

# 6 Challenges in Operationalizing Computer Vision Models

AND HOW ARTHUR IS WORKING TO ADDRESS THEM





# 6 Challenges in Operationalizing Computer Vision Models

## AND HOW ARTHUR IS WORKING TO ADDRESS THEM

### Market Predictions

According to the Allied Market Research January 2022 report, the global Computer Vision industry generated \$9.45 billion in 2020, and is [anticipated to generate \\$41.11 billion by 2030, witnessing a CAGR of 16.0% from 2021 to 2030.](#)

Weekly, we read headlines showcasing innovative use cases for computer vision including facial recognition, search engine image retrieval, smart cars/autonomous vehicles, biometric systems and surveillance applications.



While the CV market is poised for exponential growth, why hasn't the industry accelerated faster?

### Here are 6 Challenges to Solve

- 1 Single static imagery is complex to understand; video/streaming media even more so
- 2 A combination of techniques inform CV decision making
- 3 AI processing power makes it costly to train, operationalize, and scale CV models
- 4 Incomplete datasets or lack of annotated datasets lead to poor quality CV data
- 5 CV models quickly degrade due to anomalies and data drift
- 6 Bias and unfairness in visual datasets introduce model risk

## Arthur: The Only Solution for CV Monitoring and Optimization

In June 2021, Arthur was the very first company to launch ML performance and data drift monitoring for computer vision models, including in-depth explainability. **Today, we are the only AI Performance solution that supports monitoring and improving CV models.**

Enterprise teams use the Arthur platform to:

- Perform object detection.
- Find anomalies in incoming images.
- Monitor CV models for drift and bias.
- Provide local explainability.

While this paper outlines current challenges in the CV space, the Arthur team is working closely with academia and industry to collaboratively solve the exciting challenges ahead. **Together, we'll continue to develop leading technology to measure and improve computer vision machine learning models for better results across accuracy, explainability, and fairness.**

VISIT [ARTHUR.AI/CV](https://arthur.ai/cv) FOR MORE INFORMATION AND A PRODUCT DEMO.

# 1 Single static imagery is complex to understand; video/streaming media even more so

## Challenge #1

The core objective of computer vision is to understand images.

Computer vision models must be robust to objects of interest that are often off-center, out of focus, poorly lit, or have a variety of scales. As such, semantic context (determined by objects, relationships, locations, and global composition) can aid general understanding of image context.

Given complexities inherent in computer vision making sense of image context, **Arthur is concentrated on model performance, monitoring, and validation for single static images for computer vision applications.**



## Inputs

## Output

## Arthur Capability

Single static image.	Score, Binary Classification or Multi-Classification.	Yes
Multiple images over time; Video computation	Score, Binary Classification or Multi-Classification.	Potential future roadmap development.

## 2 A combination of techniques inform CV decision making

### Challenge #2

Today, advanced industry use cases for real-world applications require a combination of CV techniques. © Arthur supports image classification, regression, and bounding box object detection.

