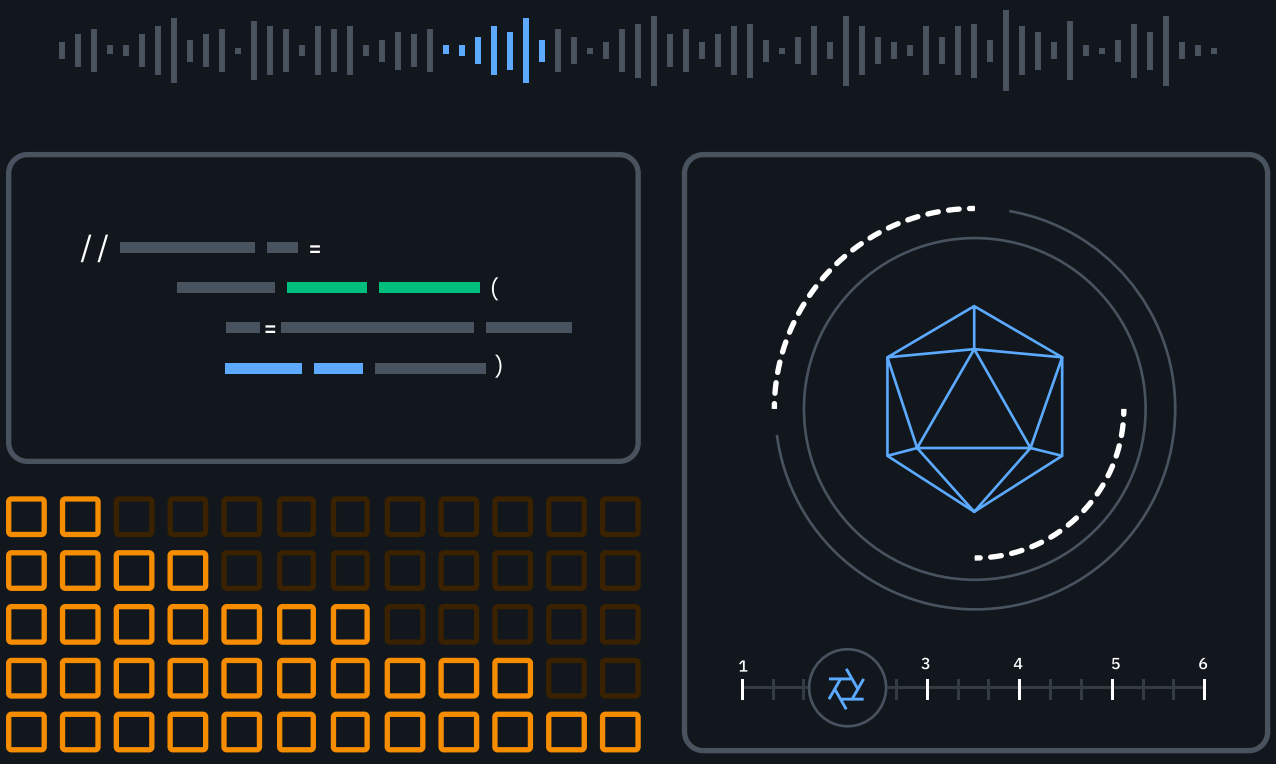
Labelbox

The guide to saving time when creating AI data



00

Content

01 Introduction

One thing all ML teams share is the need to make their data useful for their models. Wading through a vast amount of unstructured data to accurately annotate assets requires a tremendous amount of patience, organization, and time. Nearly 96% of companies encounter delays getting to production and 78% of machine learning projects stall before deployment. Moving faster is the top pain point we hear from customers; it controls their ability to meet project goals and gain advantage over competitors. Speed is essential to not only get models to production but do so on timelines that meet these increasingly aggressive expectations.

This paper covers six key time-saving practices for ML teams to implement when handling AI data, based on the experiences of hundreds of AI teams across industries. Adopting these practices can help teams improve collaboration between teams, gain better access to their data, and utilize software designed especially to optimize ML workflows. These benefits in turn unblock key barriers and speed up overall processes and capabilities for a quicker path to production AI. You'll also discover some advanced techniques that can improve efficiencies across your AI efforts. Learn how using model-assisting labeling can speed up your labeling workflow, and how, by employing active learning to find and prioritize the right data for your models, you can exponentially speed up your ML lifecycle.

