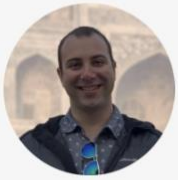
Glossary of Common Computer Vision Terms

Author:



Brad Dwyer

Founder and CTO of Roboflow, building the computer vision infrastructure so you don't have to. Previously founded Hatchlings & created Product Hunt's AR App of the Year. Email me: brad at roboflow.com

Computer Vision (and Machine Learning in general) is one of those fields that can seem hard to approach because there are so many industry-specific words (or common words used in novel ways) that it can feel a bit like you're trying to learn a new language when you're trying to get started.

We've defined many of these unique words, phrases, and acronyms here to help get you over the hump. If you think of any we're missing please let us know, we'll be keeping this list of definitions updated.

- Ablation Study - removing features from your model one by one to see how much each one individually contributes to the performance. Common to see in research papers about new model architectures that contain many novel contributions.
- Accuracy - proportion of "correct" vs "incorrect" predictions a model makes. Common in classification models that have a single correct answer (vs object detection where there is a gradient from "perfect" to "pretty close" to "completely wrong".) Often terms such as "top-5 accuracy" are used which means "how much of the time was the correct answer in the model's top 5 most

confident predictions?" Top-1 accuracy and Top-3 accuracy are also common.

- Activation - The equation of a neural network cell that transforms data as it passes through the network. See activation function.
- Activation Function - The mathematical equation that transforms data input through a neural network. Common activation functions include the sigmoid function and tanh.
- **Anchor Box** - common in object detection models to help predict the location of bounding boxes.
- Annotation - the "answer key" for each image. Annotations are markup placed on an image (bounding boxes for object detection, polygons or a segmentation map for segmentation) to teach the model the ground truth.
- **Annotation Format** - the particular way of encoding an annotation. There are many ways to describe a bounding box's size and position (JSON, XML, TXT, etc) and to delineate which annotation goes with which image.
- **Annotation Group** - describes what types of object you are identifying. For example, "Chess Pieces" or "Vehicles". Classes (eg "rook", "pawn") are members of an annotation group.
- Architecture - a specific neural network layout (layers, neurons, blocks, etc). These often come in multiple sizes whose design is similar except for the number of parameters. For example,**EfficientDet** ranges from D0 (smallest) to D7 (largest).
- AUC - Area Under the Curve. An evaluation metric for the efficacy of a prediction system that is trading off precision at the expense of recall. The precision recall curve is downward sloping as a predictions algorithms confidence is decreased, to allow more, but less precise predictions.
- **Augmentation** - creating more training examples by distorting your input images so your model doesn't overfit on specific training examples. For example, you may flip, rotate, blur, or add noise.
- **AutoML** - one-click to train models that optimize themselves (usually hosted in the cloud). They can be a good starting point, a good baseline, and in some cases a "it just works" solution vs tuning your own models.
- Backbone - an object detection model is made up of three parts, a head, a neck, and a backbone. The backbone is the “base” classification model that the object detection model is based on.
- Backprop - Back propagation is the way that neural networks improve themselves. For each batch of training data, they do a “forward pass” through the network and then find the direction of the “gradient” of each neuron in each layer from the end working backwards and adjust it a little bit in the direction that most reduces the loss function. Over millions of iterations, they get

better bit by bit and this is how they “learn” to fit the training data.