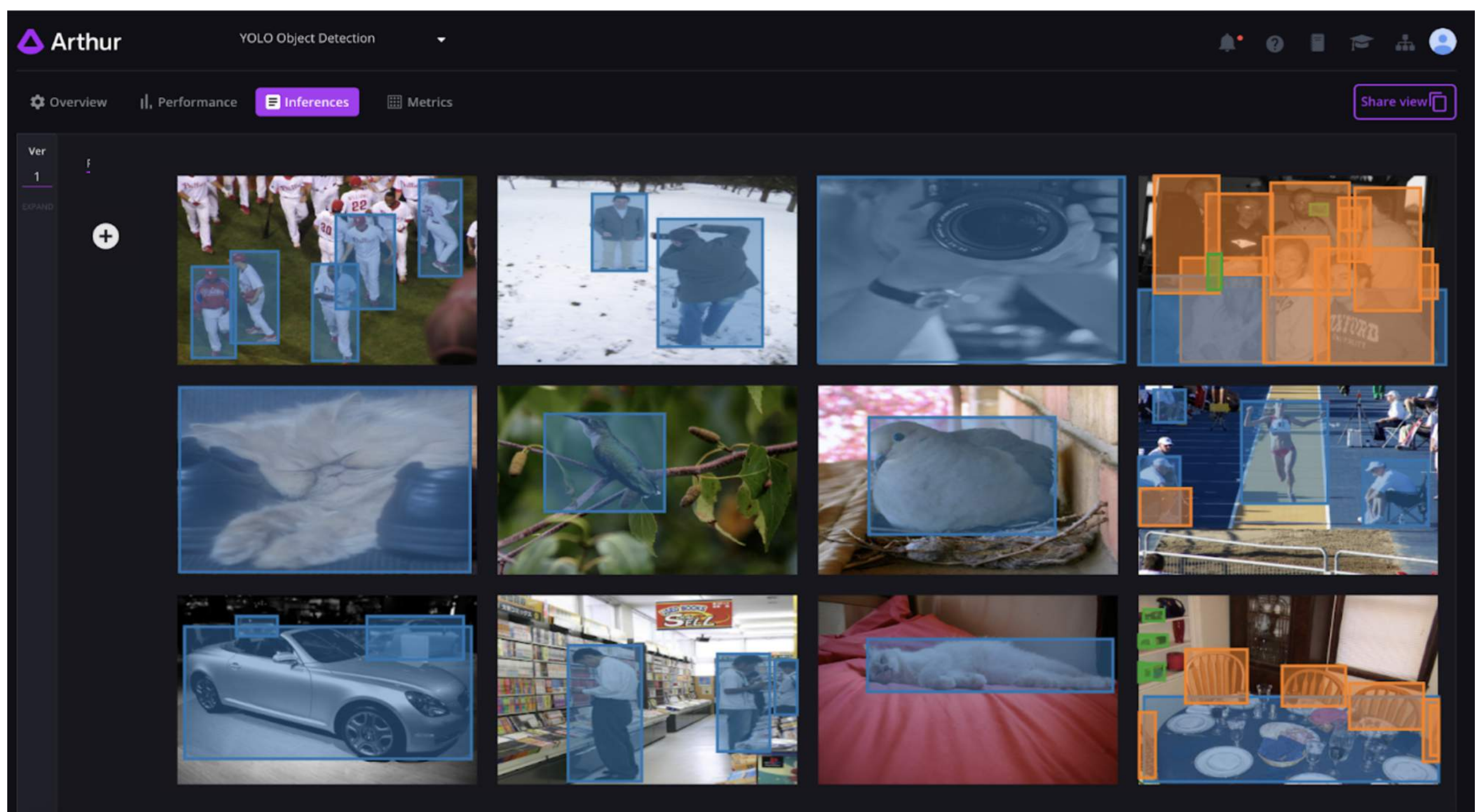


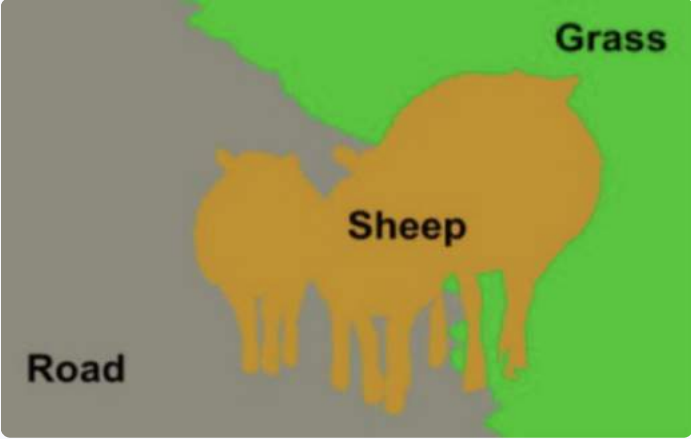
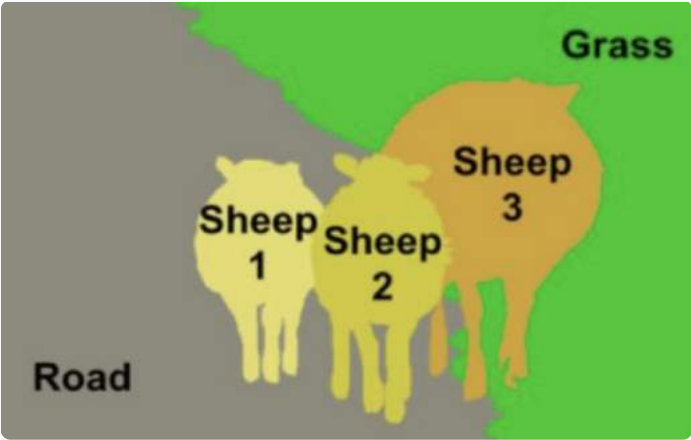



