Bilkent University

Department of Computer Engineering

Senior Design Project

AnnoHub

Project Specifications Report

T2331

Emirkan Derköken, Bora Yılmaz, Dağhan Ünal, İlker Özgen, Onurcan Ataç

Supervisor: Shervin R. Arashloo

Table of Contents

1. Introduction 1

1.1 Description 2

1.2 Constraints 3

1.2.1 Operational & Implementation Constraints 3

1.2.2 Economic Constraints 3

1.2.3 Legal & Ethical Constraints 4

2. Design Requirements 5

2.1 Functional Requirements 5

2.1.1 Data Ingestion and Partitioning 5

2.1.2 Data Labeling and Annotation 5

2.1.3 Quality Assurance 5

2.1.4 Data Aggregation and Retrieval 6

2.1.5 User Management and Access Control: 6

2.1.6 Reporting and Analytics: 7

2.1.7 Notification and Communication: 7

2.2 Non-Functional Requirements 7

2.2.1 Scalability and Performance: 7

2.2.2 Data Privacy and Security: 7

2.2.3 System Availability and Reliability: 8

2.2.4 Extensibility and Modularity: 8

2.2.5 Interoperability: 8

2.2.6 Testability: 8

3. Feasibility Discussion 9

3.1 Market and Competitive Analysis 9

3.2 Areas of Use for AnnoHub 10

References 11

1. Introduction

In the burgeoning field of machine learning and artificial intelligence, high-quality labeled data is crucial for building accurate models. However; large-scale data labeling poses challenges particularly in ensuring accuracy and consistency. This is paramount as it directly impacts the robustness and precision of developed models. Achieving a reliable and uniform data annotation for extensive datasets demands meticulous methodologies and stringent quality control. Additionally; handling sensitive data adds complexity, requiring a delicate balance between having the dataset annotated and safeguarding personal privacy and corporate secrets.

Beyond the technical intricacies, sourcing a skilled workforce for data annotation poses a fundamental challenge, requiring careful consideration from recruitment to ongoing training. In contrast to the nuanced challenges of acquiring skilled labor for complex tasks, finding unskilled workers for simple data annotation is also difficult due to the sporadic nature of the work. Even though the machine learning industry at large exhibits a high demand for labeled data; individual companies encounter intermittent demands, characterized by urgent requirements during development phases. These spikes in demand are unpredictable and occur sporadically throughout the life-cycle of a business, with periods of intense need followed by times when data labeling is not required. As a result; companies are reluctant to establish in-house teams, given the irregular demand, making it economically impractical. Simultaneously, potential workers are deterred by the lack of stability in roles, resulting in a scarcity of individuals willing to perform straightforward data annotation tasks.

AnnoHub is designed to address the main challenges of data annotation explained above while offering a high-quality user experience through easy to use and user-friendly interfaces. Packed with many features that enhances the quality of life for both annotators and companies, ensuring an efficient and effective annotation process.

1.1 Description

AnnoHub serves as a comprehensive data annotation platform, facilitating two distinct modes of operation. Users can publicly publish unlabeled datasets, engaging freelance annotators to perform annotations collaboratively. In this operational mode, publishers are relieved of the burden of sourcing the necessary workforce and their data is annotated at a more cost-effective rate compared to the alternative of hiring in-house annotators. Additionally, for the freelance annotators, this option presents a new source of side income channel that can be utilized by almost any individual. Alternatively, companies have the option to create private data annotation projects with their datasets, utilizing the platform as a sophisticated labeling tool to enlist annotators of their choosing.

In order to enhance the accuracy and consistency in annotations, particularly those conducted by third-party annotators in public datasets, AnnoHub incorporates specific features. These include techniques for validating the integrity of the annotators' work, such as the cross-validation of the same data across multiple annotators, self-validation requiring annotators to label the same data multiple times and the integration of known-annotation trick data within the dataset aiming to identify potential contradictions both within and across annotators. Furthermore; through the utilization of unsupervised machine learning algorithms for mislabel prediction, the labeling error rate is aimed to be reduced. Simultaneously; by the implementation of fraud detection algorithms, the negative effects of malicious use are aimed to be minimized. These measures and the information they are going to present to the companies collectively enhance the reliability and quality assurance of the annotation process on our platform.

Finally, owing to the AnnoHub's system architecture prioritizing enhanced security, in cases where companies choose to create a private data annotation project, the data is going to bypass our servers entirely. Instead, it is directly shared with the annotators, ensuring the safeguarding of company secrets and adherence to data protection laws not through bonds, but through the system design.