- Bag of Freebies - a collection of augmentation techniques that have been shown to improve performance regardless of model architecture. YOLOv4 and YOLOv5 have built these techniques into their training pipeline to improve performance over YOLOv3 without dramatically changing the model architecture.
- Batch Inference - making predictions on many frames at once to take advantage of the GPU’s ability to perform parallel operations. This can help improve performance if you are doing offline (as opposed to real-time) prediction. It increases throughput (but not FPS).
- Batch Size - the number of images your model is training on in each step. This is a hyperparameter you can adjust. There are pros (faster training) and cons (increased memory usage) to increasing the batch size. It can also affect the model’s overall accuracy (and there is a bit of an art to choosing a good batch size as it is dependent on a number of factors). You may want to experiment with larger or smaller batch sizes.
- **BCCD - blood cell count and detection dataset**. A set of blood cell images taken under a microscope that we commonly used for experimentation.
- Black Box - a system that makes it hard to peek behind the curtain to understand what is going on. Neural networks are often described as black boxes because it can be hard to explain “why” they are making a particular prediction. Model explainability is currently a hot topic and field of study.
- Block - to simplify their description and creation, many computer vision models are composed of various “blocks” which describe a set of inter-connected neurons. You can think of them a bit like LEGO bricks; they interoperate with each other and various configurations of blocks make up a layer (and many layers make up a model).
- Bounding Box - a rectangular region of an image containing an object. Commonly described by its min/max x/y positions or a center point (x/y) and its width and height (w/h) along with its class label.
- Channel - images are composed of one or more channels. A channel has one value for each pixel in the image. A grayscale image may have one channel describing the brightness of each pixel. A color image may have three channels (one for red, green, and blue or hue, saturation, lightness respectively). A fourth channel is sometimes used for depth or transparency.
- Checkpoint - a point-in-time snapshot of your model’s weights. Oftentimes you will capture a checkpoint at the end of each epoch

so you can go back to it later if your model's performance degrades because it starts to overfit.

- Class - a type of thing to be identified. For example, a model identifying pieces on a Chess board might have the following classes: white-pawn, black-pawn, white-rook, black-rook, white-knight, black-knight, white-bishop, black-bishop, white-queen, black-queen, white-king, black-king. The Annotation Group in this instance would be "Chess Pieces".
- **Class Balance** - the relative distribution between the number of examples of each class. Models generally perform better if there is a relatively even number of examples for each class. If there are too few of a particular class, that class is "under-represented". If there are many more instances of a particular class, that class is "over-represented".
- **Classification** - a type of computer vision task that aims to determine only whether a certain class is present in an image (but not its location).
- **COCO** - the Microsoft Common Objects in Context dataset contains over 2 million images in 80 classes (ranging from "person" to "handbag" to "sink"). MS COCO is a standard dataset used to benchmark different models to compare their performance. Its JSON annotation format has also become commonly used for other datasets.
- Colab - Google Colaboratory is a free platform that provides hosted Jupyter Notebooks connected to free GPUs.
- Computer Vision - the field pertaining to making sense of imagery. Images are just a collection of pixel values; with computer vision we can take those pixels and gain understanding of what they represent.
- Confidence - A model is inherently statistical. Along with its prediction, it also outputs a confidence value that quantifies how "sure" it is that its prediction is correct.
- Confidence Threshold - we often discard predictions that fall below a certain bar. This bar is the confidence threshold.
- **Container** - A virtualized environment that packages its dependencies together into a portable environment. Docker is one common way to create containers.
- Converge - over time we hope our models get closer and closer to a hypothetical "most accurate" set of weights. The march towards this maximum performance is called converging. The opposite of convergence is divergence, where a model gets off track and gets worse and worse over time.
- **Convert** - taking annotations or images in one format and translating them into another format. Each model requires input in a specific format; if our data is not already in that format

we need to convert it with a custom script or a tool like Roboflow.