Image Classification	Yes   Assign a class label to an image.	
Image Segmentation	<div>Yes</div> <div>→ <b>Image segmentation</b> (boundary box object rectangle outline specifying X &amp; Y coordinates)</div> <div></div> <div>CAT</div>	<div>Potential Future Roadmap Development</div> <div>Semantic/ instance, panoptic/ edge detection segmentation (pixel classification level).</div> <div>→ <b>Image segmentation</b> (pixel object outlines)</div> <div></div>
Image Segmentation		<div>→ <b>Semantic segmentation</b></div> <div></div> <div>→ <b>Instance segmentation</b></div> <div></div> <div>→ <b>Edge detection segmentation</b></div> <div></div>



## Object Detection

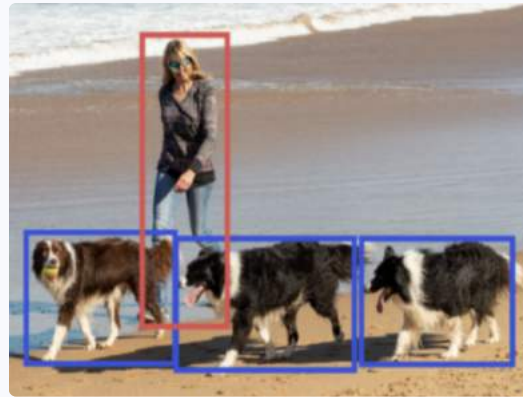
Yes

Bounding box is drawn around each object of interest in the image and assigned class labels.

Example:

● RED = WOMAN

● BLUE = DOG



## Pattern Detection

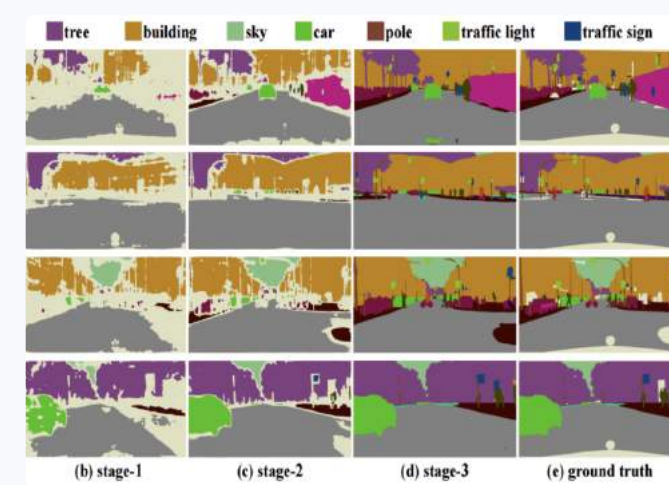
Yes

Only if image classification is utilized.



### Potential Future Roadmap Development

Pattern detection based on semantic/instance, panoptic/edge detection segmentation.



3

AI processing power makes it costly to train, operationalize, and scale CV models

### Challenge #3

**Very large datasets are required to train algorithms for computer vision applications.** It's often cost prohibitive to scale the computing power, cloud computing storage, and product hardware that are necessary to support deployment and scale objectives of ML projects involving very large CV datasets.

