linguagem de programação Python e a biblioteca SQLAlchemy, então foi incluída no AWS Gateway. O tratamento das requisições é realizado através do serviço AWS Lambda. Dessa forma, quando uma requisição é realizada para a API, esse serviço dispara a execução do tratamento da respectiva requisição.

5.3 Banco de Dados

Para se persistir todas as informações da plataforma, se faz necessário a utilização de um banco de dados. Para gerenciamento do banco de dados se utilizou o serviço Amazon RDS. No início do projeto, se optou pelo SGBD PostgreSQL, porém devido à limitação de conexões simultâneas no Amazon RDS para esse SGBD, se migrou para o SGBD Amazon Aurora. Essa migração foi simples de ser realizada porque o Amazon Aurora possui uma versão compatível com o PostgreSQL. Abaixo segue uma imagem ilustrativa das entidades existentes no banco de dados e um esclarecimento sobre cada uma delas.

Figura 5.2: Entidades do Banco de Dados

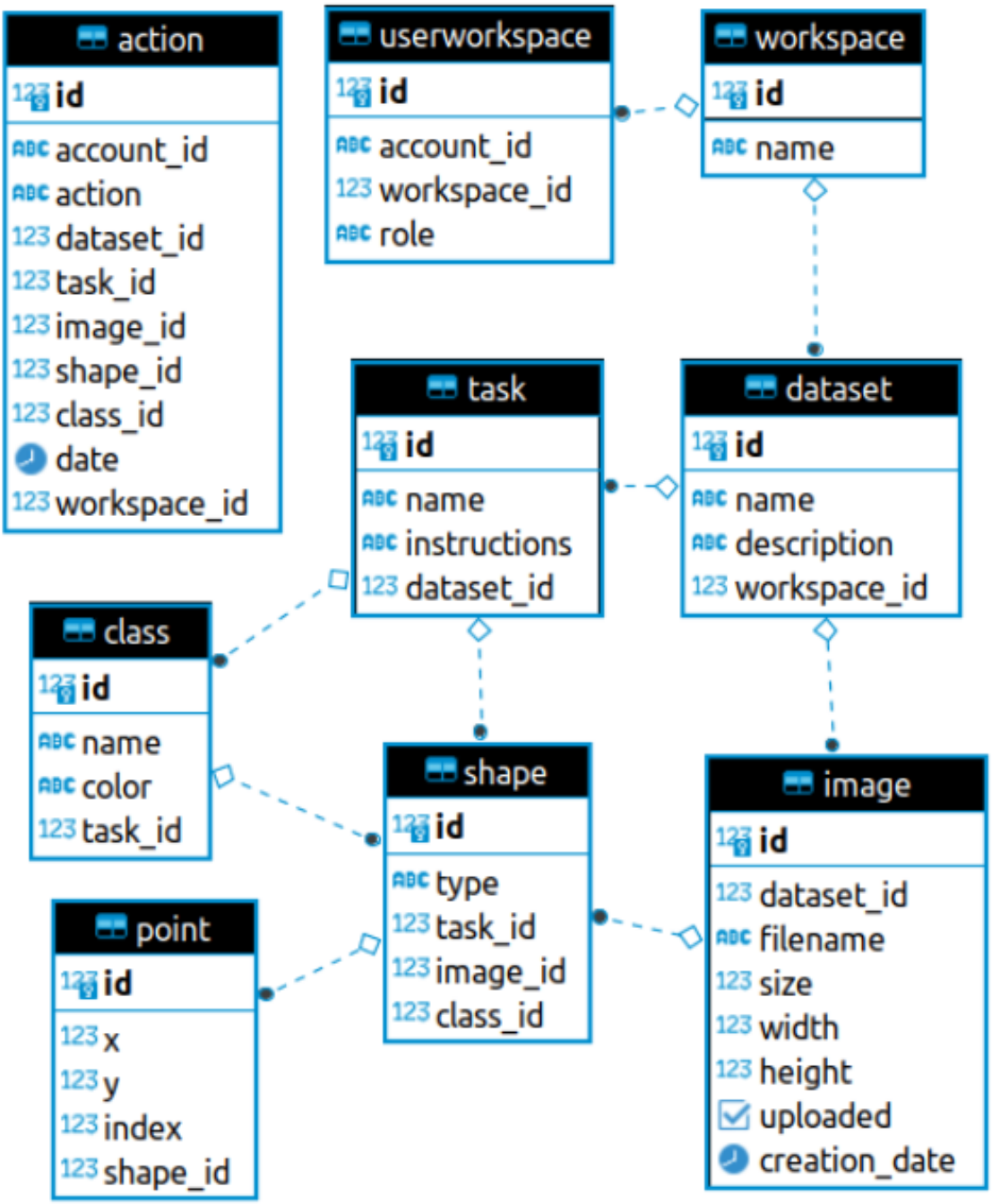


Ilustração para demonstração das entidades e suas relações

Obs.: O identificador de um usuário é sempre o e-mail (imutável) utilizado no registro do AWS Cognito.

5.3.1 Workspaces

Representação das organizações/empresas que um usuário participa. Todo dataset criado na plataforma está associado a um único workspace e todos os usuários presentes no workspace tem acesso aos datasets do mesmo.

