AYADATA

ULTIMATE GUIDE TO CREATING TRAINING DATA FOR MACHINE LEARNING

AYADATA

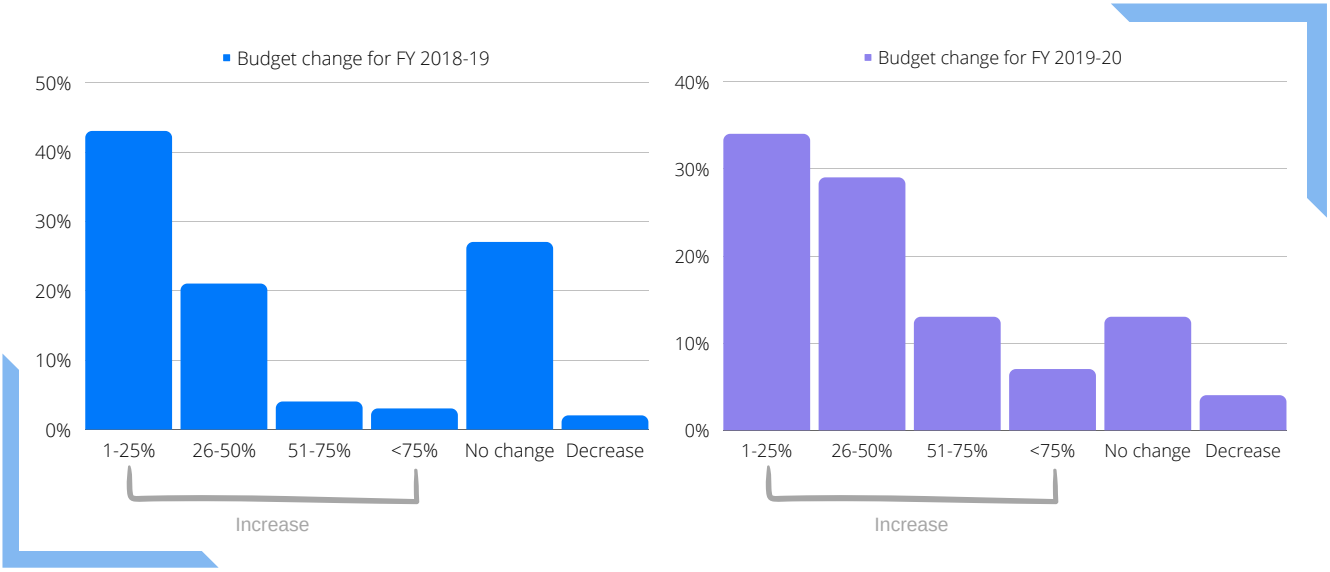
ULTIMATE GUIDE TO CREATING TRAINING DATA FOR MACHINE LEARNING

INTRODUCTION

Machine learning (ML) is a branch of artificial intelligence (AI) and is one of the most transformative technologies of our age. Coined as a term in 1952 by Arthur Samuel, inventor of an algorithm that played Checkers, **machine learning lives today inside everything from consumer devices and smartphone apps to satellites, unmanned drones, and self-driving vehicles.**

76% of surveyed businesses prioritized AI and ML in their budgets in 2021 and the machine learning market is projected to grow from $15.50 billion in 2021 to $152.24 billion in 2028, representing a CAGR of 38.6% in the 2021-2028 period. As businesses and organizations worldwide scale up their investment in AI, the tools and technologies required to support ML projects are also proliferating.

83% of organizations have increased AI/ML budgets year-on-year



Above: Investment in AI

AI and ML have been around in some form since the late 1950s, but it wasn't until the inception of neural networks and the development of complex machine learning algorithms in the 1990s that machine learning began to show its true potential.

Most machine learning algorithms fall into one of two categories: supervised and unsupervised. Supervised or semi-supervised models require some form of annotated data to learn from before being exposed to real-world data.

The methods employed to train machine learning models are similar to training our own brain - by exposing ourselves to structured, worked examples in a controlled environment, we can better interpret a process or task when we expose ourselves to it in real life.

This guide will explore how to create training data for machine learning models.

What's In This Guide?

This guide contains everything you need to know about creating training data for machine learning projects. Advice is given on both image and video labeling, audio labeling, and text labeling for computer vision, conversational AI, and NLP projects. Labeling techniques are discussed alongside examples and best practice guidance.

How Machine Learning Models Learn

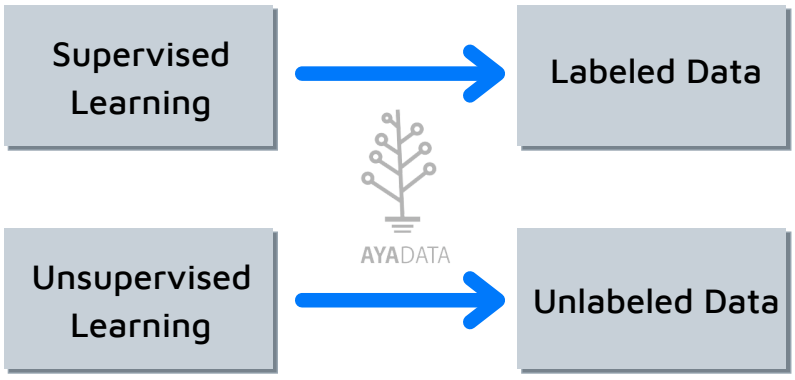
Machine learning is a concept born in the human brain - it uses computing power to solve problems and create knowledge in a similar way to humans.

While computers have always been excellent at working with structured data stored in databases and tables, machine learning takes this a step further to learn from complex, unstructured, and qualitative data, forming patterns and generating results, predictions, and decisions as it learns.

To achieve this, there are two main paradigms within machine learning:

Supervised And Unsupervised Machine Learning

Supervised and unsupervised machine learning are not the same, but they can be combined. The dividing lines between them are becoming increasingly blurry with the development of new-gen hybrid ML methods and technologies.



Supervised Machine Learning

Defined by the use of labeled and annotated datasets, supervised machine learning algorithms are trained using purpose-made datasets. By learning from labeled and annotated data, supervised models can accurately analyze and predict outcomes and make decisions when exposed to real-world data.

The purpose of a supervised algorithm is to map the function of multiple inputs to outputs with such a degree of accuracy that the model behaves as expected when subject to real-world data.

Supervised machine learning models are mainly divided into regression and classification models. Classification is focused on predicting labels, whereas regression is broadly focused on predicting quantities. Classification models are frequently used in computer vision when algorithms are trained to predict the class of real-world video and images based on what they learn from labeled inputs.

Regression and classification are further grouped into:

- Linear regression, for regression problems.
- Random forest for classification and regression problems.
- Support vector machines, for classification problems.

Unsupervised Machine Learning

In contrast, unsupervised machine learning utilizes machine learning algorithms to analyze, cluster, and categorize unlabelled data without any kind of prior training or exposure to that data. You have input variables, but no corresponding output values. The main aim of unsupervised machine learning is to explore data for correlations, patterns, and links.