[1]Algorithmia 2020 State of ML Survey

02 Annotate faster with a dynamic queuing system

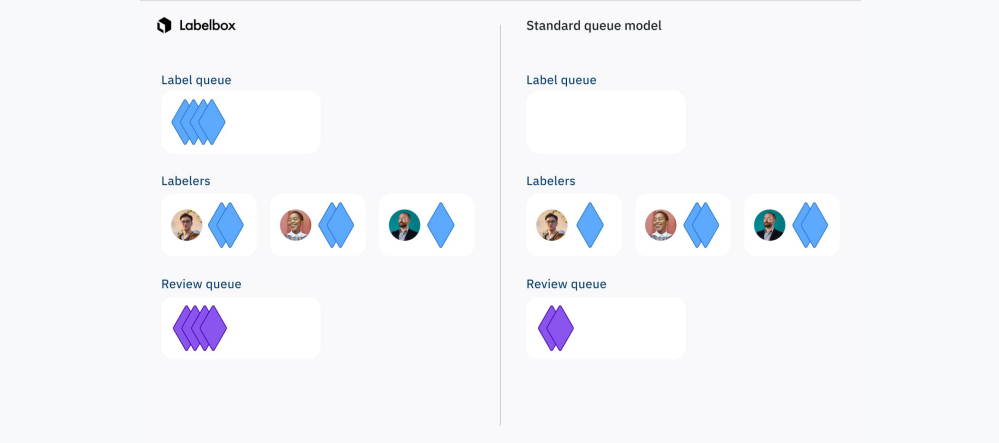
Combined with improved communication and collaboration, AI teams can additionally benefit from speedier annotations by employing a dynamic queuing system that routes labeling tasks to the right team members so they are never sitting idle. With dynamic queuing, teams are able to create labeling workflows where low confidence predictions are visible and prioritized to their data scientists and labelers. This allows you to more quickly analyze and correct labels.

In a dynamic queuing system, once a dataset is attached to a project, AI teams will want to enqueue the data rows in that dataset. Once the data rows enter the labeling queue, they are distributed in batches to each active labeler, saving the team a significant amount of time. When an active labeler is assigned a batch of data rows, it automatically holds those data rows just for that labeler, so those same assets cannot be labeled by another team member.

Once a labeler begins a labeling session, the dynamic queuing system fetches two data rows from the labeler's assignments and executes them in the browser: one will be served up immediately to the editor and the second will be cached in memory to ensure a smooth transition after the first data row is submitted. The system continues pre-fetching the next queued asset during an active labeling session until all assets have been distributed.

Dynamic queuing allows you to save time by automatically routing labeling tasks to the right team members.

- Only active labelers will be assigned assets to annotate to eliminate duplication and waiting
- Dynamic queuing reserves multiple assets at a time to make sure your labeling team isn't waiting around in between assets
- The team shares a backlog that's automatically distributed among labelers and reviewers

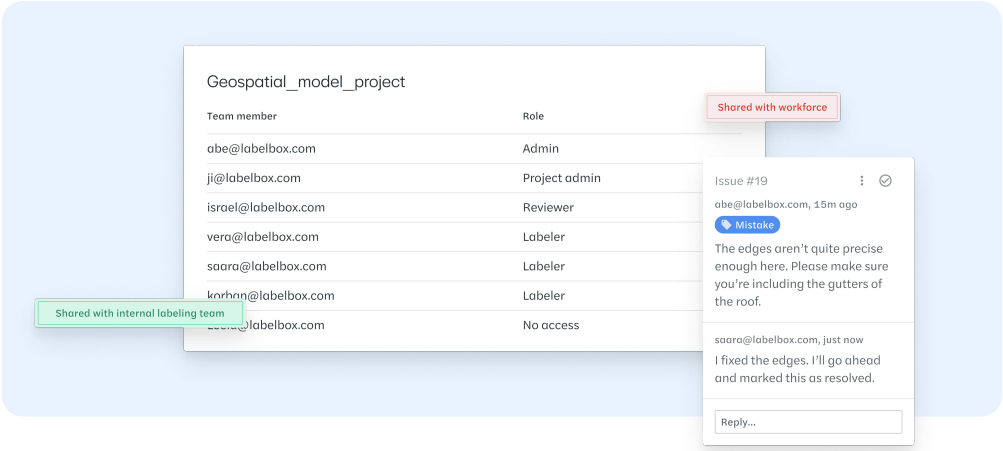


Click here to view the animation in a new window.
In the illustration above, Labelbox's dynamic queueing process is compared to a standard queue model. Without this automation in place, labelers will sit idle while waiting on their next task.