5.3.2 Userworkspace

Representação da relação de presença de um determinado usuário em um determinado workspace.

5.3.3 Dataset

Entidade relacionada com um único workspace e composta por um conjunto de imagens e tarefas a serem realizadas sobre esse conjunto. Todas as imagens e tasks pertencem a um único dataset.

5.3.4 Image

Representação de uma imagem do dataset. Cada imagem está presente em apenas um dataset.

5.3.5 Task

Representação de uma tarefa de rotulação a ser realizada sobre o conjunto de imagem do dataset. Podem existir diversas tasks sobre o mesmo conjunto de imagens.

5.3.6 Shape

Representação de uma anotação. Todo shape está relacionada com uma única imagem e uma única task.

5.3.7 Action

Representação de uma ação realizada por um usuário. Pode ser do tipo criação, edição ou remoção. É utilizada para geração de estatísticas e histórico de ações.

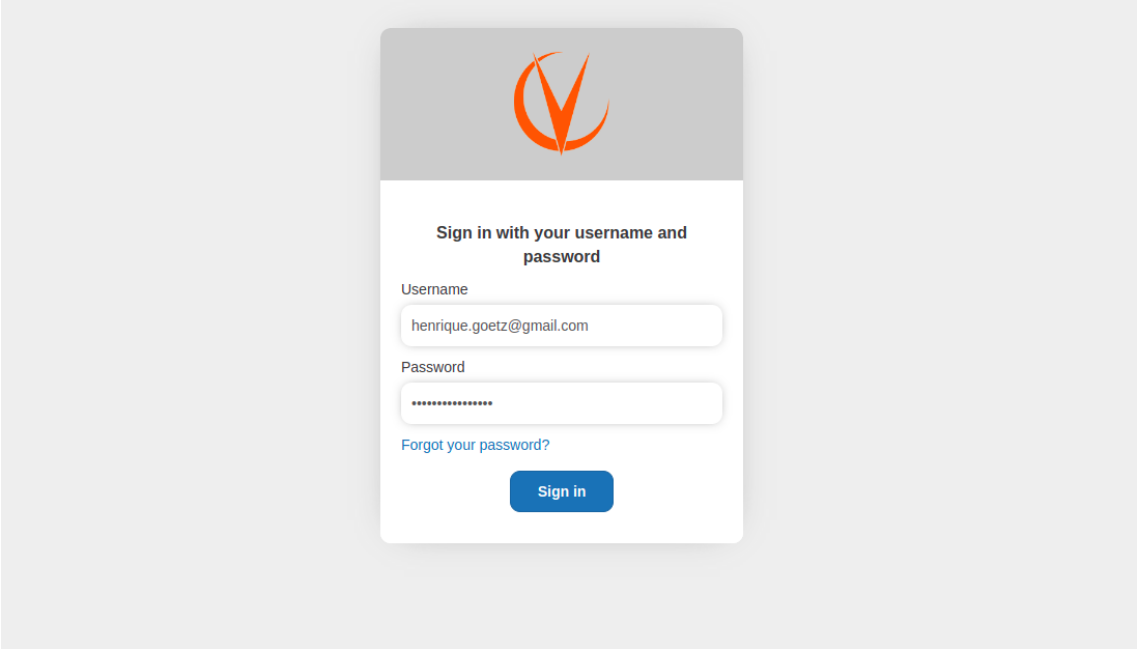
6 CVLABEL: FUNCIONALIDADES

Esse capítulo procura apresentar as funcionalidades, decisões de usabilidade e o visual da plataforma CVLabel.

6.1 Autenticação

Antes de realizar qualquer requisição para a API, o front-end precisa enviar um token de sessão válido. Caso o usuário não esteja autenticado ou o token tenha expirado, a aplicação redireciona o usuário para tela de autenticação do serviço AWS Cognito.

Figura 6.1: Formulário de Autenticação do Cognito



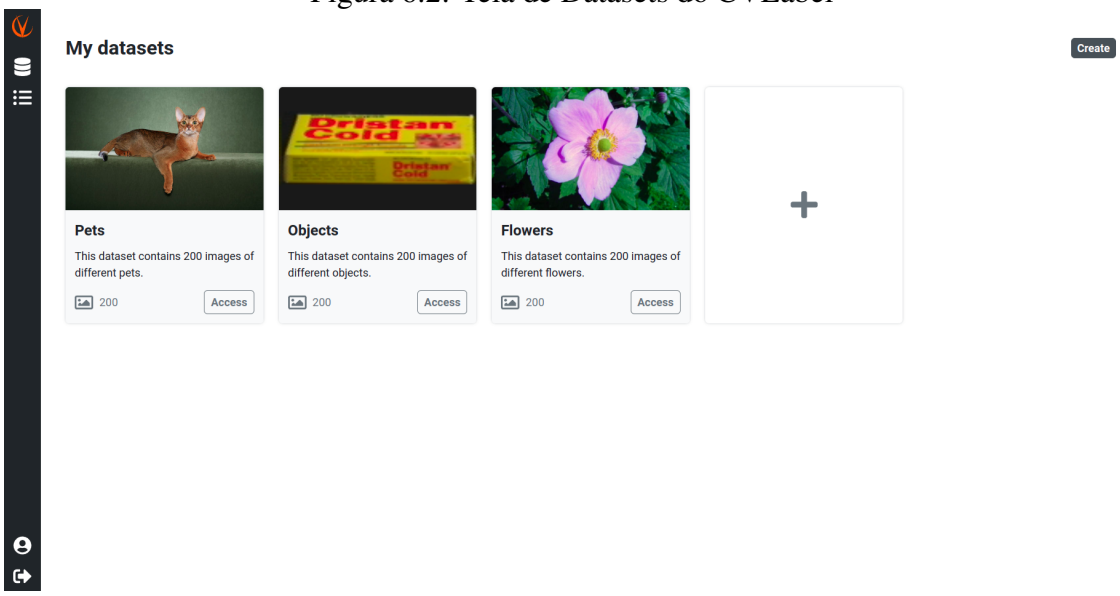
Captura de tela do formulário de autenticação do CVLabel