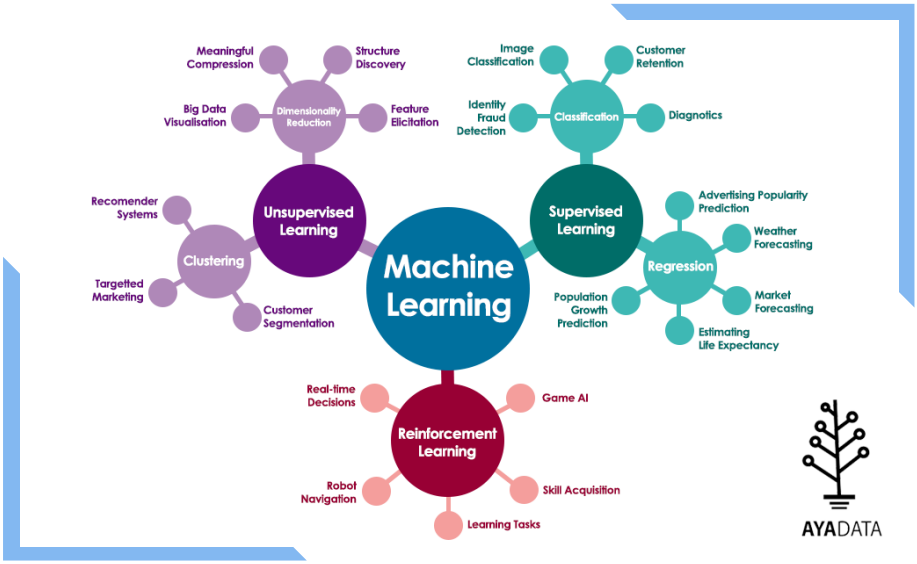
Unsupervised algorithms can work with vast volumes of data, such as customer analytics data, working to form links between different customers, their demographics, purchase history, etc.

Here, there are two foundational processes:

- Clustering, which involves grouping data points based on similarities.
- Association, which helps define and describe rules, such as when X customer also tends to buy Y product.

In addition to supervised and unsupervised learning, machine learning models can also be built using semi-supervised learning and other hybrid techniques. Semi-supervised models are fed small quantities of labeled input data to help guide classification and feature extraction from unstructured or raw datasets.

Reinforcement learning can also intersect with both supervised and unsupervised machine learning techniques and uses algorithms to discern the best course of action when exposed to a sequence. Reinforcement learning occupies its own category separate from supervised and unsupervised learning.



Above: Machine Learning

The Role of Training Data

Training data applies only to supervised machine learning projects.

- In the case of supervised machine learning algorithms, training data needs to be used to optimize the model.

- In the case of unsupervised machine learning algorithms, only raw data needs to be supplied to the model.

Here, we're focusing on creating training data for supervised machine learning projects.

Training data comes in a wide range of types and formats, such as text (words and numbers), images, video, and audio.

- A supervised natural language learning (NLP) algorithm will likely work with strings of annotated text.

- A supervised computer vision algorithm (CV) will work with annotated images and video.

- A voice assistant or conversational AI (like Siri or Alexa) will work with annotated audio clips.

Above: Image Labeling

To supply data to supervised machine learning projects, it needs to be labeled or annotated as per the project's requirements. The terms 'data annotation' and 'data labeling' are used interchangeably but the techniques used to label data vary depending on the data in question.

By explicitly labeling data with the features you want your algorithm to recognize, you're training it to react accordingly when exposed to real data that bears some or all of the same characteristics as your training data. Annotated datasets instruct the algorithm on how to behave when it's exposed to unannotated real-world data.

Once the model is trained on annotated data, it can be validated using a test sample, to measure the performance. The model has importantly not been exposed to the test sample during training and the data is typically randomly selected to be representative of real world data. If the performance on the test sample is satisfactory for the use-case, we gain reassurance the model is generalisable.

Labeling Project Planning

Before embarking on a labeling project, it's crucial to assess the project's requirements. The type of media required will have been selected, e.g., images or video for a CV project, text for an NLP project, etc. It's then necessary to determine what labels to use and why.

Labels will likely be chosen in line with the complexity of the model and the media required.

For example, a simple image classification model may only require class labeling, whereas an object detection model will require some combination of bounding boxes, polygon labels, semantic segmentation, polylines, etc.

Aya Data labeled datasets for a maize disease classification model. Here's a worked example of how we chose what labels to use for the project and how we applied them to our media.

1: The Problem and Considerations

Maize disease is a persistent problem that heavily impacts year-on-yield maize yields in Africa. While many maize diseases look similar, some tend to leave characteristic lesions on the leaves of maize plants. Nevertheless, it's often difficult for farmers to identify different types of diseases.

Early intervention in crop disease is key as it allows farmers to localize outbreaks and prevent spread. Supported by an initiative from Demeter Ghana, Aya Data went about labeling various diseased maize leaves to train a bespoke computer vision model that could identify diseases and suggest treatment options. Farmers would access the model via a portable app that helped them diagnose maize diseases.

A major consideration here was access to necessary domain specialists in maize disease. Aya Data worked with local agronomists, combining our labeling expertise with their farming expertise. This example demonstrates that domain knowledge is often critical in labeling projects, as is labeling real data from genuinely representative media.

2: The Media

Here, Aya Data would be labeling digital photographs of real maize disease. These photographs would be assessed with the assistance of local agronomists, who helped identify the specific diseases.

Aya Data thought it critical that training data should represent the real data that farmers would feed into the model. In other words, we needed real, photographed photos of maize leaves from the local area. The maize leaves should be presented to the camera in much the same way as a farmer would present a leaf to the model, e.g., by simply holding it up in focus against a typical background.

Aya Data ensured that the source media was high-quality, in-focus, and unobstructed throughout the course of the leaf. So long as the leaf was shown in its near-entirety without major obstruction, we were able to label the leaf accurately. See labeled example below.

We labeled 5000 images of diseased and non-diseased maize plants. 5000 images were sufficient to train an accurate model, but since Aya Data labeled data in-situ, we would have been able to label more if required. This is an important consideration - it's crucial to collect sufficient data from the site while you can access it.

3: Labeling

Since we labeled just one object (the leaf) with a class (the relevant disease), this task was ideally suited to semantic segmentation labeling. Semantic segmentation involves the pixel-by-pixel labeling of objects semantically related to a given class. Groups of pixels are assigned a class (in this case, the specific disease, if present, or no disease if not).

We can train a computer vision model to identify diseases on new, unseen leaves by labeling diseased leaves and giving diseases a class label. Semantic segmentation allows us to assign that class to all relevant pixels that identify a leaf in each image, as you can see below.

4: The Result

Our maize disease dataset enabled our partners to train a computer vision model that could detect common maize diseases in the area with 95% accuracy. Farmers were able to present leaves to the model via an app, which would return relevant results on the disease, severity, and treatment options. The training data was precisely representative of the real data exposed to the finished model, which turned out to be a major asset to the success of the project.

In summary, there were numerous problems to solve here, even despite the labeling task itself being relatively clear-cut as a semantic segmentation task. By working with local experts, our labeling team was able to accurately label disease classes, which was fundamental to the project. Moreover, by ensuring training data represented the media and format of real data exposed to the model, the resulting model was both accurate and user-friendly.