- Convolution - a convolution is a type of block that helps a model learn information about relationships between nearby pixels.
- Convolutional Neural Network (CNN, ConvNet) - the most common type of network used in computer vision. By combining many convolutional layers, it can learn about more and more complex concepts. The early layers learn about things like horizontal, vertical, and diagonal lines and blocks of similar colors, the middle layers learn about combinations of those features like textures and corners, and the final layers learn to combine those features into identifying higher level concepts like “ears” and “clocks”.
- CoreML - A proprietary format used to encode weights for Apple devices that takes advantage of the hardware accelerated neural engine present on iPhone and iPad devices.
- **CreateML** - A no-code training tool created by Apple that will train machine learning models and export to CoreML. It supports classification and object detection along with several types of non computer-vision models (such as sound, activity, and text classification).
- Cross Validation -
- CUDA - NVIDIA’s method of creating general-purpose GPU-optimized code. This is how we are able to use GPU devices originally designed for 3d games to accelerate neural networks.
- CuDNN - NVIDIA’s CUDA Deep Neural Network library is a set of tools built on top of CUDA pertaining specifically to efficiently running neural networks on the GPU.
- curl - a command line program commonly used to upload and download files on UNIX-like operating systems (which is now included with Windows 10 as well).
- **Custom Dataset** - a set of images and annotations pertaining to a domain specific problem. In contrast to a research benchmark dataset like COCO or Pascal VOC.
- Custom Head - an object detection head that has been customized with **anchor boxes** based on a custom dataset.
- Darknet - A C-based neural network framework created and popularized by PJ Reddie, the inventor of the YOLO family of object detection models.
- Data - information of any kind. It could be images, text, sound, or tabular.
- **Dataset** - a collection of data and a ground truth of outputs that you use to train a machine learning model by example. For object detection this would be your set of images (data) and annotations (ground truth) that you would like your model to learn to predict.

- **Deploy** - taking the results of a trained model and using them to do inference on real world data. This could mean hosting a model on a server or installing it to an edge device.
- Differentiable - in order for backprop to work, all of the operations that the neural network performs must be able to have their derivative calculated in order to determine the gradient.
- Distributed - spread across multiple devices. Distributed training usually means using multiple GPUs (often located on separate physical machines) to train your model.
- **Docker** - a common standard for building containers.
- Domain Specific - problems or techniques that are not generally applicable. For example, if you're trying to detect tumors in X-Rays, anything that has to do with the cancer biology is domain-specific because it wouldn't apply to someone working on measuring traffic flows via satellite imagery.
- Download - taking a file from a remote machine and moving it to another machine. For example, you may want to download your model weights from Google Colab to your local machine or download your dataset from Roboflow to a virtual machine on AWS.
- Early Stopping - detecting when your model has reached peak performance and terminating the training job prior to "completion." There are a number of heuristics you can use to determine your model has reached a local maximum; stopping early can prevent overfitting and save you from wasting time and compute resources.
- **Edge Deployment** - deploying to a device that will make predictions without uploading the data to a central server over the Internet. This could be an iPhone or Android device, a Raspberry Pi, a NVIDIA Jetson, a robot, or even a full computer with a GPU located on-site.
- EMA - exponential moving average. Helps to smooth noisy inputs.
- Environment - a set of machine specifications, operating system, programming language, and frameworks. For example, your training environment might be Ubuntu running on a p2.xlarge on AWS EC2 using PyTorch 1.6.
- Epochs - the number of times to run through your training data.
- **EXIF** - metadata attached to images (for example, orientation, GPS data, information about the capture device, shutter speed, f-stop, etc).
- Export - In Roboflow, an export is a serialized version of a dataset that can be downloaded.
- F1 - A measure of efficacy of a prediction system. F1 is a combination of recall (guessing enough times) with precision (guessing correctly when the system does guess). High F1 means guessing correctly when there is a guess to be made.