Após a autenticação, o Cognito redireciona o usuário novamente para a aplicação fornecendo o token de sessão. A partir de então, a aplicação pode realizar as requisições utilizando o token fornecido. Por questões de segurança, o token possui um tempo de expiração de 24 horas.

6.2 Telas de Datasets

Essa tela tem com objetivo criar e acessar os datasets do usuário. O usuário pode criar um dataset clicando no botão Create e preenchendo o nome e uma descrição opcional para o dataset. A partir desse momento, o dataset estará disponível e aparecerá como um card nessa tela. Para acessá-lo, basta clicar sobre o card.

Figura 6.2: Tela de Datasets do CVLabel



Captura da tela de Datasets do CVLabel

6.3 Tela do Dataset

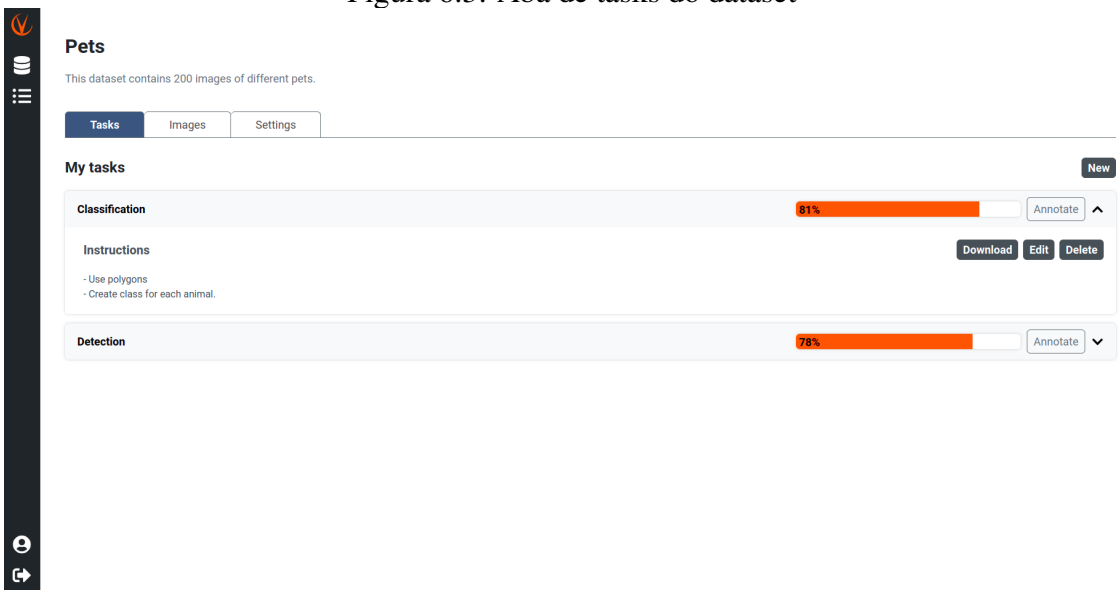
Para que o processo de anotação seja realizado, antes de tudo é necessário fornecer as imagens que serão anotadas. Além disso, para o CVLabel, é necessário criar as tasks de anotação que serão realizadas. A tela de Dataset busca permitir as funcionalidades de gerenciamento das imagens, tasks e do próprio dataset. Para organizar essas informações, foram desenvolvidas três abas: Tasks, Images e Settings.

6.3.1 Aba Tasks

A aba Tasks é a aba mais usada entre as abas de um dataset. Nela, é possível criar, gerenciar e acessar as tasks do dataset. Através da listagem das tasks, rapidamente,

os usuários podem acompanhar o progresso da todas as tasks do dataset. Além disso, podem partir para a anotação dessas tarefas clicando no botão Annotate. Clicando sobre a task, a mesma se expande permitindo que o usuário possa ler suas instruções e realizar as operações de: Download, Edit e Delete.