The Three Main Types of Data Labeling: Text, Image, and Audio

The three main domains of supervised machine learning revolve around text (including numbers), image, and audio.

1: Labeling in Computer Vision

Supervised computer vision applications require annotated visual data in either or both video and image formats. Both 2D and 3D images can be annotated:

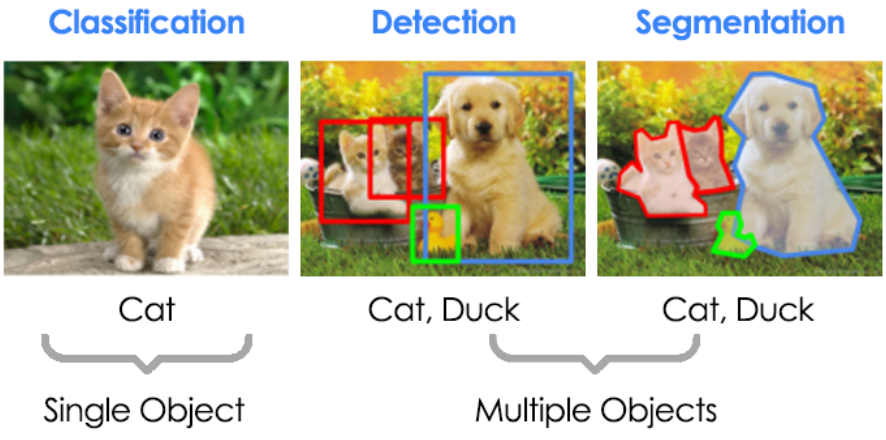
- 2D images and video, such as standard camera or video footage obtained from anything from a telescope to a microscope.
- 3D images and video, including light detection and ranging (LIDAR) and images from electron, ion, or scanning probe microscopes.

There are four main ways to label data for computer vision projects, but annotation methods and processes are flexible to the project's requirements and vary depending on the visual data and use case.

Above: LIDAR

1: Image Classification

By classifying instances of objects across a dataset, image classification is used to train an algorithm to recognize the class of an unlabelled image when exposed to real data.



Above: Image Classification

There are two subtypes of image classification:

- Binary class classification (two tags per dataset). Used for simple classification, e.g. 'car crash or no car crash.'
- Multiclass classification (multiple tags per dataset). Used for mapping one-to-one and one-to-many relationships.

Image classification identifies the class of an object contained within an image - if you train an algorithm on labeled single images of dogs or cats, it should predict non-labeled images as being either dogs or cats (in binary classification).

2: Object Recognition and Detection

Object recognition goes a step further and looks at objects within the image rather than the entirety of the image. By labeling objects within training images, you can teach an algorithm to locate and classify those objects.

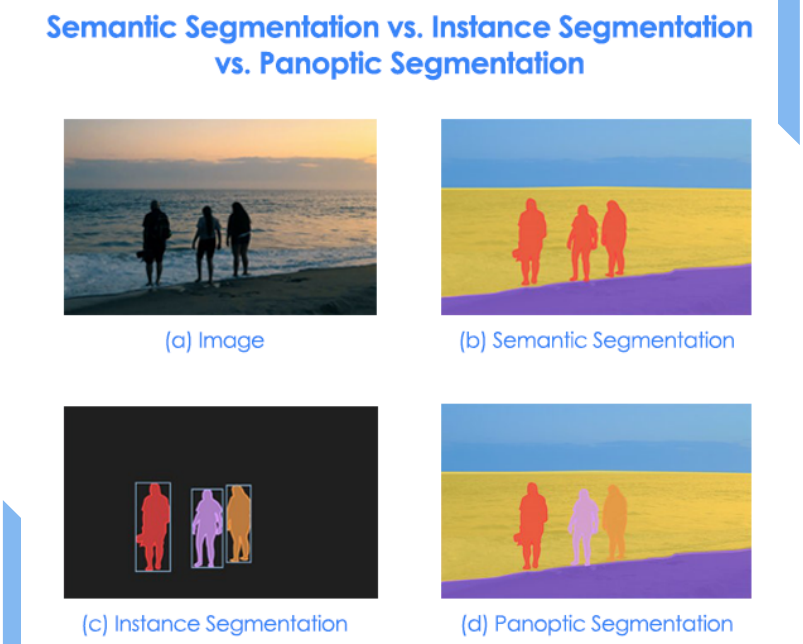
Standard object recognition tasks typically require a combination of bounding box and polygon labeling.

3: Segmentation

Delving further into image labeling, segmentation delineates the boundaries between image and video features to a higher degree of accuracy than bounding box or polygon annotation permits.

Segmentation is broken down into three areas:

- Semantic segmentation: the prediction of all objects semantically related to a given class.
- Instance segmentation: identifying what pixels belong to what object on an instance-by-instance basis, rather than a class-by-class basis.
- Panoptic segmentation: a combination of the above; classifying pixels for each instance and also predicting what class they belong to.



Above: Image Labelling

4: Boundary Recognition

Boundary recognition is explicitly designed to teach algorithms about the boundaries between different objects. Key examples include road markings, sidewalks, powerlines, etc.

Instead of focusing on individual objects, boundary recognition focuses on separating backgrounds from foregrounds, or focal features from peripheral features.

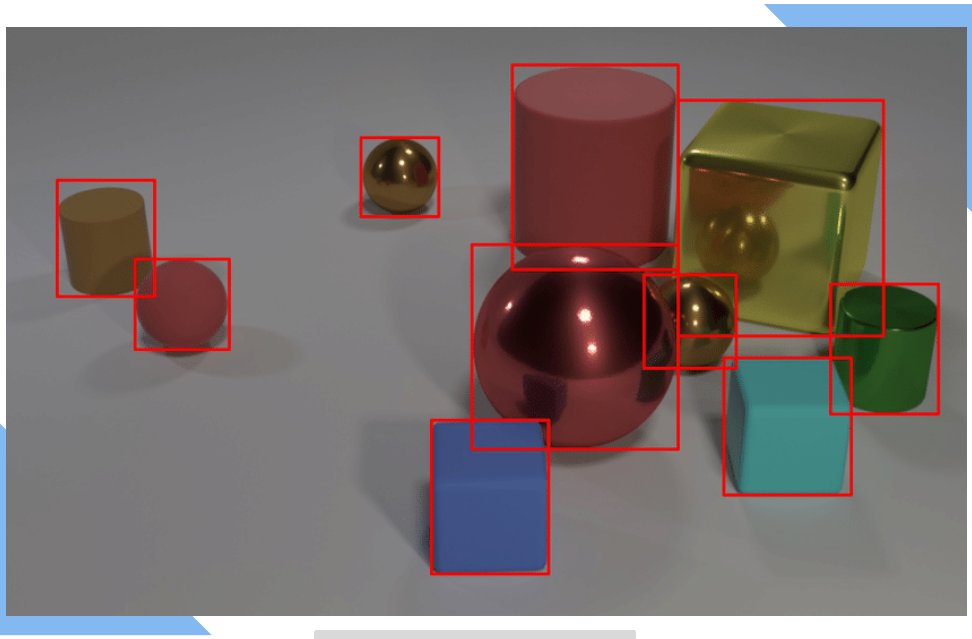
In medical imaging, boundary recognition can help identify ambiguous structures and classify features such as density or opaqueness.