- False Negative - when your model fails to predict an object that is actually present.
- False Positive - when you model predicts that an object is present when it actually isn't.
- Family - a set of models that are all related to each other. For example, the YOLO family of models follow a lineage from YOLOv1 all the way to YOLOv5. The core concepts of the models are all the same but they have had new techniques and improvements bolted on as time has progressed.
- **FastAI** - A library built on top of PyTorch for rapid prototyping and experimentation. There is a companion course that teaches the fundamentals of machine learning.
- Feature - a derived property of your data that is learned by your model. For example, a set of convolutions may learn how to recognize zigzag lines in images. Zigzag lines are then a learned feature.
- Feature Fusion - Combining derivative data features in a neural network.
- Feature Pyramid Network (FPN) - A basic feature fusion strategy in object detectors. Combines convolutional neural network features sequentially.
- **Filter Null** - removing some proportion of null examples from your dataset so that your model doesn't learn to optimize its loss function by predicting "null" too often.
- **FLIR** - forward looking infrared. Infrared measures the heat of objects in the infrared spectrum rather than the color of an object in the visual spectrum. Models can be trained on infrared images as well as visual.
- FLOPS - floating point operations per second (used as a measure of computing power). For example, you may see a GPU purport to do 8 TFLOPS which menas 8 trillion floating point operations per second.
- FP8 - 8-bit floating point. (Also known as quarter-precision.) Reducing the precision of your model can improve its speed and accuracy and can also take advantage of features of newer GPUs like tensor cores.
- FP16 - 16-bit floating point. (Also known as half-precision.)
- **FPS** - frames per second. In real-time inference, this is the measure of how many sequential inference operations a model can perform. A higher number means a faster model.
- **Framework** - Deep learning frameworks implement neural network concepts. Some are designed for training and inference - TensorFlow, PyTorch, FastAI, etc. And others are designed particularly for speedy inference - OpenVino, TensorRT, etc.
- GAN Synthesis - using a generative adversarial network to create more training data.

- Generalize - the ability of a model to make accurate predictions on input data it has never seen before.
- Generate - in Roboflow, generating images means processing them into their final form (including preprocessing and augmenting them).
- GPU - graphics processing unit. Originally developed for use with 3d games, they're very good at performing matrix operations which happen to be the foundation of neural networks. Training on a GPU lets your model's calculations run in parallel which is vastly faster than the serial operations a CPU performs (for the subset of operations they are capable of).
- GPU Memory - the amount of information your GPU can fit on it. A bigger GPU will be able to process more information in parallel which means it can support bigger models (or bigger batch sizes) without running out of memory. If you run out of GPU memory it will crash your program.
- Gradient - neural networks use gradient descent to improve bit by bit. The gradient is a set of directions calculated (by taking the derivative of the loss function) that will most improve predictions. By taking a short step in the direction of the gradient and then recalculating the gradient and repeating the process, a neural network can improve its performance over the course of training.
- Ground Truth - the "answer key" for your dataset. This is how you judge how well your model is doing and calculate the loss function we use for gradient descent. It's also what we use to calculate our metrics. Having a good ground truth is extremely important. Your model will learn to predict based on the ground truth you give it to replicate.
- Head - The portion of an object detector where prediction is made. The head consumes features produced in the neck of the object detector.
- **Health Check** - a set of tools in Roboflow that help you understand the composition of your dataset (eg size, dimensions, class balance, etc).
- **Hold Out Set** - another name for "test set" -- the part of your dataset you reserve for after training is complete to check how well your model generalizes.
- **Hosted Dataset** - Robofow stores your dataset in the cloud (secured by your API keys) so that you can access it from whichever machine you are training on.
- **Hosted Model** - A set of trained weights located in the cloud that you can receive predictions from via an API. (As opposed to an edge-deployed model.)
- Hyperparameter - the levers by which you can tune your model during training. These include things like learning rate and

batch size. You can experiment with changing hyperparameters to see which ones perform best with a given model for your dataset.