© **Arthur's platform has a highly scalable microservices ingestion architecture to monitor all your models in one place.**

It scales up and down to ingest up to 1+ million transactions per second and deliver insights quickly with Kafka (queuing), Go (programming language), and Clickhouse (database management). **Most other industry vendors do not use tech stacks that support CV model scalability.**



# 4

## Incomplete datasets or lack of annotated datasets lead to poor quality CV data

### Challenge #4

The key to computer vision model training and retraining is having an accurately labeled and annotated dataset.

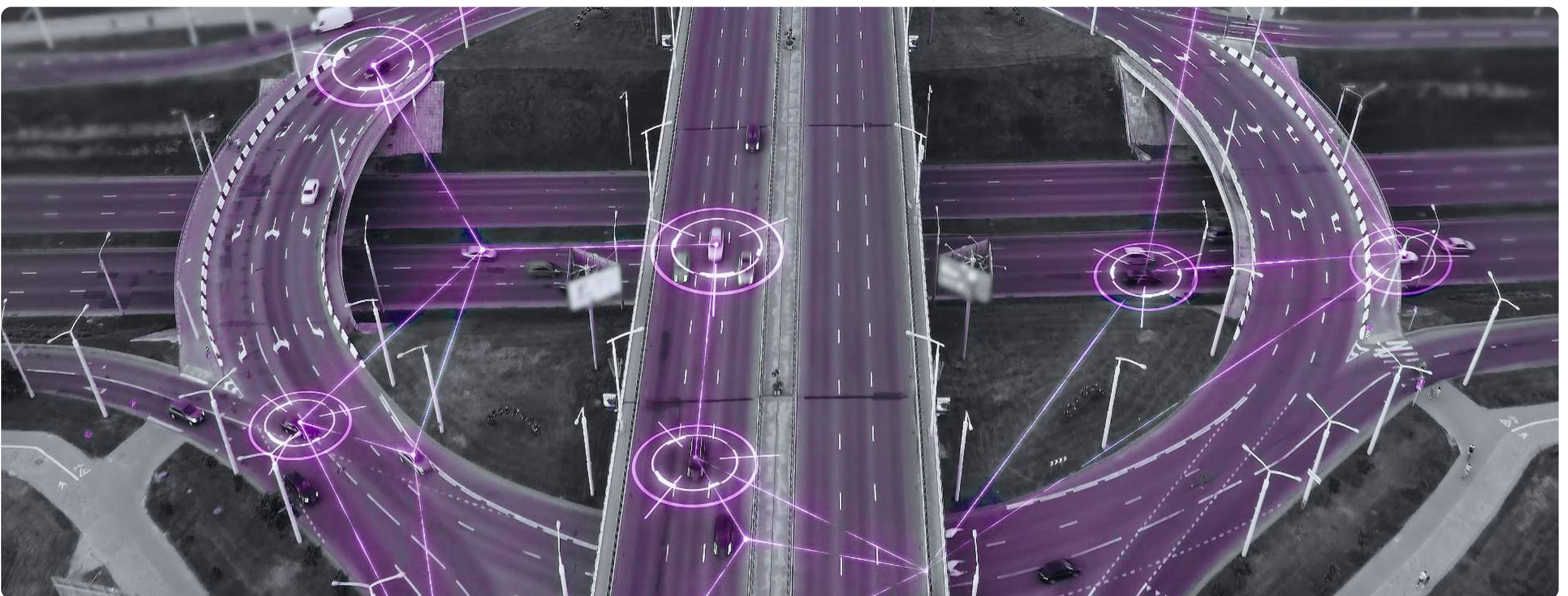
Better training labels make for better models. Labeling errors or shortcuts taken during the data labeling process can immediately result in poor model performance and/or bias in data.

Data labeling and QA annotation tasks require human-in-the loop workflows and demand both labor resources and time.