How do you Annotate Videos and Images?

Videos and images are typically labeled using a combination of:

1: Bounding Boxes and 3D Cuboids

The go-to labeling method for clear, high-definition images or videos with clearly defined objects. Different classes of bounding boxes can be applied to data to label objects and their respective classes quickly.



Above: Bounding Boxes

3D cuboids are the same as bounding boxes but are 3D, enabling annotators to map the depth of objects.

2: Polygon Annotation

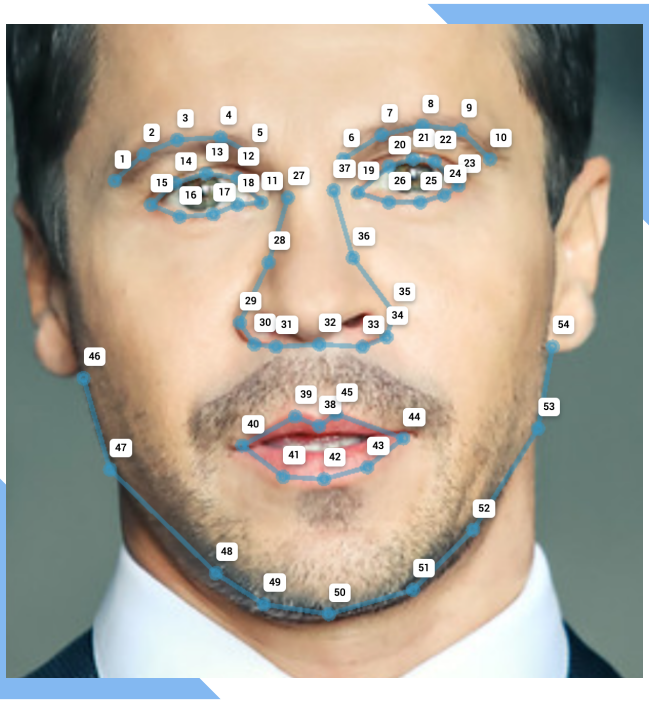
Similar to bounding boxes, polygon annotation uses multiple vertices to map complex shapes. Complex polygons can be labeled with thousands of vertices.

3: Masking/Segmentation Mask

Pixel-level labeling is used for the purposes of segmentation and boundary recognition. Masking can help label only the parts of the image that are relevant to the model (e.g., the sky and not the ground).

4: Keypoint/Landmarking

Similar to polygon annotation, landmarking involves using a small collection of connected polygons to annotate multiple objects of similar shapes. Human body parts are one example - keypoint labeling is often used to map faces. Key points can highlight segments or components without drawing any sort of box or shape.



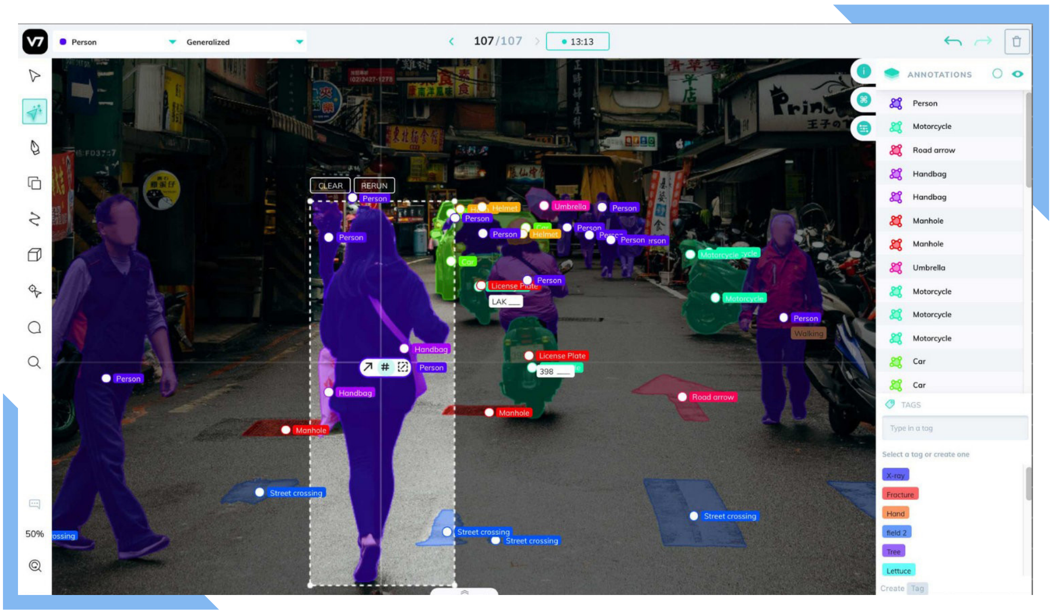
Above: Keypoint Labeling

5: Lines and Polylines

Lines are essential when working with featureless images, like roads. They're used to map line-like objects, such as pavements or road markings, and are employed heavily in AV training data.

6: Tracking

Tracking is used to track an object's movement in video data. For example, interpolation tools in data annotation platforms allow annotators to annotate a still image, then skip forward to a new frame and move the annotation - the annotation will then automatically track the movement from the first frame to the last frame.



Above: Complex Image and Video Annotation

Image and Video Labeling Best Practice

- **Ensuring bounding boxes are closely fitted around the object** (ideally with pixel-perfect precision). The same applies to polygon annotation. When masking or drawing boundaries, it's also necessary to highlight the exact boundary with pixel-perfect precision.

- **Including occluded or obscured images**, and making sure the hidden part is contained in the label.

- **Tagging all instances.** It's easy to miss some instances in complex images, but each instance of a relevant object needs to be labeled.

- **Being specific with labels.** Oftentimes, it's better to be over specific. For example, if there are two potential breeds of the same dogs (e.g. a short-haired chihuahua and a long-haired chihuahua), then it's better to label them differently, even if the machine learning model only needs to predict a chihuahua. If there are issues with the model, it will be easier to edit one of two sets of labels, or merge them into one.

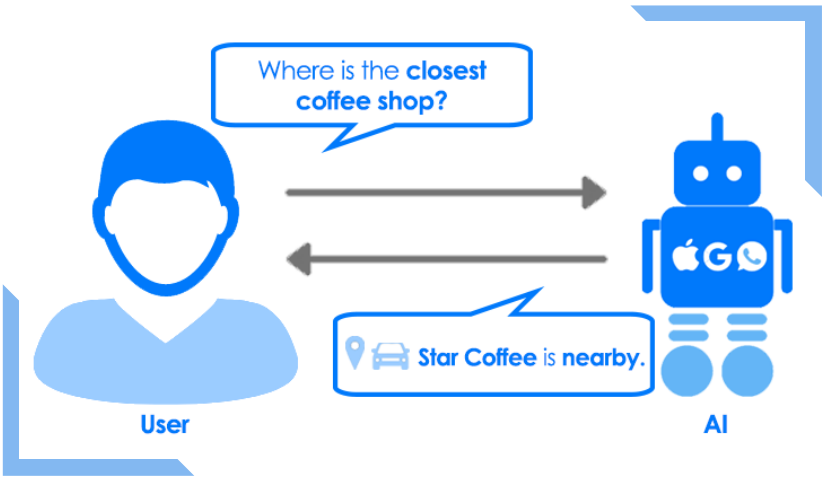
- **Maintaining consistency across the entire dataset.** This is especially important when multiple teams work on the same dataset over multiple days, or when modifications are required after training.