- **Inference** - making predictions using the weights you save after training your model.
- **IoU** - intersection over union (also abbreviated I/U). A metric by which you can measure how well an object detection model performs. Calculated by taking the amount of the predicted bounding box that overlaps with the ground truth bounding box divided by the total area of both bounding boxes.
- **Jetson** - an edge computing device created by NVIDIA that includes an onboard GPU.
- JSON - a freeform data serialization format originally created as part of JavaScript but now used much more broadly. Many **annotation formats** use JSON to encode their bounding boxes.
- Jupyter Notebook - a common data science tool that enables you to execute Python code visually. Each “cell” in the notebook is a block of code that you can execute by hitting “Ctrl+Enter”. The results of the execution are displayed below the cell.
- Keypoint Detection - a type of computer vision model that predicts points (as opposed to boxes in object detection). Oftentimes keypoint detection is used for human pose estimation or finger tracking where only the position of an object, not its size, matters.
- Label - the class of a specific object in your dataset. In classification, this is the entirety of the prediction. In object detection, it is the non-spatial component of the bounding box.
- Layer - layers are made up of neurons (and, more commonly, blocks of neurons). Deep neural networks are composed of several layers. The neurons in each layer are connected to neurons in one or more other layers. Adding layers makes a network “deeper”. As a network gets deeper it becomes more complex which gives it more predictive power (but also makes it harder to train as it exponentially increases the solution space).
- Learning Rate - a hyperparameter that defines the size of the steps along the gradient that you take after each batch during training. Often the learning rate will change over the course of training (this is called having a “cyclical learning rate”. If your learning rate is too small, your model will converge very slowly. If it’s too large, it might lead your model’s weights to explode and your model to diverge.
- LiDAR - laser imaging detection and ranging. This is a device that uses lasers to Detect depth. Built into many self-driving cars and now included in the iPad Pro for constructing a 3d world map for use in augmented reality.

- Localization - identifying where in an image an object resides. This is the part of object detection and keypoint detection that gives the x/y coordinates (as opposed to the class label).
- Loss Function - A differentiable calculation of “how far off” a prediction is. This is used to calculate the gradient and in turn steer which direction your model steps at each iteration of your training loop. The output of the loss function is called the “loss” and is usually calculated on the training set and validation set separately (called “training loss” and “validation loss” respectively). The lower this value, the more accurate the model’s predictions were.
- Machine Learning - a field of teaching computers by example. Instead of traditional programming where you write the “rules” by which your program converts inputs to outputs, you instead give it many examples of inputs and the desired output and let it write the rules by (smart) trial and error.
- **mAP** - mean average precision. A metric to judge how well your object detection model is performing. We have **a whole post breaking this down**.
- Memory Footprint - how much space in memory a model takes. This is largely a function of the number of parameters in your model and your batch size. You want to make sure this fits inside of your GPU memory.
- Metadata - ancillary information stored about your data. For example, the date and time it was collected. Often stored as EXIF.
- Metrics - Evaluation metrics are used to assess the performance of a machine learning system.
- Mixed Precision - using both full precision and half precision floating point numbers during training. This has been shown to increase speed without degrading performance.
- **Mobile Deployment** - deploying to an edge device like a mobile phone. Considerations like battery usage and heat dissipation come into play.
- **Model** - a specific incarnation of an architecture. A model has a defined input size and layout for its weights file. For example, YOLOv5s is the smallest version of YOLOv5 which is an architecture in the YOLO family.
- Model Configuration - used to adjust an architecture into a specific model and set its hyperparameters.
- Model Size - the number of parameters (or neurons) a model has. This can also be measured in terms of the size of the weights file on disk.
- **Model Zoo** - a collection of model architectures (and sometimes pre-trained model weights) available for download.

- **Mosaic** - an advanced augmentation that combines multiple images from your training set that has been shown to improve object detection training performance.
- Neck - The portion of an object detection model that forms features from the base convolutional neural network backbone.
- Neural Architecture Search - automatically trying many variations of model layouts and hyperparameters to find the optimal configuration.
- Neuron - also known as a parameter, a neuron or perceptron is a mathematical functions takes several inputs and outputs, multiplies them together with its weights (which change over time as the network learns) and outputs a single value which is then fed into other neurons as one of their inputs.
- NMS - non maximum suppression.
- Non-Destructive - an operation that can be reversed without losing information. Roboflow's preprocessing and augmentation steps are non-destructive because they do not overwrite the source values.
- Normalization - standardizing data inputs based on a distribution.
- Novel - an insight is novel if it is a new idea.
- **nvidia-smi** - a tool that can be used to inspect the state of the GPU on machines with an NVIDIA graphics card. You can use this command's output to determine how much of the GPU memory is being consumed at a given point in time, for example.
- **NVIDIA Container Toolkit** - a helper library to assist in creating Docker containers that can access their host machine's GPU.
- **Null Annotation** - a purposely empty annotation that can help teach your model that an object is not always present.
- **Object Detection** - a category of computer vision models that both classify and localize objects with a rectangular bounding box.
- Occlusion - when an object is partially obscured behind another object it is "occluded" by that object. It is important to simulate occlusion so your model isn't overly dependent on one distinctive feature to identify things (eg by occluding a cat's ears you force it also to learn about its paws and tail rather than relying solely on the ears to identify it which helps if the cat's head ends up being hidden by a chair in a real world situation).
- Offline Prediction - the opposite of "real-time" prediction, this is when the model does not have a hard limit on how quickly it needs to return an answer. An example would be indexing a user's photo library to do search. This task can wait to be performed while the user is sleeping and the device is not otherwise in use.