1.2 Constraints

1.2.1 Operational & Implementation Constraints

- AnnoHub will be a web application.
- GitHub will be used for team collaboration, code implementation, and version control.
- Python Flask will be used for the backend implementation and React will be used for the frontend implementation. This was found suitable for agile development.
- MongoDB will be used for data annotation because a NoSQL database allows flexibility in handling frequently changing schemas. PostgreSQL will be used to store user information since it performs better when dealing with bigger datasets.
- A cloud service (such as Amazon Web Services) is planned to be used for hosting.
- Electron JS will be used for the on-premise desktop application implementation because it is interpreted and its HTML-CSS-JS implementation is beneficial for agile development.
- Customers are expected to have decent servers to ensure data privacy while keeping the application functional.
- There should exist customers and labelers for AnnoHub's operational purposes.
- The data to be labeled does not have to be hosted in AnnoHub. However; in the event of customers choosing to keep their data private, certain features -such as mislabel prediction- are either going to be disabled or the source code will be shared with the customer through the on-premise application since both cannot co-exists.

1.2.2 Economic Constraints

- The main economic constraint for AnnoHub is the expenses that are associated with the server maintenance. This includes costs related to hosting, cloud services, and storage. Server resources must be efficiently managed to control these expenses and ensure that AnnoHub is functioning smoothly. The storage costs are expected to be higher than the others and should be given importance.
- Another economic constraint is the costs related to financial transactions, like money transfers to labelers. These transactions involve fees and currency conversion charges, which need to be considered in the financial planning of AnnoHub.

- Initial investment is another significant economic constraint. While the project can operate with a moderate initial investment, future financial planning is essential. Adequate funding is needed not only for the initial setup costs but also for the ad expenses and other ongoing needs.

1.2.3 Legal & Ethical Constraints

- A mandatory objective of AnnoHub is to stay observant to the Turkish Personal Data Protection Law (KVKK).
- User-provided information must not be leaked to any third-party organization unless explicitly approved by the user and is within KVKK's scope.
- The user's stored data should be destroyed within a time interval or when the user requests the data's destruction. The detailed policies on data storage and destruction must be provided to the user as stated in KVKK.
- For the data to be stored and processed, several clarification and express consent texts must be given to the user. The methodologies used to store/process user's data should be introduced to the user together with the other are legal and ethical considerations. The user must approve these texts with his own consent before using the application.
- The data must be stored/processed exactly the way that was introduced to the user which gets the user's approval.

2. Design Requirements

2.1 Functional Requirements

2.1.1 Data Ingestion and Partitioning

- Support for various data types including text, images, audio, video, and 3D point clouds. More can be added in the future.

- Data partitioning to ensure privacy and manageability. If chosen so, data can also be partitioned manually by clients, where the data is sensitive even in smaller subsets. Otherwise, automatically partition data.

- Clients can choose to store their data directly to our servers or hide it so that the data can be sent to the annotators directly bypassing our servers while still using our front-end and back-end logic.

2.1.2 Data Labeling and Annotation

- User-friendly interface for labeling data.

- Support for various labeling tasks such as classification, object detection, segmentation, etc.

- Classification algorithms that help identify the category or class of an object within an image.

- Object detection algorithms that help identifying and locating the objects within an image by drawing a box around each detected object.

- Semantic segmentation process that labels each pixel in an image with a class.

- Instance segmentation process that differentiates between different instances of the same class such as two different types of cars.

2.1.3 Quality Assurance

- Automated and manual quality checks to ensure label accuracy

- Real-time monitoring and reporting of labeling accuracy and labeler performance.

- Ability for supervisors to review and correct labels.

- Ability for supervisors to rate labelers based on their performance.

- Qualification tests and exams that make sure each labeler is qualified for labeling the data provided. Different tests can be provided from the clients based on the requirements of the data.

- Fraud detection algorithms that dig through annotators to detect any frauds and tag labelers in the danger zone for further view.

- Cross validation and self-validation techniques that train the model in some subsets and test it on the others.

- Mislabel prediction using unsupervised machine learning algorithms that aims to identify possibly mislabeled data in datasets.

- Usage of trick data to determine false labelers to further validate their labels.

2.1.4 Data Aggregation and Retrieval

- Tools to aggregate labeled data from various labeling tasks, enabling an organized collection of labeled datasets for further use.

- Efficient data retrieval mechanisms to allow quick access to the labeled data, aiding in timely analysis and usage.

- Indexed data storage to expedite the search and retrieval of specific data sets or data points and features to export aggregated data in various formats for further analysis or for sharing with stakeholders.

- Ability to run queries to filter and retrieve specific subsets of data based on certain criteria and filters.

2.1.5 User Management and Access Control

- Role-based access control to define permissions for different user roles.

- Secure authentication and authorization mechanisms to ensure data privacy

- Trackable user actions to ensure accountability.

2.1.6 Reporting and Analytics

- Dashboards to monitor project progress, labeler performance, and data quality.
- Exportable reports for sharing with stakeholders.
- Analytics tools to derive insights from labeling data and improve project outcomes.

2.1.7 Notification and Communication

- Automated notifications for task assignments, completions, and quality issues, communication tools for collaboration and discussion among labelers and supervisors.
- Feedback mechanism for labelers to report issues/bugs and suggest improvements.