Many enterprises may have a **massive image library (billions of inferences)**, but do not have the **in-house resources to complete these tasks**. So, they turn to third-party companies like Scale.ai and outsource data labeling which can significantly delay CV model deployment timelines.

Another obstacle is **not being able to collect enough real-world imagery for a CV data set**. For example, healthcare industries may wish to use radiological imagery for computer vision applications, but there is a challenge in obtaining individual data opt-in for utilizing high-fidelity imagery due to data privacy regulations (GCP, HIPAA, GDPR, etc.).

With the rise of synthetic media for certain use cases, the pressure on enterprises to amass large, costly CV datasets may be lessened. However, it is still uncertain how training ML models on 100% synthetic data can impact real-world model accuracy.





## 5 CV models quickly degrade due to anomalies and data drift

### Challenge #5

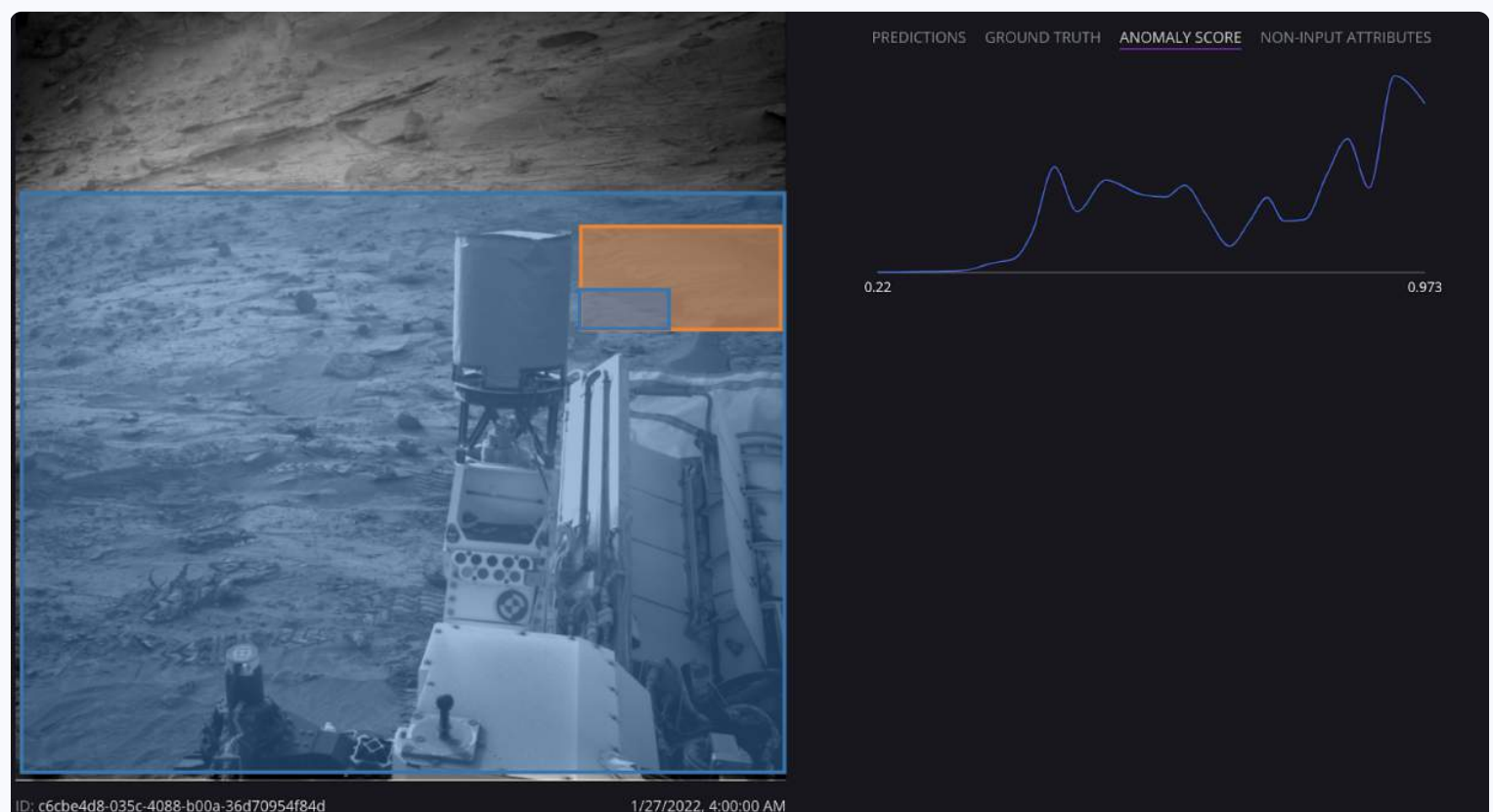
Real world model accuracy (finding ground truth) is also essential to computer vision applications. In the real world, images and streaming video are fluid and dynamic and subject to evolution over time (think radiological images over a year period for a Stage 2 cancer patient). **Your image inputs, which are used to train a ML model and predict outputs, are assumed to be static—but in reality, this assumption is incorrect.**