Figura 6.3: Aba de tasks do dataset



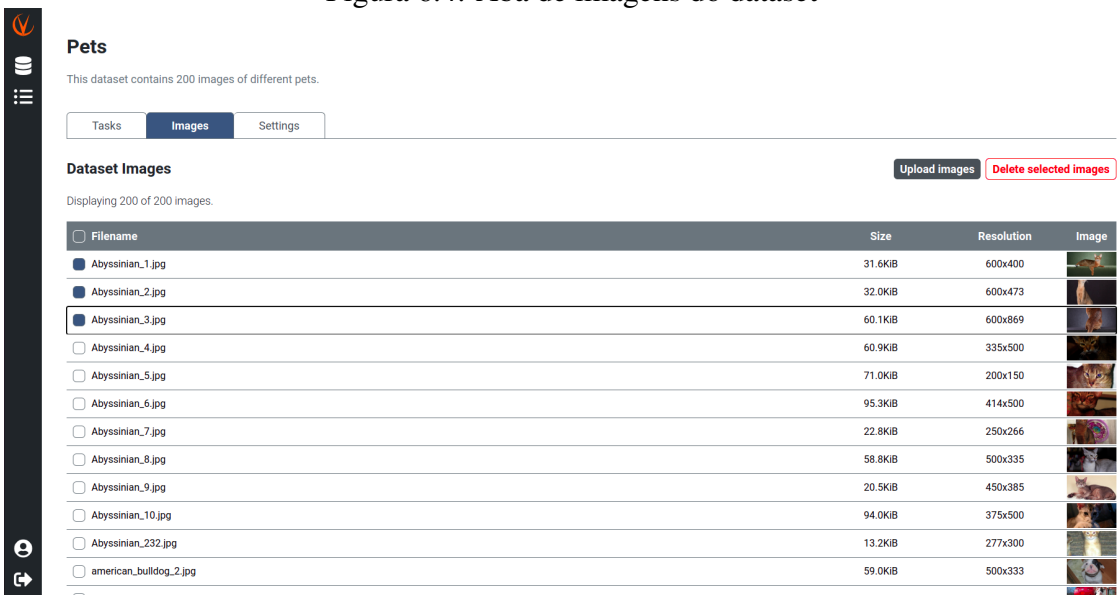
A opção Download é a opção que permite coletar o resultado do trabalho de rotulação. O resultado é devolvido como um arquivo comprimido (.zip) que contem um arquivo no formato json para cada imagem do dataset. Cada um desses arquivos representa as anotações realizadas sobre a imagem em questão. A opção Edit permite atualizar o nome e a descrição da task e a opção Delete permite excluir a task.

6.3.2 Aba Images

A aba Images permite o upload e a remoção das imagens. Para realizar o upload, basta que o usuário clique no botão Upload images e selecione as imagens do seu dispositivo. Quando o usuário realiza esse procedimento, a aplicação inicia o processo de upload e fornece um barra de progresso para que o usuário tenha noção do andamento.

Para remover imagens, o usuário deve selecionar as imagens que deseja excluir e clicar no botão Delete selected images que aparecerá quando ao menos uma imagem estiver selecionada.

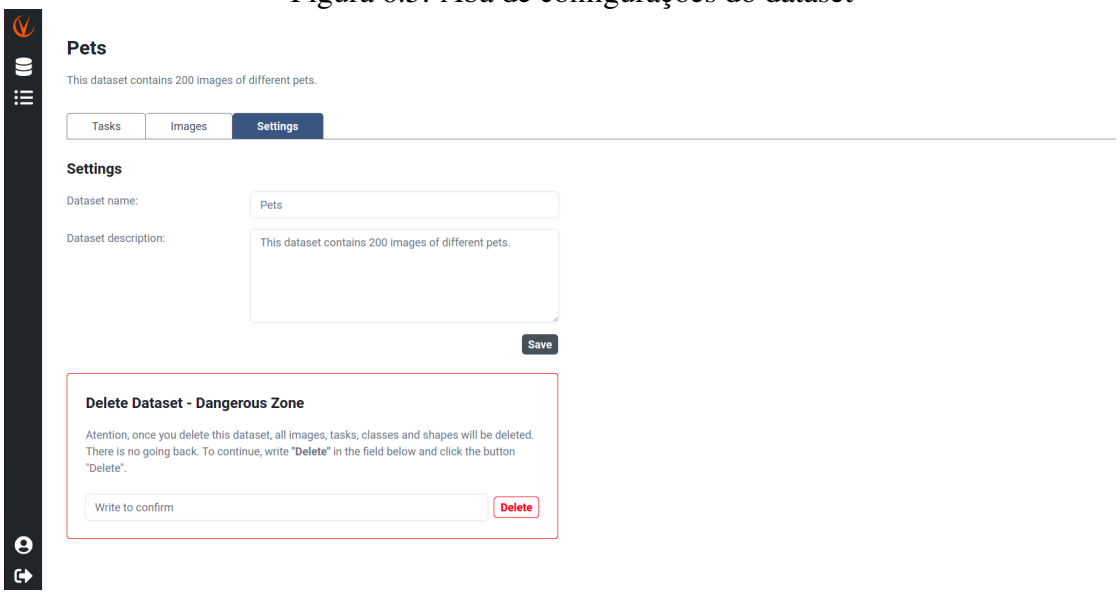
Figura 6.4: Aba de imagens do dataset



6.3.3 Aba Settings

Por fim, a aba Settings possibilita o gerenciamento do dataset. Através dela, o usuário pode atualizar o nome do dataset e sua descrição. Além de poder realizar a remoção do dataset, escrevendo Delete no campo de texto e confirmando ao clicar no botão Delete.