For further examples of these techniques, check out Aya Data's Use Cases.

2: Labeling for Natural Language Processing (NLP)

"Language is our Rubicon, and no brute will dare to cross it," said linguist Max Müller in 1862. Today, chatbots and conversational AIs live all around us.

AI chatbots and a huge range of other language-oriented models invoke the power of NLP and are exceptionally diverse.



Above: Chatbot Example

Two core concepts in NLP are syntactic analysis and semantic analysis:

Syntactic Analysis

Syntactic analysis assesses the structure, formation, and logic of a piece of text, and involves:

- **Parsing**: checking the text for spelling and grammatical features.
- **Sentence deconstruction and word segmentation**: Breaking up larger pieces of text and segmenting sentences or even words into smaller fragments.
- **Grouping**: Grouping words based on certain characteristics.
- **Stemming**: Converting complex words into root forms.

Most forms of syntactic analysis occur in the pre-processing stage and help prepare the corpus for semantic annotation. It's worth mentioning that, unless you're building your model from scratch, many NLP AI frameworks and services provide libraries to automate syntactic analysis processes. Syntax is broadly continuous across a language - it's the semantic meaning of text that changes.

Semantic Analysis

Syntactic analysis extracts structure, but semantic analysis extracts meaning, and involves:

- **Context**: using the context and other assistive data to help derive meaning (e.g., location, weather, recent purchases, etc.)
- **Entity Extraction, Named Entity Recognition and Sentiment Analysis**: Extracting proper noun entities (e.g. locations and names), other entities (e.g. kettle, car, or dog), and emotions (e.g. anger, happiness, neutrality), then sorting those into categories.
- **Relationship extraction**: Discovering links between different words, e.g., a subject, object, and action (e.g. the man closes the shop). Recognizing when one named entity links to another, despite not being referred to as the same thing (e.g., when a product is initially named, then referred to thereafter named simply as 'it').
- **POS Tagging**: Identifying the individual components of text (noun, tense, adverb, adjective, etc.

How Do You Annotate Text?

Language is complex and varies in script, grammar, syntax, and semantic structure.

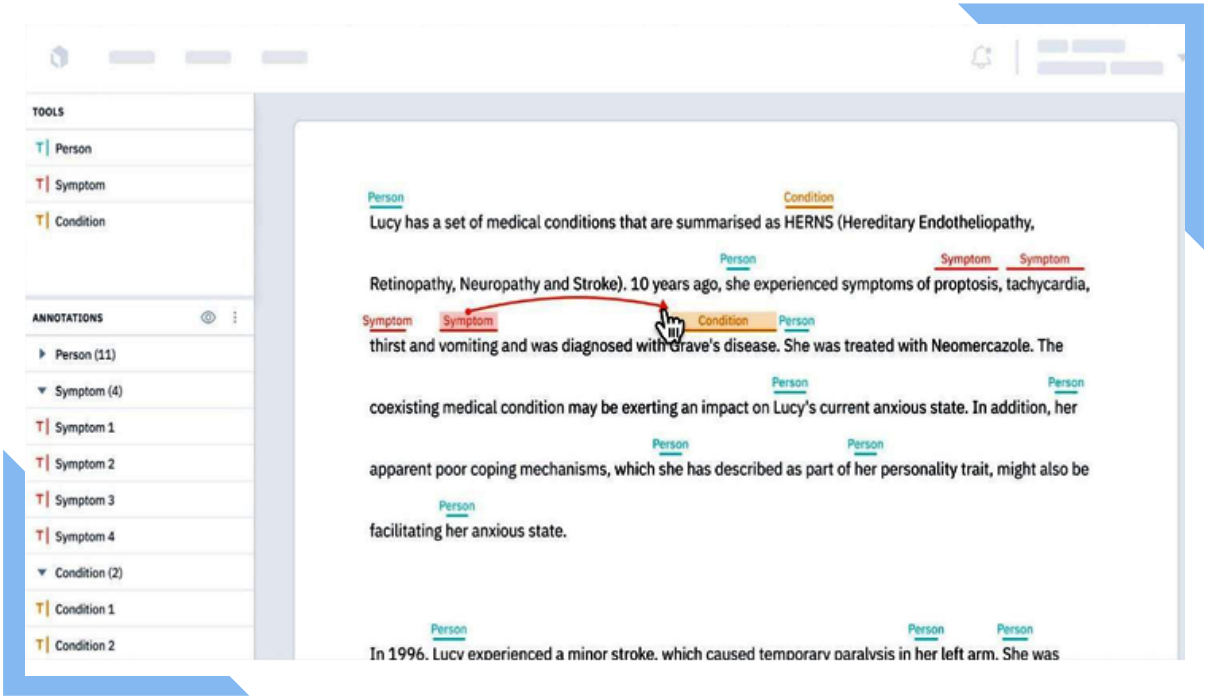
Text annotation is used to teach NLP algorithms about expected text inputs and involves adding both labels and metadata to pieces of pre-processed text.

1: Assigning Labels

Annotators can apply labels to the entire sentence or phrase. For example, if a phrase is an example of an idiom, "every cloud has a silver lining," then it can be tagged as so. Labels also allow different phrases and sentences to be pre-sorted before sending through to other models, or human teams.

You may not need to specify many label types. For example, if you're training an NLP model to extract disease/illness features from a doctor's summary, you may only need to label the subject (e.g. the person, a named entity), the symptoms, and the disease/condition (another named entity).

Below is an example of NLP labeling for medical diagnostics (in LabelBox).



Above: NLP Labeling

In the above example, you can see how the person, condition and symptoms are labeled in the text.

2: Metadata

Metadata is added to specific words or phrases to provide additional data beyond a singular label. So, if the text says "I want to order pasta tonight," then metadata can be applied to the word 'pasta', to mark it up as a food item, potentially also with a product ID, whereas 'tonight' can be marked up with a specific range of times, e.g., 5 to 11 pm.

3: Sentiment Annotation

Sentiment analysis seeks to uncover the text's tonality and emotional qualities, i.e., the sentiments. Marking up pieces of text that help determine the emotional state of the text allows algorithms to predict sentiments in real data.

For example, in the phrase "the product is average and customer service reasonable," the words 'average' and 'reasonable' are both indicative of a neutral emotional state.