03 Improve communication, collaboration, and consensus between teams

Building a machine learning model involves many people, all working in tandem in pursuit of solving a problem. Many ML teams experience limited collaboration between these domain experts, labelers, product managers, and data scientists, and have no standardized plan for making decisions that affect the team's ML initiatives and goals. If your team faces similar challenges, here are a few practices to help you remove friction.

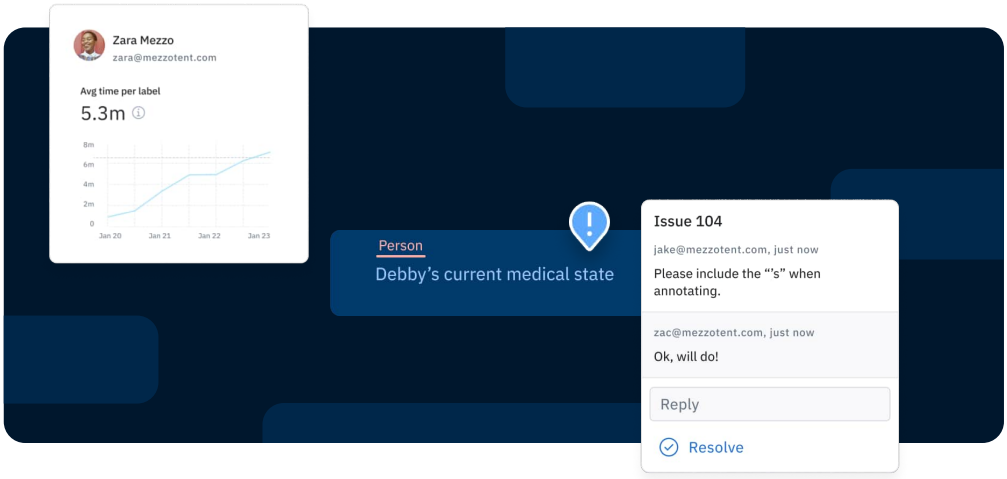
Oftentimes, AI teams experience friction communicating during an ML project because it can require a lot of jumping between platforms to address a specific issue. Communication shared over email or workplace communication tools about the status of AI data becomes hard to find again and prevents visibility to the other members of the team. Having an organized system to unite all your labelers during an annotation project can remove some of those hurdles and expedite other processes. For example, the most repetitive processes of setting up a project, sharing data sets, and inviting new members can be made easy in a single platform. While every organization has a different team structure, fostering seamless collaboration between data science teams, domain experts, dedicated external labeling teams, and anyone else you need on your labeling team can exponentially increase your productivity and save your organization time.



Efficient collaboration between data science teams, domain experts, and dedicated external labeling teams can be enabled by uniting the team in a single platform.

To enable better collaboration during the labeling process, teams should look to implement processes and dedicated tooling to make repetitive tasks effortless, such as adding and clarifying feedback on a labeled asset. In order to elicit and clarify feedback, the ability to quickly raise issues and add comments on a labeled asset provides a simple and reliable channel for escalating questions to reviewers or subject matter experts.

A labeler can create an issue to ask a question, submit the label completed to the best of their ability, and receive feedback and clarification during the label review process. This streamlined way of collaborating on labels provides teams with an expedited way to handle inevitable issues that arise during labeling.



With an issues and comments tool, you can leave a comment directly on an issue for your workforce or your internal labeling team by tagging them. They can reply back or mark it as resolved.

A leading digital health company providing AI-supported expertise for orthopedic surgery and musculoskeletal radiology leverages an issues and comments tool to streamline their communication on difficult labels. Since the medical implants they see in data assets are diverse and rare in the image datasets, finding common rules for annotations and explaining every edge to outsourced labelers became a massive hurdle. They leveraged an issues and comments tool to quickly clarify questions on edge cases and make these images available for AI training.

To improve consensus amongst team members, you can quickly drive toward an agreement as a team with a consensus tool that helps you arrive at a shared understanding of what good looks like. This tool allows you to automatically compare annotations on a given asset to all other annotations on that asset. Consistency is measured by the average of the label agreement between labelers. For example, when the consensus is set to 3, every label is grouped with two similar labels and the average of the three becomes the consensus score.

In one healthcare use case, doctors disagreed about diagnoses based on tissue imagery up to 40% of the time — and an ML model with that level of accuracy would have a hard time obtaining FDA approval. To get around the issue of differing diagnoses, the ML team needs to implement a quality tool like consensus, where each piece of training data is labeled by multiple expert physicians to account for their different perspectives and create higher quality ground truth.