Figura 6.5: Aba de configurações do dataset



6.4 Tela da Task

A tela de anotação é a principal tela do sistema, porque é nesse local que o objetivo do projeto acontece. Nessa interface é possível navegar entre as imagens do dataset e realizar a anotação das mesmas.

Figura 6.6: Tela de anotação do CVLabel

No lado esquerdo, está a seção do canvas. Nessa seção, são carregadas as imagens para serem anotadas. Para facilitar a visualização das imagens, foram implementadas as funcionalidades de scale e translate. Portanto, é possível realizar um efeito de zoom e arrastar a imagem para a posição que permite uma melhor visualização para o anotador. É nessa seção que os shapes serão desenhados e seus pontos serão criados relacionados com os pontos da imagem em anotação. Quando o anotador navegar para outra imagem, o canvas é atualizado com a respectiva imagem e os shapes da nova imagem são renderizados.

Na barra lateral direita, existem três seções: Classes, Shapes e Images.

6.4.1 Seção Classes

A seção Classes permite que anotador crie, visualize e selecione as classes que pretende anotar. Com a classe criada, pode-se clicar sobre a mesma para selecioná-la ou utilizar o atalho "Shift"+ i, onde i é o índice da classe na lista de classes. (Esse atalho só existe para as primeiras 9 classes). A classe selecionada é identificada pelo item com cor

de fundo laranja.

6.4.2 Seção Shapes

A seção Shapes mostra todos os shapes criados na imagem atual. Para selecionar um shape, além de clicar sobre o mesmo na seção do Canvas, é possível realizar o clique nessa listagem. Assim como para as classes, o shape selecionado é identificado pelo item com fundo laranja.

6.4.3 Seção Images

Por fim, a seção Images é a lista de todas as imagens do dataset. Semelhante ao caso dos shapes, além de navegar pelas imagens usando as setas de navegação, é possível pular para uma determinada imagem clicando no respectivo item dessa listagem.

Com uma classe selecionada, o anotador pode criar um dos seguintes shapes: retângulo, polígono, ponto e linha.

- Retângulo: é composto por dois pontos (top-left e bottom-right) que representam o ponto superior esquerdo do retângulo e o ponto inferior direito do mesmo. A representação se dá desenhando o retângulo que contem esses pontos.
- Polígono: é composto por uma lista ordenada de pontos. Na visualização desse shape, se traça uma linha ligando os pontos com seu ponto seguinte e, no caso do último ponto, liga-se com o primeiro ponto da sequência.
- Ponto: é composto apenas por um ponto.
- Linha: é composto por uma lista ordenada de pontos. A visualização é muito parecida como o polígono. Entretanto, para esse shape, não existe uma linha ligando o último ponto com o primeiro ponto.

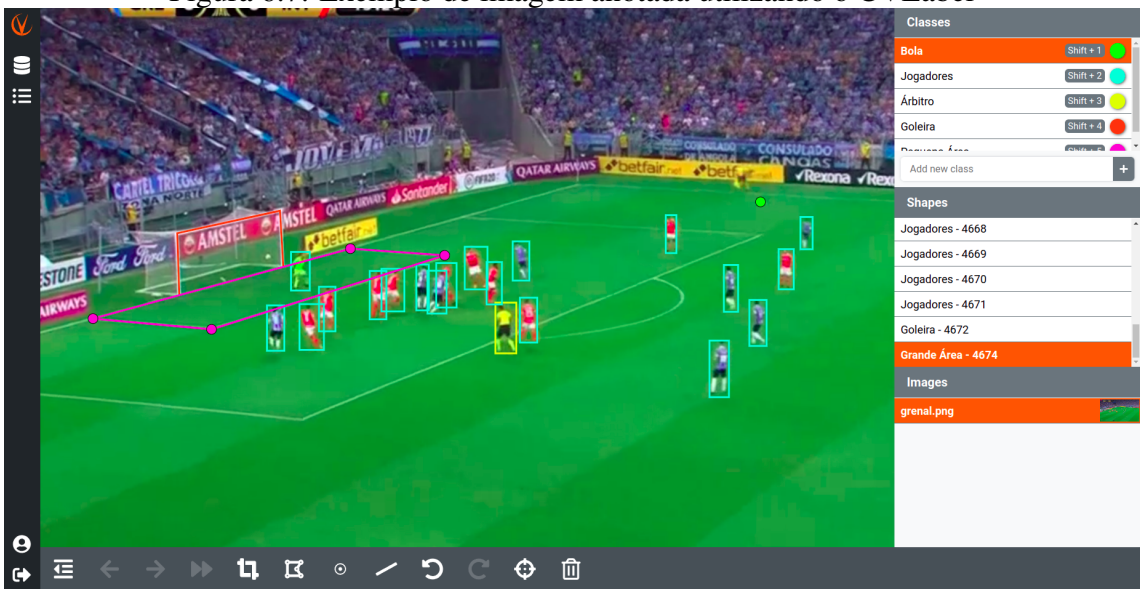
Após a criação do shape, o anotador pode editar o shape clicando sobre um de seus pontos e arrastando para a posição desejada a fim de obter um resultado mais preciso.