**Trained model predictions based on past or historical data are no longer valid because visual data drifts. Model performance will degrade over time due to visual data drift and this will be accelerated.**

© Arthur's platform includes two components for detecting visual data drift for single static imagery.

#### ① Image drift

Image drift occurs when you train your ML model on media that is different from the real-world visual data that your CV model sees in production. © With Arthur, every image in your dataset will receive an anomaly score. An anomaly score communicates how different the image is from the entire set of images the model is trained on. Furthermore, we allow for sorting and searching through images based on their anomaly score: how dissimilar are they to the training data.



#### ② Non input metadata

You can include tabular metadata associated with each image when registering your inferences with Arthur, and Arthur will calculate data drift for these fields.

To identify where your model is likely making mistakes, you can monitor CV model pipelines for data anomalies using built-in out-of-distribution detection and track the accuracy of bounding box models. This helps users identify any images being sent to the model where the model is likely to be underperforming.

© Arthur includes recommended and custom alerts as well as both UI and API access for all visual data drift metrics.



## 6 Bias and unfairness in visual datasets introduce model risk

### Bias Detection & Mitigation

With binary or multi-classification outputs for CV model types, bias detection operates the same way as tabular models on the Arthur platform. When you include tabular metadata columns with the ingestion of visual datasets, it increases data accuracy by not only giving you the tools to detect visual drift but also by helping you proactively detect bias in your dataset or model.