The consensus tool works in real-time so you can take immediate and corrective actions towards improving your training data and model performance.

04 Utilize a programmatic approach for faster access to data

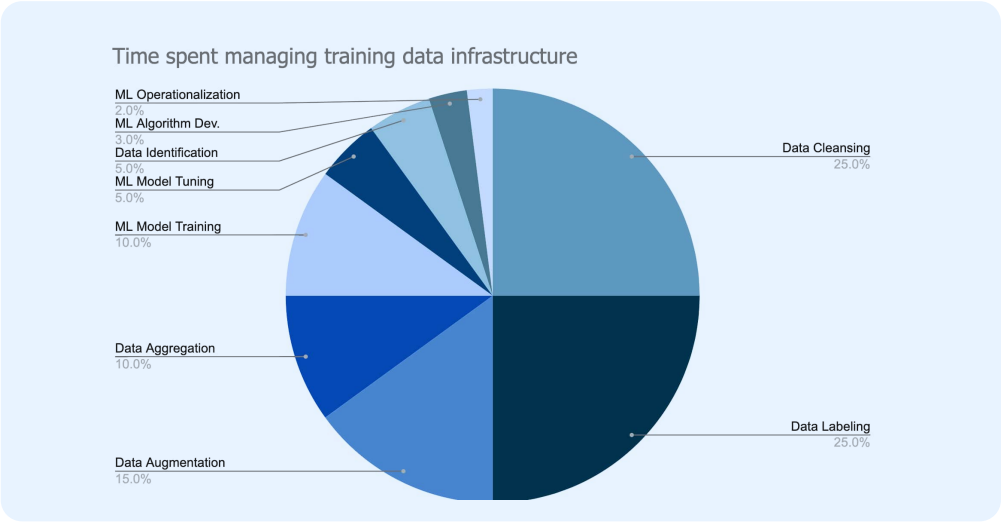
Teams can now speed up the data import and labeled data export process through the use of a programmatic approach via SDKs and/or APIs rather than slowly handling data transfer methods manually. By connecting your team's data and automating bulk operations, labeled data (including all the metadata, how the dataset was created, as well as labeler feedback) becomes much easier and quicker to manage and track as the complexity of your projects grows over time. This approach offers endpoint flexibility, so you can plug and play into your existing workflows without losing any valuable aspects of your training data.

One example of how this can be accomplished is through the use of a Python SDK, which offers more control over your data while simplifying and speeding up data import. For example, it simplifies the data import process so that you can use bulk DataRow creation. This process is asynchronous, so you don't have to wait for bulk creation to finish before continuing with other tasks (though you can wait if you want to). In some cases, you can also create projects and datasets programmatically, export labels, and add metadata to your assets all in an object-oriented way with a Python SDK — complete with all relationships between objects, speeding up your labeling workflow. By utilizing this approach, you can even export newly created training data automatically for an active learning workflow and adjust your labeling queue to focus on improving confidence in specific classes.

By using automated workflows and programmatic methods, you can stream data into your training data platform and push labeled data into model training environments like TensorFlow and PyTorch. This allows you to simplify your data import without writing and maintaining your own scripts. ML teams achieve significant time savings over the course of their MLOps lifecycle and improved performance throughout their labeling operations, model iterations, and more with the use of a Python SDK.

05 Leverage software designed for speed and flexibility

Many ML teams spend a considerable amount of time attempting to build out custom infrastructure for the training data that doesn't scale and leads to disjointed development efforts. According to research firm Cognilytica, ML teams spend up to 80% of their time building and maintaining training data infrastructure.



Source: Cognilytica

Building such custom infrastructure can be both difficult and expensive, potentially costing millions of dollars in engineering resources to build and maintain. As a time-saving alternative, many teams have found the answer in adopting a training data platform (TDP). A TDP is purpose-built software engineered for speed and ease of use in handling AI data. It allows teams to move faster with best practices already built in place, rather than home-grown tools and recreating the wheel. By incorporating a TDP, teams save exponential time on data preparation and labeling, so ML teams can spend more time on their core competency: building production models.

But not all TDPs are created equal. You can learn more about what TDPs offer ML teams in our TDP 101 guide, and how to choose the best one for your team by reading our buyer's guide.

This space, while still early, has competitive offerings designed to help teams reach production AI. However, they do vary greatly in capabilities and ease of use, so teams should be deliberate about what they are getting with their investment, so they don't unintentionally undo the time savings they've been searching for.