No rodapé, foram adicionados botões que permitem realizar algumas funcionalidades:

- Voltar a tela do dataset
- Ir para imagem anterior (Atalho: seta esquerda)

- Ir para a próxima imagem (Atalho: seta direita)
- Ir para próxima imagem sem anotações
    - Com objetivo de facilitar o trabalho em equipe, essa função permite saltar para a próxima imagem do dataset que ainda não possui shapes criados.
- Desenhar retângulo (Atalho: Ctrl + R)
- Desenhar polígono (Atalho: Ctrl + N)
- Desenhar ponto (Atalho: Ctrl + P)
- Desenhar linha (Atalho: Ctrl + L)
- Desfazer (Atalho: Ctrl + Z)
- Refazer (Atalho: Ctrl + Y)
- Enquadrar (Atalho: Ctrl + F)
    - Essa opção permite ajustar a imagem ao canvas de forma que a imagem tenha o máximo tamanho possível dentro do espaço.
- Remover shape selecionado (Atalho: DEL/Backspace)

Figura 6.7: Exemplo de imagem anotada utilizando o CVLabel



Na imagem acima, se realizou uma demonstração de anotação com todos os tipos de shapes. Na barra lateral, pode-se perceber a utilização das seguintes classes: Bola, Jogadores, Árbitro, Goleira e Pequena Área. Cada classe possui sua cor correspondente e o shape desenhado adquire a cor de sua respectiva classe.

Então, no canvas, é possível verificar que a bola foi desenhada utilizando um shape

do tipo Ponto, os jogadores e o árbitro foram anotados utilizando shapes do tipo Retângulo, a goleira foi desenhada utilizando um shape do tipo Linha e por fim a pequena área foi desenhada utilizando um shape Polígono.

6.4.4 Observações Técnicas

Para melhorar o desempenho dessa tela, foi necessário se atentar ao carregamento das imagens. Como os datasets podem possuir milhares de imagens, simplesmente adicioná-las na seção Images resultaria em milhares de downloads muitas vezes desnecessários. Para resolver esse problema, se utilizou a biblioteca citada no capítulo 2 para carregamento lazy. Dessa forma, as imagens só são requisitadas quando forem aparecer na tela.

Porém, esse ajuste transformou a troca da imagem atual mais lenta. Isso aconteceu porque, como o navegador não fez a requisição pela imagem com antecedência, a mesma ainda não estava disponível e o download precisa ser feito quando a troca é solicitada.

Para otimizar essa situação, se desenvolveu um comportamento que realiza um scroll na seção Images para a imagem atual. Dessa forma, as imagens imediatamente antes e depois da imagem atual aparecem na seção Images, disparando o carregamento dessas imagens pelo navegador. Dessa forma, quando o usuário trocar para uma dessas imagens o carregamento já deve ter sido realizado e a resposta será praticamente instantânea.

6.5 Tela de Estatísticas

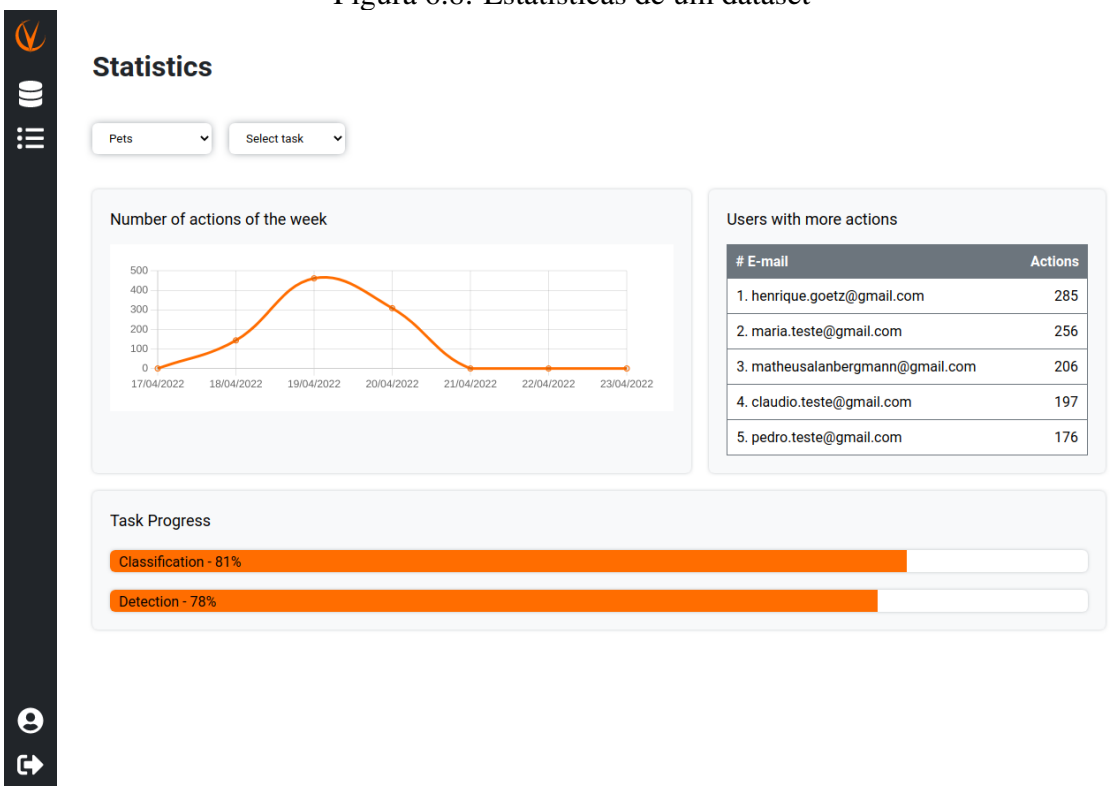
Como o objetivo de fornecer uma visualização do andamento das tarefas de anotação e controlar o histórico do trabalho, se desenvolveu uma tela de estatísticas. Essa tela permite dois tipos de análise: gráficos e histórico.