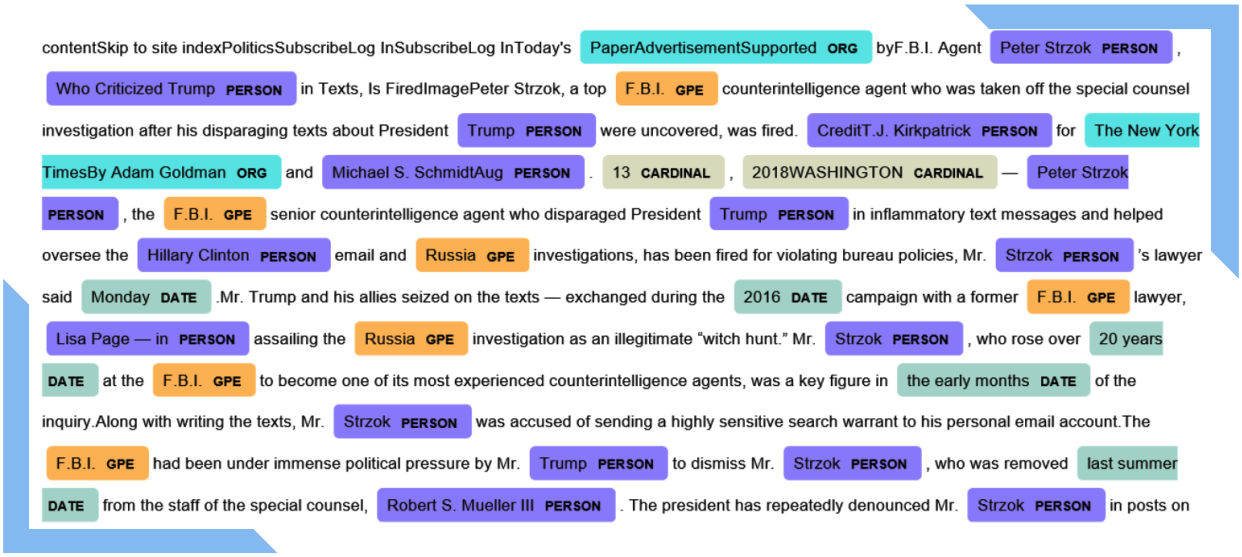
On the other hand, a phrase like "really good experience! I loved it!" reveals happy emotions with modifiers that help rank that sentiment as much better than average, e.g. 'really good', 'loved it' and also the exclamation mark.

4: Intent Annotation

Intents are a key element of chatbot data and are used to express the intention of the query. For example, in the phrase "I want to speak to an advisor," the intent is obviously connecting to an advisor. However, "directions to Kyoto," is also an intent - the user wants to receive directions.

5: Named Entity Annotation and Relationship Linking

Named entities include practically all real-world objects, including brand names, company names, products, people, places, etc. A passage might only mention a named entity once, e.g., "BA Tobacco," with subsequent mentions simply referring to it as "it" or "the business," etc. NLP algorithms must be trained to delineate named entities from each other, even when they are no longer named in the text. This is called entity linking or entity disambiguation.



Above: Entity Annotation

6: Keyphrases

It's also often necessary to label entire keyphrases, known as utterances. This is particularly useful for chatbots, especially when chat queries have already been pre-sorted, meaning users are likely to ask similar questions.

7: Part-of-Speech or POS Tagging

Involves annotating parts of speech, including nouns, verbs, prepositions, adjectives, and a whole host of other linguistic tags. This comprehensive approach teaches complex algorithms how to understand similar groups of words that have different meanings depending on the POS elements involved and their order.

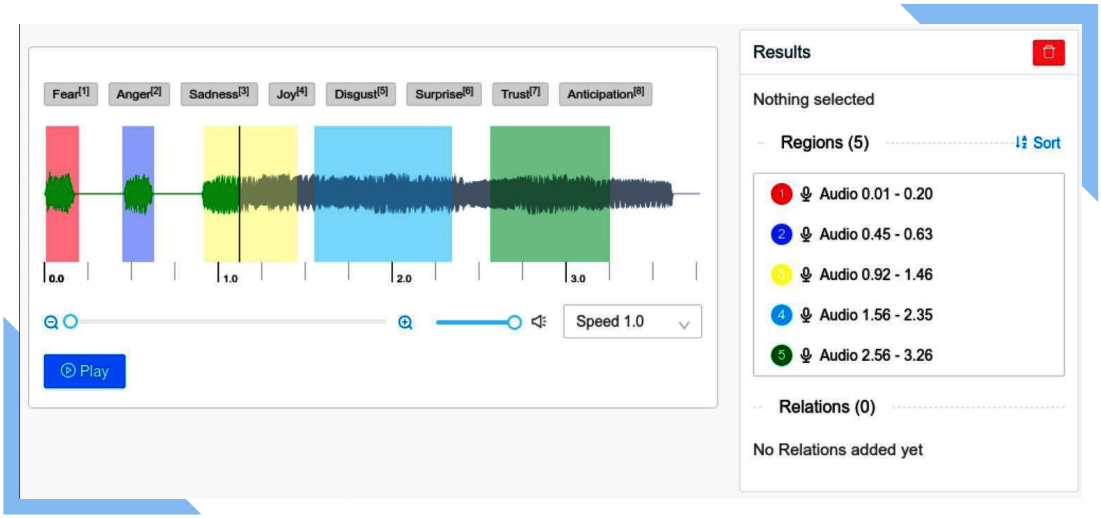
Text Labeling Best Practice

- **It's crucial to work with relevant statements and text.** For example, training a model with US customer service inquiries when it's to be deployed in the UK or Australia may result in subtle inaccuracies. Make sure that training data is representative.

- **Using perfect training data can result in poor functioning models.** When real people speak in short-hand, use slang, colloquialisms, and poor spelling or grammar, perfect training data is likely non-representative. Instead, try to incorporate such variations into the training data and label appropriately. Reducing multiple terms to their root meaning is called normalization.

3: Labeling For Audio

Audio labeling is broadly analogous to text labeling. Labeling for audio often involves speech recognition, converting words to text, and applying NLP annotations combined with annotations specific to speech, e.g., intonation to indicate different emotions.



Above: Audio Labeling

It's also worth mentioning that while many of these examples are focused on human speech, it's possible to annotate a whole host of non-human sounds including background noise, noise emanating from objects within the sample (e.g. a door slam, or keys jangling), and music.

Many subtitles are generated using audio-to-text algorithms that parse soundwaves into text, providing both text that corresponds to speech, e.g. "Good evening, Mr. Bloggs," as well as other noises, e.g. "*jazz music plays in the background* or *door slams shut*.

The four main audio annotation techniques are:

1: Speaker Identification

Speaker identification involves applying labels to regions to identify who is speaking, including any background noise (such as car noises or blustering trees), as well as silence.

2: Audio Classification

Classifying clips based on their purpose, intent, dialect, and other semantic information.

3: Audio Emotion Annotation

Annotating emotional qualities indicated by pitch intonation, speech rate, voice intensity, and other speech artifacts and articulations.

4: Noise Annotation

While it's possible to convert audio speech into text, this isn't possible with background noise and other environmental sounds which must be annotated from the audio clip. To accommodate this, many audio tagging workflows integrate speech-to-text annotation with soundwave annotation, allowing annotators to label both the text and the audio file.

The Issue of Bias and Representation

While you can refine data, using test results to re-annotate, enrich, or otherwise improve your training datasets, starting with a good-quality, well-annotated and representative dataset is essential.