Teams save time when they are able to easily execute daily tasks. Software designed especially for creating and maintaining training data helps lower the cognitive load on labelers, who can be labeling for a significant number of hours.

Even on lower-spec PCs and laptops, performance becomes critical for professional labelers who are working in an annotation editor all day. Another place where you'll save time by using dedicated software for AI data is ensuring the solution is enterprise grade and enterprise ready. A platform that meets your organization's security requirements while providing quick setup options and simple onboarding will go a long way to speeding up adoption for your entire organization.

Lastly, you can save a significant amount of time by choosing a TDP that can handle growth as your labeling requirements scale. It is important to be able to configure your labeling editors to your exact data structure (ontology) requirements while being able to re-use and expand on ontologies and projects. Custom attributes, hierarchical relationships, and infinite nesting are key capabilities to tailor the labeling tools to your needs.

With an issues and comments tool, you can leave a comment directly on an issue for your workforce or your internal labeling team by tagging them. They can reply back or mark it as resolved.

By leveraging purpose-built software that's optimized for speed and flexibility, you can accelerate your entire ML lifecycle. A TDP provides a single source to unite your data, your team, and all the processes required to create high-quality training data and get ML models to production quickly and efficiently.

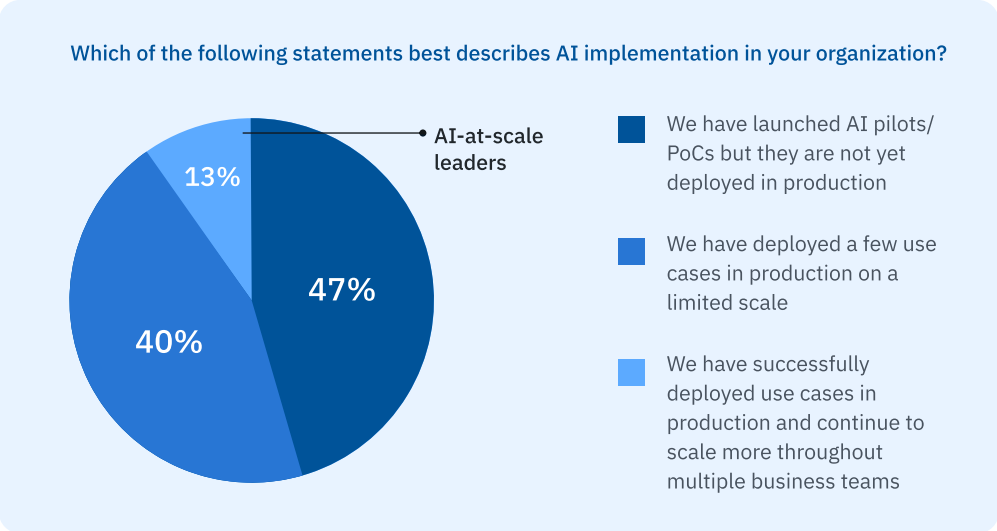
06 Incorporate automation

When organizations start looking to scale, automation becomes the necessary answer for faster time savings, quicker turnaround time, and optimizing human capital. According to CapGemini's research, only 13% of organizations have successfully rolled out multiple AI applications across teams. [2]

[2] The AI-Powered Enterprise, CapGemini Research Institute, 2020

Even on lower-spec PCs and laptops, performance becomes critical for professional labelers who are working in an annotation editor all day. Another place where you'll save time by using dedicated software for AI data is ensuring the solution is enterprise grade and enterprise ready. A platform that meets your organization's security requirements while providing quick setup options and simple onboarding will go a long way to speeding up adoption for your entire organization.

Lastly, you can save a significant amount of time by choosing a TDP that can handle growth as your labeling requirements scale. It is important to be able to configure your labeling editors to your exact data structure (ontology) requirements while being able to re-use and expand on ontologies and projects. Custom attributes, hierarchical relationships, and infinite nesting are key capabilities to tailor the labeling tools to your needs.



Source: Capgemini Research Institute, State of AI survey, March-April 2020, N=954 organizations implementing AI.