2.2 Non-Functional Requirements

2.2.1 Scalability and Performance

- Ability to handle and optimize increasing volumes of data without compromising performance.
- Scalable workforce management to scale labeling capacity up or down based on demand.
- Optimization tools to further enhance performance.

2.2.2 Data Privacy and Security

- Encryption of sensitive data at rest and in transit, data masking and anonymization tools to protect privacy.
- Taking industry standards, state laws and regulations into account regarding data privacy and security.

2.2.3 System Availability and Reliability

- Reliable data storage and backup solutions to prevent data loss.
- Error handling and recovery mechanisms to ensure system stability and reliability.

2.2.4 Extensibility and Modularity

- Modular system design to allow for the addition of new features and components more easily.
- Extensibility to easily integrate with other systems and tools.
- API documentation for developers to extend and integrate the system.

2.2.5 Interoperability [1]

- Standards-compliant data formats to ensure interoperability with other systems.
- SDKs for integration with external systems and data sources.
- Compatibility with common data labeling and machine learning tools.

2.2.6 Testability

- Testing tools and frameworks to validate system functionality, stability, reliability and performance.

[1] One important thing to note is that in the context of data labeling, annotation format is crucial. It specifies how the labels or annotations are structured and associated with the data being labeled. For instance, in object detection tasks, annotations might include coordinates of bounding boxes and class labels.

3. Feasibility Discussion

3.1 Market and Competitive Analysis

The utilization of AI and machine learning methodologies is expanding across diverse sectors, leading to an increased demand for high-quality labeled datasets. As of 2022, the global machine learning market attained a valuation of USD 36.73 billion, with an anticipated compound annual growth rate (CAGR) of 34.8% from 2023 to 2030 [1]. Consequently, it is plausible that existing tools in the market exhibit similar functionalities to our application. To address this, we conducted a comprehensive market analysis, identifying key competitors for AnnoHub. Following the research, three prominent tools -Labelbox, Label Studio, and Amazon Sagemaker- emerged as primary contenders due to their widespread usage and analogous functionalities.

A primary challenge addressed by our application is the facilitation of data labeling without the necessity of sharing sensitive information with external servers. For instance, Labelbox mandates customers to upload data on Google Cloud Services, storing it compulsorily in the US [2]. Catering to clients unwilling or unable to share data beyond their national borders (e.g., due to KVKK laws in Turkey), AnnoHub offers a solution by maintaining only references to sensitive data on its servers and sharing the actual data exclusively with designated labeler employees. Consequently, AnnoHub ensures privacy by design, alleviating the need to entrust external entities with sensitive information.

Furthermore, for clients with insensitive data, AnnoHub aims to establish a public workforce, allowing customers the option to label their data through this workforce. Customers can select qualified public labelers via proficiency tests tailored to the labeling task. While Labelbox and Amazon Sagemaker partially support outsourcing labeling, neither provides a comprehensive public workforce system. Although Amazon employs a similar system known as Amazon Mechanical Turk, our project's innovation expert, Ahmet Kocamaz, expressed dissatisfaction with the service due to inadequate information about labeled data metrics. Mechanical Turk is also notorious for not returning to the applications of the public. AnnoHub strives to offer both cost-effective labeling and job opportunities through this system, a distinctive feature absent in competitors.

In addition to these capabilities, although AnnoHub does not support model training like some competitors, it aims to match their labeling functionalities by providing interfaces for various data types. Moreover, AnnoHub employs machine learning algorithms, including cross-validation, self-validation, and fraud detection algorithms. These mechanisms, coupled with the public workforce system, underscore AnnoHub's commitment to ensuring efficient and effective labeling, ultimately contributing to heightened customer satisfaction.

3.2 Areas of Use for AnnoHub

AnnoHub holds the potential for widespread adoption across diverse industries and academic institutions alike. The implementation of the privacy by architecture principle, along with the provision of private project options, positions AnnoHub as an ideal solution for companies necessitating the processing of sensitive data in their labeling endeavors. Furthermore, the public workforce option introduced by AnnoHub constitutes a valuable asset for both industry and academia, addressing the demand for cost-effective and high-quality labeling services. Notably, startup companies are anticipated to derive significant advantages from the availability of public workforce services. In essence, AnnoHub emerges as a valuable asset in any industry that involves a data labeling process, offering flexible and advantageous data labeling options.

References

[1] “Machine learning market size, share & growth report, 2030,” Machine Learning Market Size, Share & Growth Report, 2030, https://www.grandviewresearch.com/industry-analysis/machine-learning-market#:~:text=Report%20Overview,how%20businesses%20and%20people%20operate (accessed Nov. 16, 2023).

[2] “Access, storage, and security,” Labelbox docs, https://docs.labelbox.com/docs/access-storage (accessed Nov. 16, 2023).