Training data must also accurately represent the use case and purpose of the algorithms, accommodating all known possible inputs. Introducing bias into machine learning algorithms via poorly representative data is a pervasive issue, as proved when Amazon scrapped their AI recruiting tool - it was trained on years of data where men over featured in tech roles and proved prejudiced against women.

There is still a lack of diversity in datasets which has an unquestionable social impact, also leading to the downfall and eventual redundancy of the models themselves.

To fight bias, it's crucial to maintain a tight circle of labelers who understand the role of potential bias in data labeling. Flagging issues during the annotation phase enables more data to be added to the training set, or for annotations to be modified, added, or removed.

How Can I Get Training Data?

Anyone can label simple data using a bounding box, but as you drill down into more nuanced forms of ML labeling, creating quality training data becomes increasingly difficult. For example, some 3D or LIDAR data labeling tasks require deep domain knowledge and expertise, and even when using simple data formats projects might have extensive labeling requirements to be strictly adhered to.

Firstly, training data can be either real or synthetic. Real data is exactly that - data taken from the real world in the form of a piece of text, audio, image, or video. Synthetic data, on the other hand, is artificially generated.

Real vs Synthetic Training Data

As today's machine learning applications grow in complexity, there is a greater need for effective, scalable training data. Synthetic data is on the rise - Gartner estimates that around 60% of data used for supervised and hybrid ML projects will be synthetically generated by 2024.

Synthetic data is generated using platforms such as Nvidia's Omniverse Replicator. This graphical processing platform allows users to generate vast quantities of synthetic data. For example, instead of collecting and labeling thousands of hours of footage from real streets to train AVs, you could simply generate the equivalent in synthetic data and train models on that instead.

Above: Synthetic Pixel Segmentation

Synthetic data has benefits and drawbacks:

Benefits of Synthetic Data

Scale: By generating huge datasets in a short space of time, synthetic data collection solves the issue of scale. A key issue in advanced ML projects (such as training unmanned vehicles) is obtaining sufficient high-quality data.

Privacy and Regulation: Synthetic data does not contain any personally identifiable information. This aids in privacy law compliance (e.g. GDPR).

Specific: Synthetic data can be created for situations where real data simply doesn't yet exist, e.g. training spacecraft for missions on the surface of Mars where probabilistic models of the environment must suffice.

Drawbacks of Synthetic Data

Narrow in Scope: Synthetic data can't be generated for anything and everything. It's good for creating large, systematic datasets, but fails when it comes to creating truly natural datasets with outliers and noise.

Bias and Misrepresentation: Since synthetic data is entirely controlled by the user, it places an additional onus on the user to ensure that the dataset is unbiased and representative. The data is not guaranteed to be a natural cross-section of the truth.

Trust: Synthetic training and testing environments can lead to false conclusions and unexpected outputs when exposed to real test sets.

See our guide for an in-depth comparison between real and synthetic data.

Is Real Data Better Than Synthetic Data?

There isn't a yes or no answer, but currently, real data is more appropriate for most supervised ML projects for a number of reasons. One of the most exciting aspects of machine learning is applying models in creative and novel applications. In these situations, synthetic data often doesn't provide the genuine, real-world, representative datasets required to train an innovative model.

For example, Aya Data provided annotated images of maize diseases to help our client build a disease classification application. In this situation, using synthetic data wouldn't have been appropriate. The project required real, genuine human-annotated images of maize disease. Labeling these images by hand enabled our team to apply their domain knowledge, with assistance from local agronomists.

Synthetic data becomes a consideration when datasets need to be scaled up beyond human capabilities. Moreover, while synthetic training data seems like a shortcut, it's often the opposite. If a model is trained with synthetic datasets, more testing has to be done to ensure it works properly with real training datasets.

Obtaining Training Data

Obtaining training data depends on the project requirements. In some cases it's possible to acquire completely free open-source data from services such as Google Dataset Search, Kaggle, Data.gov.uk, and Data.gov.

Data can be purchased pre-processed and pre-annotated, Otherwise, it will need to be annotated in-house, by crowdsourcing, or by a managed data labeling service like Aya Data.

Read our comprehensive post on obtaining machine learning data here.

Hiring Labeling Workforces

Some datasets are pre-processed and pre-annotated. Otherwise, the dataset will need to be annotated in-house, by crowdsourcing data labeling services, or by a managed data labeling service like Aya Data.

In-house teams provide the tightest control over labeling projects, which is an advantage in long-term, sensitive and business-critical projects. However, hiring an in-house labeling team is undoubtedly the most expensive and long-winded way to label datasets.

Crowdsourcing data labeling services involves services such as Amazon's mTurk. Crodsourcing is an advantage when a very large manually-labeled dataset is required with minimum quality control. Crowdsourced labeling projects also forgo access to domain experience.

Finally, managed labeling services provide access to high-level labeling skills, while also ensuring tight project security and control. Professional labelers work with the latest labeling software, which improves training dataset quality, and can incorporate domain experience into datasets when required. Managed services are becoming a catch-all for a wide range of HITL services.

Managed labeling services like Aya Data are an example of HITL workforces. Read our in-depth comparison of in-house vs. crowdsourced vs. managed data labeling here.

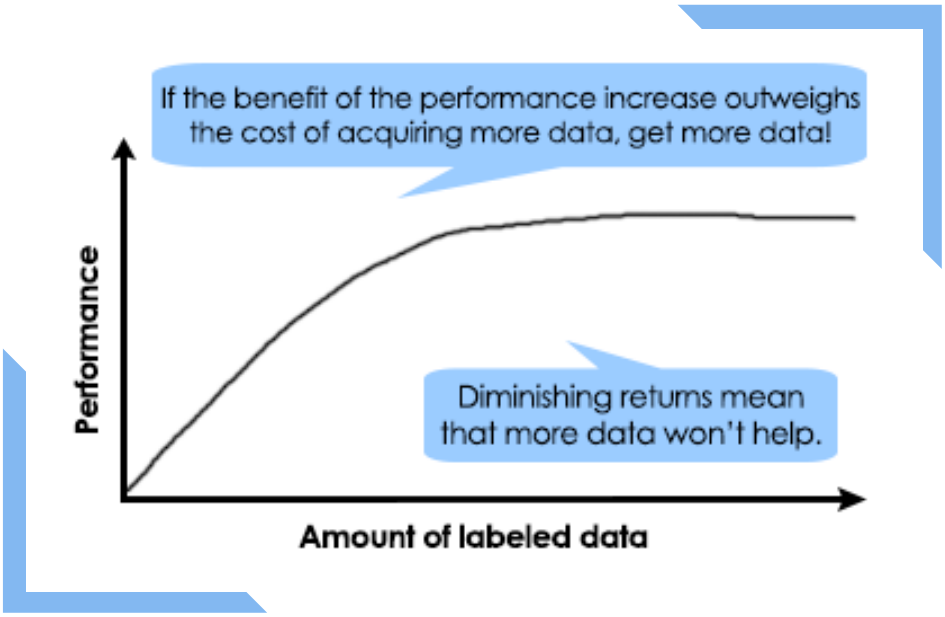
The quality of training data and annotations is directly linked to the accuracy of the model. Data annotation is a sub-field of AI in its own right, especially when dealing with intricate data such as complex and 3D images, LIDAR, complex video, and multi-speaker audio.