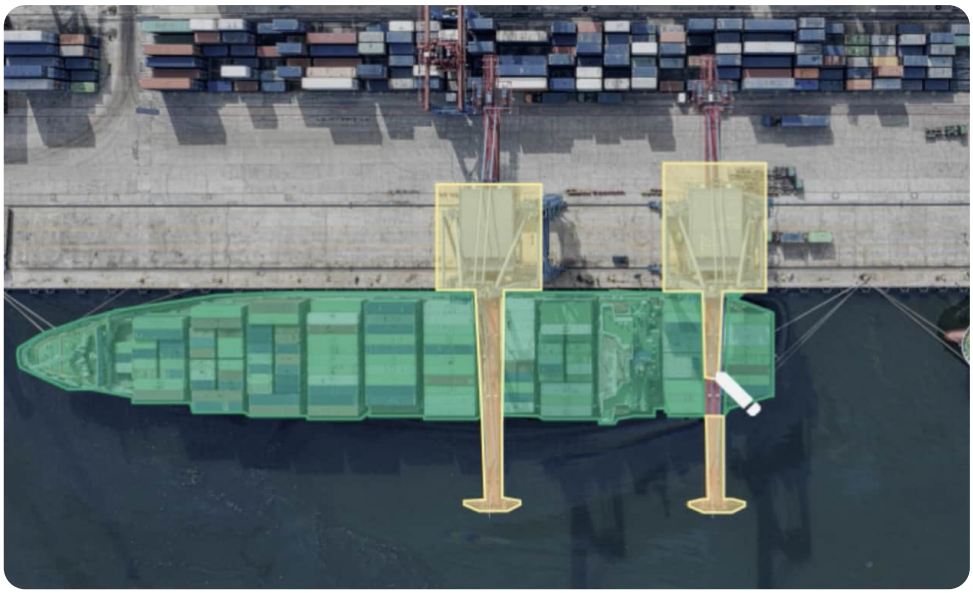
Accurately annotating the tens of thousands if not millions of assets necessary to achieve a comprehensive data set has blocked many promising AI initiatives from ever reaching production-ready models. That's why automation through model-assisted labeling is one of the simplest ways to reduce the waste of precious capital. Model-assisted labeling uses your own model to make labeling easier, more accurate, and faster. In some cases, we've seen ML teams save 50-70% on their entire labeling budget by utilizing this method.

Here are a few steps in a typical model-assisted labeling workflow:

1. Quickly integrate your model: Upload pre-labeled assets in bulk with a few lines of code.

```
upload_job = project.upload_annotations(
    name="upload_annotation_job_1",
    annotations=predictions)
```

2. Accelerate human annotation: Give labelers pre-labeled assets so they can confirm, reject, or edit annotations rather than labeling from scratch.



With model-assisted labeling, labelers don't have to manually label each asset. Instead, a labeler can simply review a prediction and accept it, reject it, or add some manual adjustments if needed, as shown above.

3. Narrow the field of view: Speed up labeling by drawing attention to the areas of low confidence, so labelers don't need to analyze the entire asset.

Rather than a labeler manually scanning an entire asset and determining labels, your model can narrow the field of view and make predictions. This efficient workflow reduces the strain on labelers and opens up their bandwidth to make more impactful decisions.

4. Export labels: Send labeled assets to your model in bulk or real time via webhooks.

```
project = client.get_project("<project_id>")
url = project.export_labels()
    print(url)
```

5. Spend time where you're needed most: Automate labeling where your model confidence is high and spotlight assets where model performance is low. You can dramatically improve model performance by focusing labeler time on more examples of data with low confidence predictions.

With each iteration, the amount of tedious work required by humans decreases even as the amount of training data increases. So, labeling teams can devote more time to low-confidence predictions, as shown above on an asset, and improve their model's performance.

Case study: How Sharper Shape utilizes model-assisted labeling in the real world with utility inspections

Sharper Shape creates technology for safe, efficient transmission and distribution solutions for utilities by using drones to perform utility inspections. The company uses computer vision models in advanced aerial sensor systems to power the automatic collection and analysis of unmanned aerial inspection data.