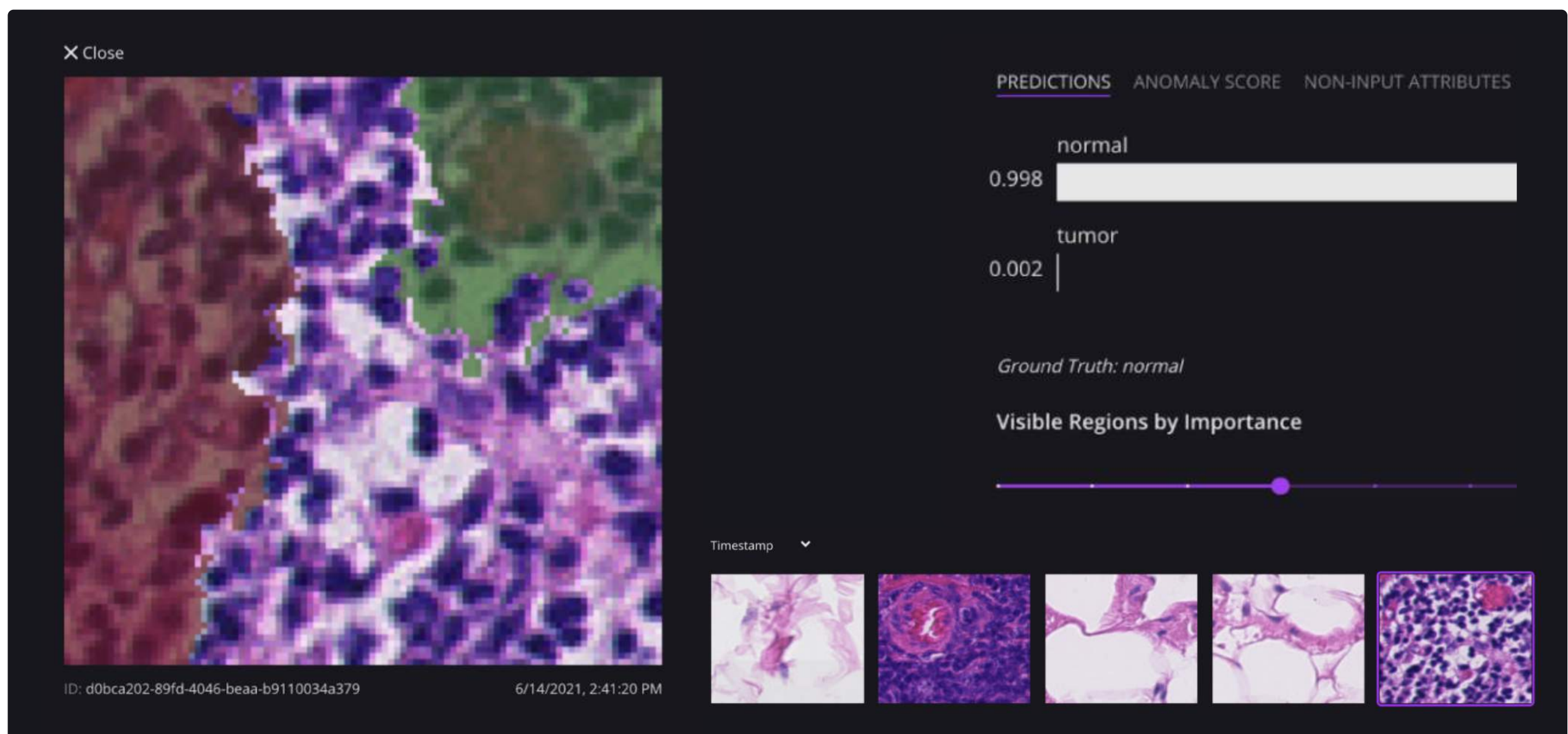
As computer vision continues to offer new opportunities for innovation and growth, we must ensure that its applications are equitable and inclusive to avoid encoding dangerous systemic biases. © Arthur has built-in bias monitoring so you can easily compare equity across various groups, and maintain high standards of fairness.

### Explainability

© Arthur utilizes XAI libraries for supporting CV models and inputs for binary, multi-classification, and regression based output.

This capability gives users a clear visualization of which regions of an image are impactful for a model's decision, making it easy to identify when models perform unexpectedly.

Below is one example of how Arthur gives you the ability to see importance at various levels in the UI.



Using Arthur's local explainability techniques, you can visualize saliency maps over images to reveal which image components were particularly important for the model's decision. The importance scores are associated with a class, so a positive score indicates that a region strongly contributed to a positive class prediction. A negative score indicates that a region was negatively associated with a target class.

© The Arthur platform makes image explanations easy to use—it shades regions of the original image with green or red to indicate the importance of that region. Green shading indicates a region that positively contributed to the selected class, while red shading indicates a region that negatively contributed to the selected class. A user can drag a slider bar interactively, indicating how many regions are shown, sorted by overall importance.

Additionally Arthur gives teams the ability to retrieve explanations directly from API so you can serve them to downstream users.

## The Future

---

In the spirit of advancing computer vision R&D and to inform future MLOps product development, Arthur is partnering with Dumbarton Oaks, a Harvard University research institute and library in Washington, D.C.

**We've been awarded a grant to jointly develop a domain-specific classification model using an archeological antiquities visual dataset, including a web interface for viewing images and visual explanations.** By 2024, the goal is to build an open-source online resource which will include a confidence score of model accuracy (delivered to the researcher who uploaded the photo) and incorporate active learning to accelerate labeling of new historical image collections.