How Much Data Do You Need?

The volume of data required for machine learning projects varies massively. For example, a simple computer vision model that is only exposed to a few different objects will require much less data than a complex model with thousands, millions, billions, or even trillions of variations.

These variations are called degrees of freedom, which are logically independent values that have the freedom to vary in the data sample.

Any parameter or attribute that affects the model is a degree of freedom. Some machine learning engineers apply heuristic calculations based on the degrees of freedom available to the model to estimate how much data is needed. Most of the time, the sample size is chosen based on an ad-hoc evaluation of the project. The human in the loop can add more data if the training sample proves insufficient.



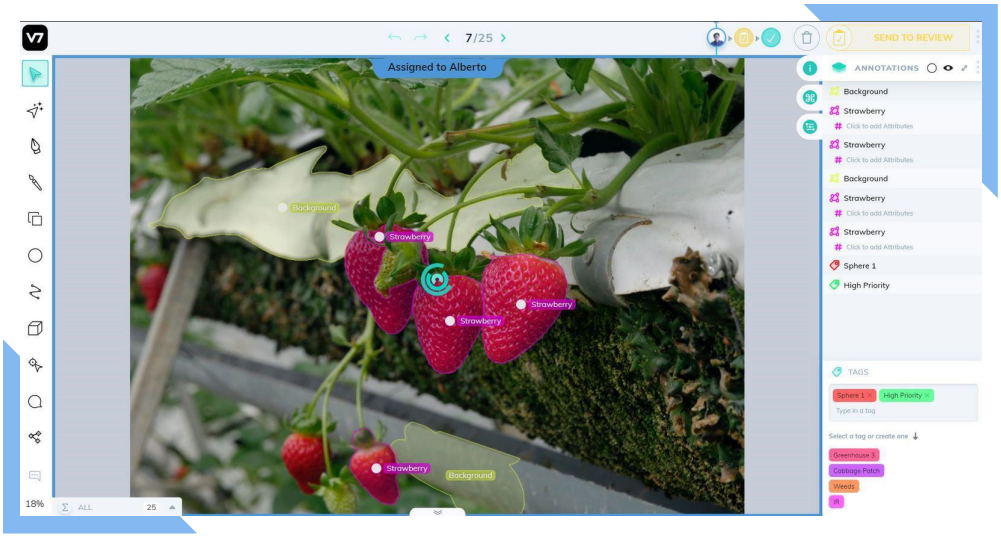
Above: How Much Data Do I Need?

It's also possible to plot the model's accuracy, add new data, and compare accuracy across iterations. Once the graph flattens out, adding more randomly selected has little impact on the model's accuracy. At this stage, a technique called active learning is useful in selecting training data to label which will most likely have a performance impact on the model.

Data Annotation Platforms

To assist in the data annotation process, there are many data annotation platforms that break down various annotation tasks into manageable bitesize chunks. Annotators can select from various tools, e.g. bounding boxes, polygons, and key points. Files are loaded sequentially into the interface, and annotations are applied.

Here is an example of V7 Lab's annotation interface:



Above: Labeling in V7

Whilst most high-quality annotation platforms are paid services, there are free and open-source options too.

Below is a compilation of some paid and open-source data annotation and labeling tools:

Paid Annotation and Training Platforms:

1. V7 Labs
2. Labelbox
3. DataLoop
4. Alegion
5. Playment
6. Prodigy
7. Scale AI
8. SuperAnnotate
9. Supervise.ly
10. Appen

Free Annotation and Training Platforms:

1. Audacity
2. CVAT
3. ImgLab
4. Labelimg
5. LabelMe
6. VoTT

Training Data, Validation, and Test Sets

When annotating data, it's also essential to create validation and test sets in addition to a training set. All supervised machine learning projects require at least three independent sets of data.

Training data: Training data consists of the majority of data required for a machine project. Around 60% of your total data should be used for training - this is just a rough estimate.

Validation data: Validation data is held back from the training data, and should be a true cross-sectional representation of the training set. Validation data is used to test the skill and accuracy of the model on a periodical basis.

Test data: Test data varies from validation data in that it's only used after the model has been trained, whereas validation is used during training. Test data is usually completely unlabelled to replicate real data.

Test data can be split into multiple sets. For example, testing an AV will require different sets for driving in different conditions:

- A test set for driving at sunset
- A test set for a snow
- A test set for driving in fog
- A test set for driving in storms

Gold Sets

Gold sets are perfectly labeled and scrutinized time and time over again for their precise accuracy. Moreover, gold sets help guide annotators and are used to weigh annotated sets against the gold set to check for issues.

It's also essential to create gold sets when distributing labeled data. For example, professional medical imagers might create a gold set to distribute to international healthcare services to add their own data and train their own models with guidance from the gold set.

Multi-Pass Blind Annotation

Rather than annotators working through images chronologically, multiple annotators can work on parallel tasks which are then checked against each other for consensus. Annotators can't see each other's work, and any discrepancies are adjudicated by an external supervisor.

Automated Data Labeling

The process of data labeling can be automated, at least partially. Automated data labeling models are machine learning models in their own right - their job is to predict what label is required and apply it with minimal human decision-making.

You can read our guide to automated data labeling here.

Model-Assisted Data Labeling

Model-assisted data labeling involves utilizing pre-trained models that extract features from the data and apply their own labels or training new auto-labeling models using a subset of the training data. For example, a data labeling platform can apply some bounding boxes or vertices to groups of images ready for human teams to add, remove or adjust if necessary. A more advanced example of automated data labeling is programmatic data labeling, which involves training auto-labelers with heuristic rules that guide the labeling process organically.

Are Automated Annotations Accurate?

Automated annotations can work well in the case of simple data with easily decipherable features.

Right now, automation expedites the annotation process without replacing human annotators.

Keeping humans in the loop is sometimes essential for detecting issues in the data, e.g., failure to properly represent the use case, or possible bias or missing features.

Building a culture of collaborative data labeling helps iron out issues in the training data, leading to more robust training sets. Automation cannot yet achieve the critical thinking of expert data labeling teams.

Summary: Ultimate Guide to Creating Training Data for Machine Learning

Training data is essential to any supervised machine learning project. Supervised algorithms need guidance - they need to be shown the target.

To train supervised machine learning models, you need to create accurate, representative datasets. The process of data annotation transfers the kind of logic and decision-making that humans take for granted to the model, teaching it to react and understand complex real-world data in a similar way to humans.

Both humans and machines are intrinsic to machine learning. After all, many machine learning concepts, processes, and technologies are modeled on the human brain. Data annotation bridges the gap between humans and machines - it allows us to train models based on skill and intuition.

Aya Data's mission is to provide leading-edge data annotation services. We have deep industry expertise across a huge range of projects that have proven real-world impact. Our team of experienced data labelers and annotators combines wide-ranging domain knowledge with class-leading data annotation tools and our own skills and intuition.

Do you need help with data labeling? Contact us today to discuss your labeling project