A visualização em gráficos visa de apresentar um resumo sobre o trabalho que está sendo realizado sobre uma entidade dataset ou task. Para definir qual entidade o usuário pretende acompanhar, são disponibilizados dois filtros: dataset e task.

Preenchendo somente o dataset, o resultado carregado nos gráficos estará representando todas ações sobre aquele dataset. Por outro lado, preenchendo também o seletor de task, os gráficos serão carregados utilizando apenas as ações realizadas na task seleci-

onada.

Figura 6.8: Estatísticas de um dataset



Já a visualização em tabela permite realizar uma busca com filtros específicos para analisar as ações realizadas sobre entidades da plataforma. Nessa visualização é possível realizar análises mais específicas como:

- Ações de criação de shapes;
- Ações de remoção de imagens;
- Ações de atualização de shapes durante a última semana;
- Ações realizadas por um determinado usuário nos dois últimos dias;
- Ações realizadas por um determinado usuário em uma determinada task;

Para isso, se foram disponibilizados mais os seguintes filtros: tipo de ação, usuário, imagem e intervalo de tempo.

Figura 6.9: Estatísticas de uma task

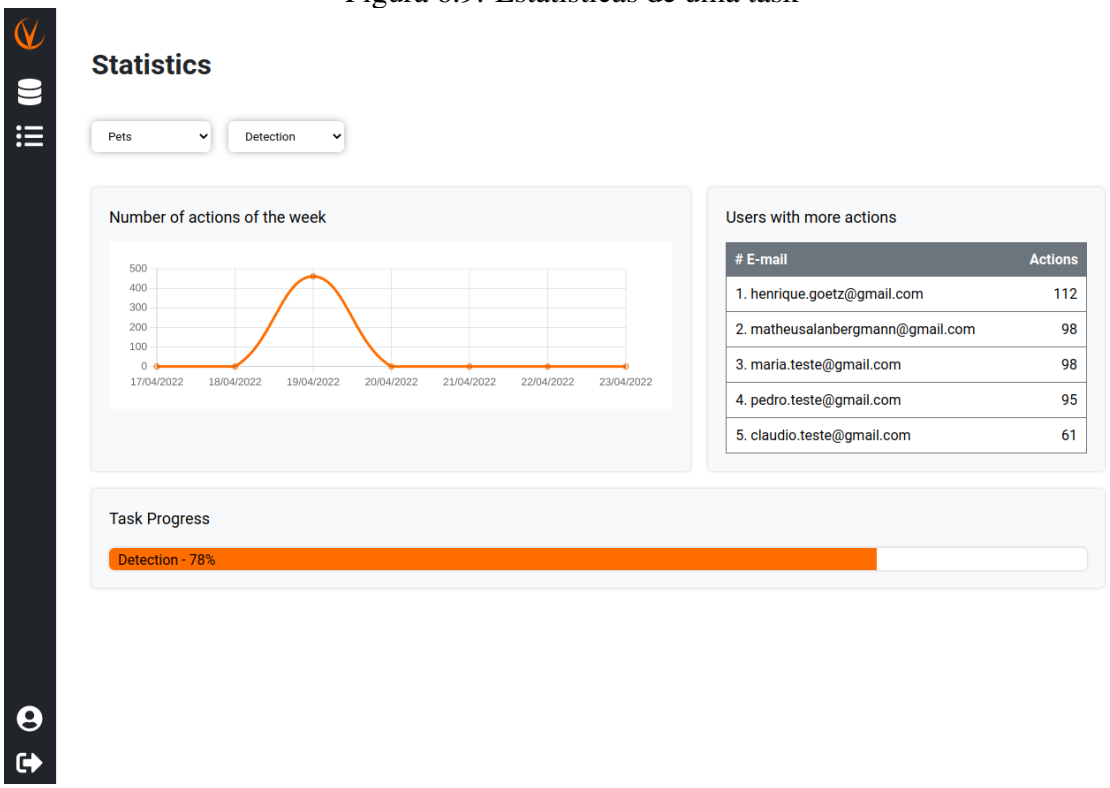
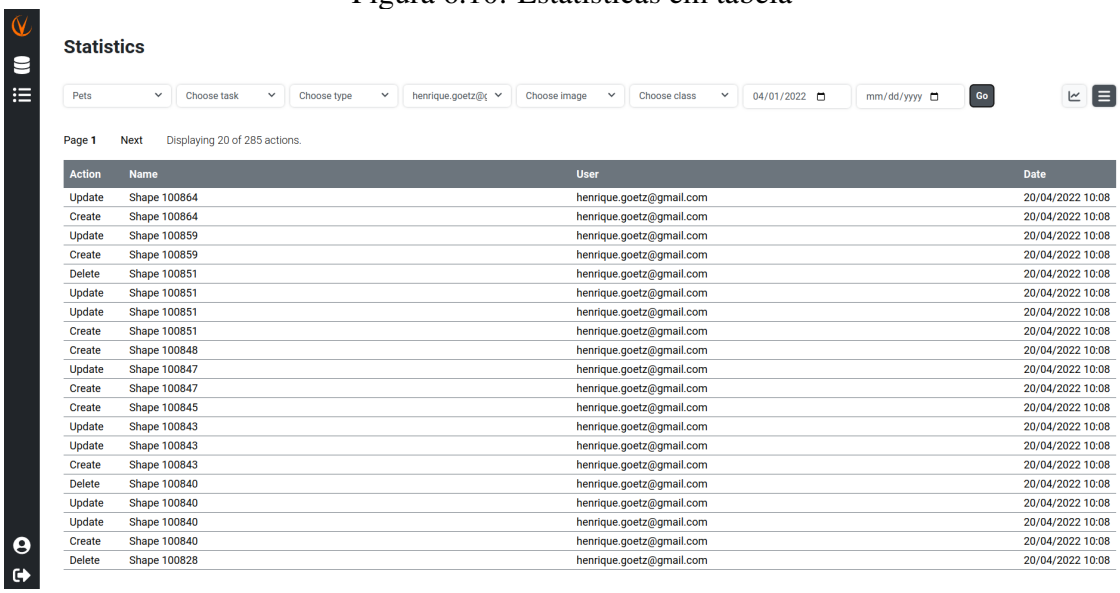