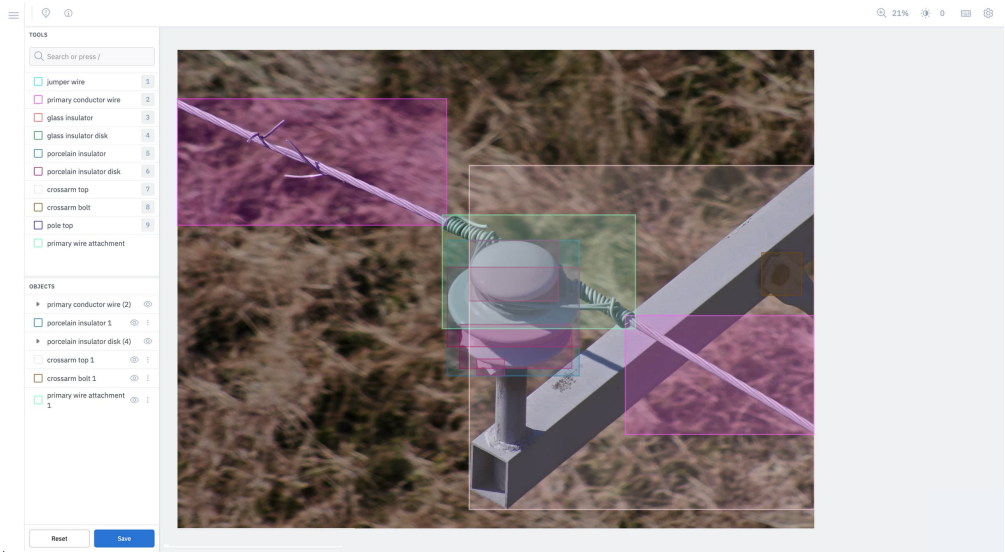
A common use case for their technology is the identification of dangerous setups with electric wirings, such as vegetation growing too close, broken insulators, and more, so that utility companies can find and address potential hazards.

The Sharper Shape team typically relied on heavily manual workflows and experimented with open-source labeling tools that did not provide the required amount of configuration needed for their needs. With Labelbox, the team could connect their raw data via a simple API. The team at Sharper Shape also utilized the collaboration features to enable rapid onboarding, training, and throughput for both internal and skilled external labelers to work together in one centralized environment.

The Sharper Shape team is now accelerating their labeling process even more with model-assisted labeling, which allows teams to import their model and address edge cases.

“With the streamlined design of Labelbox, we are able to cut costs on labeling by as much as 50% while maintaining the highest quality in our training data and get to training our models faster. With human-in-the-loop model-assisted labeling, we expect another huge reduction in time and costs to the labeling process.”

Edward Kim, Data Analyst at Sharper Shape.



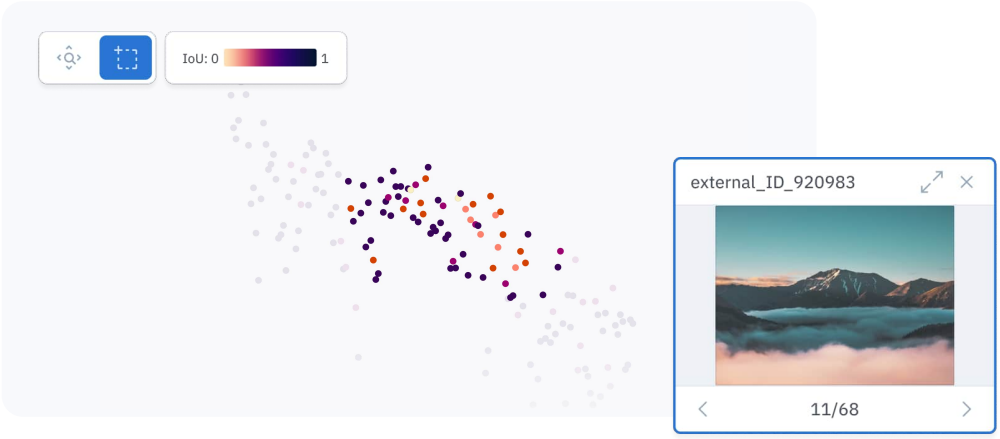
Distribution line components with chipped porcelain insulators and spliced conductors are shown in Labelbox.

“After a preliminary model is trained, we can run a loop to generate labels from our model’s inference and feed those back into Labelbox, effectively cutting the labeling load of our labelers to that of reviewing for false positives,” said Kim. “That allows us to increase our capabilities and model accuracy exponentially with respect to time for the amount of components and defects we can detect and classify.”

By using model-assisted labeling, Sharper Shape was able to speed up the labeling process, cut their labeling costs in half, and train their models faster. This is just one part of their journey to make energy grid inspections safer, faster, and more affordable.