**We're also developing new CV capabilities in close collaboration with our cutting-edge ML R&D team. Stay tuned for our latest CV product capabilities.**



# Glossary

---

## → BOUNDING BOX

The bounding box is an imaginary rectangle drawn around a given object in an image and it serves as the region of interest. Misaligned bounding boxes throw a wrench in your algorithm and can take significant time to diagnose and fix.

## → IMAGE CLASSIFICATION

Given a group of images, the task is to classify them into a set of predefined classes using solely a set of sample images that have already been classified. As opposed to complex topics like object detection and image segmentation, which have to localize (or give positions for) the features they detect, image classification deals with processing the entire image as a whole and assigning a specific label to it.

## → IMAGE SEGMENTATION

Image segmentation is the division of an image into subparts or sub-objects to demonstrate that the machine can discern an object from the background and/or another object in the same image. A “segment” of an image represents a particular class of object that the neural network has identified in an image, represented by a pixel mask that can be used to extract it.

## → PANOPTIC SEGMENTATION

Combination of semantic and instance segmentation. Unlike instance segmentation, semantic segmentation and panoptic segmentation do not require confidence scores associated with each segment.

## → INSTANCE SEGMENTATION

Classifies the objects in the image at a pixel level, like the Semantic Segmentation does, but it can also differentiate different instances of that class. Meaning that if you have cars parked next to each other, if you have semantic segmentation, you can tell that there is a big blob of cars, but with instance segmentation, you can tell that there are 5 distinct cars, and this will probably change what you can do with that information.

## → SEMANTIC SEGMENTATION

When an image is segmented, discreet background features are processed with as much precision as prominent aspects. Semantic segmentation is a huge use case for autonomous vehicles (lanes, cars, pedestrians, signs, traffic lights, sidewalk), virtual try on for e-commerce sites (try on sunglasses, makeup), social media camera apps (person in foreground, change filter/background), etc. Unlike instance segmentation, semantic segmentation and panoptic segmentation do not require confidence scores associated with each segment. For semantic segmentation, IoU, pixel-level accuracy and mean accuracy are commonly used metrics. These metrics ignore object-level labels while considering only those at pixel-level.

## → OBJECT DETECTION (AKA BOUNDING BOXES)

Object detection, as the name suggests, refers to detection and localization of objects using bounding boxes. Object detection looks for class-specific details in an image or a video and identifies them whenever they appear. These classes can be cars, animals, humans, or anything on which the detection model has been trained.

## → OBJECT TRACKING

Object tracking is a deep learning process where the algorithm tracks the movement of an object. In other words, it is the task of estimating or predicting the positions and other relevant information of moving objects in a video.

## → FACIAL RECOGNITION

Facial Recognition is a subpart of object detection where the primary object being detected is the human face. While similar to object detection as a task, where features are detected and localized, facial recognition performs not only detection, but also recognition of the detected face.

# Glossary

---

## → **EDGE DETECTION**

Edge detection is the task of detecting boundaries in objects. It is algorithmically performed with the help of mathematical methods that help detect sharp changes or discontinuities in the brightness of the image. Makes use of discontinuous local features of an image to detect edges and hence define a boundary of the object.

## → **PATTERN MATCHING**

Pattern matching in computer vision refers to a set of computational techniques which enable the localization of a template pattern in a sample image or signal. Such template pattern can be a specific facial feature, an object of known characteristics, or a speech pattern such as a word.

## → **ACTION CLASSIFICATION**

Assign the correct label (e.g. “cooking,” “writing,” etc.) to a given action within a video.

## → **FEATURE MATCHING**

Feature matching or generally image matching, a part of many computer vision applications such as image registration, camera calibration, and object recognition, is the task of establishing correspondences between two images of the same scene/object. A common approach to image matching consists of detecting a set of interest points each associated with image descriptors from image data. Once the features and their descriptors have been extracted from two or more images, the next step is to establish some preliminary feature matches between these images.