Figura 6.10: Estatísticas em tabela



7 CONCLUSÃO

Conforme pode ser percebido ao longo dos últimos capítulos, a plataforma evoluiu bastante em relação os requisitos iniciais do projeto. Isso porque, naquele momento, como o projeto pertencia a um trabalho da faculdade, o escopo precisou ser específico. Entretanto, ao término do semestre, se confirmou que, embora a ferramenta estivesse funcional, as funcionalidades disponíveis ainda não contemplavam todas as necessidades para substituir a ferramenta em uso na empresa.

Visto que a equipe já esperava esse acontecimento, o planejamento era seguir o desenvolvimento adicionando cada vez mais funcionalidades para suprir os requisitos da empresa e evoluir o sistema para o mercado. Foi dessa forma que a plataforma incorporou os novos shapes do tipo Ponto, Polígono e Linha, telas de análise de estatísticas, melhorias de usabilidade, entre outras atualizações. A empresa seguiu utilizando o CVLabel de forma parcial. Porém, nos últimos meses, o uso do sistema tem aumentado e, com isso, a partir de uma reunião com o COO da empresa, foi consolidada uma cobrança mensal. Espera-se que, nos próximos meses, o CVLabel se torne a principal ferramenta de rotulação da empresa.

Como limitações, o CVLabel ainda não permite criação de contas de forma automática. Atualmente, é necessário realizar o cadastro do usuário pelo dashboard da Amazon. Além disso, o gerenciamento de workspaces também está pendente de desenvolvimento. Para associar um usuário a um workspace, é necessário fazer a associação de forma manual. Como próximos trabalhos, a equipe planeja solucionar esses pontos permitindo gerenciamento de contas e workspaces pela plataforma. Além disso, para facilitar o controle das tarefas, pretende-se permitir associar tasks com usuários. Estão previstas mais funcionalidades referentes a usabilidade como: permitir edição de múltiplos shapes, adicionar opção de duplicar shape, adicionar opção de mover shape e permitir manipular brilho e contraste. Ainda, para melhorar a sincronização das tarefas, pretende-se incluir websockets para exibir notificações e atualizações sem interação do usuário. Por fim, estuda-se a possibilidade de acrescentar anotações automatizadas.

REFERÊNCIAS

CVLabel - A secure highly customizable labeling tool, 2022. Disponível em: <https://cvlabel.com>

Amazon. Amazon Relational Database Service (RDS), 2022. Disponível em: <https://aws.amazon.com/pt/rds/>

Amazon. AWS Lambda, 2022. Disponível em: <https://aws.amazon.com/pt/lambda/>

Amazon. Amazon API Gateway, 2022. Disponível em: <https://aws.amazon.com/pt/api-gateway/>

Vue.js. The progressive JavaScript Framework. Disponível em: <https://vuejs.org/>

Sass. Sass Basics. Disponível em: <https://sass-lang.com/guide>

Juszczak J., vue-chartjs, MIT. Disponível em: <https://vue-chartjs.org/>

Chart.js, MIT. Disponível em: <https://www.chartjs.org/>

Python. Documentação. Disponível em: <https://docs.python.org/3/>

Labelme - Image Polygonal Annotation with Python. Disponível em: <https://github.com/wkentaro/labelme>

CVAT - Computer Vision Annotation Tool. Disponível em: <https://www.cvat.ai/>

SuperAnnotate - The ultimate training data platform for AI. Disponível em: <https://www.superannotate.com/>