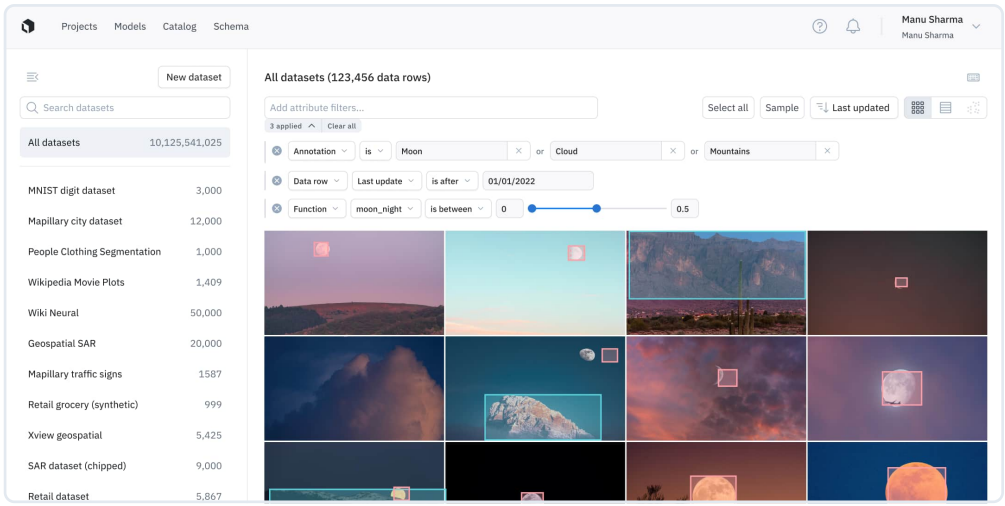
07 Utilize active learning and prioritize the right data

Your team can achieve considerable gains in productivity and efficiency by employing active learning and prioritizing the right data. We've seen customers reduce their iteration cycles by up to 8X and use less training data by curating their datasets. Data selection for active learning typically involves selecting rare data or difficult cases. In the case of rare data, models need a minimum number of examples in order to perform well on a chosen task. Without a sufficient number of examples to feed the model, it can be too inefficient to sample data randomly. Using tools like a similarity function with model embeddings can help you to discover more matching data examples. Teams can visually find patterns and identify edge cases in data through the use of model embeddings. By clustering visually similar data, teams can better understand trends in model performance as well as data distribution. While teams can calculate and plot clusters manually, some use cases in ML require more time-sensitive trend detection and a quicker approach. This is where a visual embeddings tool can be helpful to boost model performance and support an active learning workflow.



Model embeddings, shown here in Labelbox, can help visualize trends in data and cluster similar data, helping you uncover outliers faster for an active learning workflow.

Some data samples are more challenging for models due to obstructions, light, and even data source. For example, rain might obstruct the subject of an image. Labeling more of these cases and improving label quality can help the model learn the patterns required to perform better in those situations. You can curate the data that will have the most meaningful impact on model performance. To do so, teams must be able to search their data, labeled and unlabeled, quickly and easily with a catalog search function. Frequently, ML engineers need to write one-off queries to find data, but this inefficient process can slow down your MLOps. Now, teams are adopting flexible and scalable code-free search tools. With a tool like a searchable catalog, you can search for labeled and unlabeled data using filters for metadata, model inferences, and other attributes. You can then send this batch of data directly to a labeling project, or use model-assisted labeling and accelerate your active learning workflow.



The catalog search function shown here in Labelbox can help you search and discover data faster.

Active learning gives teams the ability to reduce annotation time, reduce backlogs, and achieve ML efforts faster and more accurately. With a training data platform equipped for active learning, your team can employ methods that help you reach a more reliable model and an easier, more standardized approach to addressing outliers, class imbalances, data drifts, and other edge cases.

08 Additional resources

To optimize your ML workflow and incorporate some of the processes mentioned above, take a look at the following guides, webcasts, and product documentation.

How to improve performance through active learning: Watch this on-demand webcast to learn how you can use Labelbox to diagnose your model's errors and curate your next labeled dataset.

The Labelbox guide to labeling automation: Download this comprehensive guide to learn more about how and why model-assisted labeling is the labeling automation strategy proven to reduce time and effort, and in real-world use cases.

Getting started with Python SDK: Take a look at our docs for a quick start guide to utilizing the Python SDK for better access to data.

How to optimize your labeling operations: View this on-demand webcast to learn how Labelbox's world-class workforce and dedicated labeling expertise can help you achieve seamless collaboration across teams, improve both time and cost savings, as well as boost productivity and efficiency.

Request a free Labelbox demo: As you now know, production AI teams spend a significant amount of resources and time building and maintaining training data infrastructure. Rather than building expensive and incomplete home-grown tools to create and manage training data, Labelbox is a complete training data platform that acts as a central hub for your team. Learn more about how Labelbox can save you precious time and accelerate your ML team's path to production AI by requesting a free demo.

Labelbox

Learn more about our offerings, sign up for a demo, or start using our free version today at www.labelbox.com.