- ONNX - a cross-platform, cross-framework serialization format for model weights. By converting your weights to ONNX you can simplify the dependencies required to deploy it into production (oftentimes converting to ONNX is a necessary step in deploying to the edge and may be an intermediary step in converting weights to another format).
- **Ontology** - the categorization and hierarchy of your classes. As your project grows it becomes more and more important to standardize on common nomenclature conventions for your team.
- OpenCV - an "traditional" computer vision framework popularized before deep learning became ubiquitous. It excels at doing things like detecting edges, image sticking and object tracking. In recent years it has also started to expand into newer machine learning powered computer vision techniques as well.
- OpenVINO - Intel's inference framework. Designed for speedy inference on CPU and VPU devices.
- Output - the result of a process. The output of training is a set of weights. The output of inference is a prediction.
- **Outsourced Labeling** - paying people to annotate and/or label your images. There are several companies specializing in taking on this task. It's most effective when little domain expertise is required to determine the correct annotation (and difficult to do in cases like plant disease detection where an expert is needed to provide an accurate ground truth).
- Overfitting - if your model starts memorizing specific training examples to such an extent that it starts degrading its performance on the validation set. Tactics to counteract overfitting include collecting more training data, augmentation, and regularization.
- **PaddlePaddle** - a deep learning framework developed by Baidu.
- Paper - (usually peer-reviewed) academic literature describing novel techniques. Academic papers are often freely published to sites like arXiv.
- Parameters - the number of weights. There is one parameter for each connection between two neurons in the network. Each one is stored as a floating point number and is adjusted during each backpropagation step during training.
- **Pascal VOC** - the Visual Object Classes was an early benchmark dataset. It has largely been replaced with newer datasets like COCO in the literature but its XML annotation format has become widely used by other datasets, labeling tools, and models.
- Performance - how fast and accurate your model is.
- Platform - a **computer vision platform** is a (usually cloud-hosted) meta-tool that ties into various other tools to manage all (or part of) your pipeline.

- Pipeline - your **computer vision pipeline** is the process of going from raw images to a prediction. Usually this encompasses collecting images, annotation, data inspection and quality assurance, transformation, preprocessing and augmentation, training, evaluation, deployment, inference (and then repeating the cycle to improve the predictions).
- Polygon - a (usually non-rectangular) region defining an object with more detail than a rectangular bounding box. Polygon annotations can be used to train segmentation models or to enhance performance of object-detection models by enabling a more accurate bounding box to be maintained after augmentation.
- Precision - A measure of how precise a model is at prediction time. True positives divided by all positives that have been guessed.
- Prediction - an attempt by a model to replicate the ground truth. A prediction usually contains a confidence value for each class.
- Preprocessing - deterministic steps performed to all images (training, validation, testing, and production) prior to feeding them into the model.
- Pretrained Model - a model that has already been trained on another dataset. Many things it learns will be broadly applicable to images in other datasets (for example, finding lines, corners, and patterns of colors). Pre-training on a large dataset like COCO can reduce the number of custom images you need to obtain satisfactory results.
- Production - the deployment environment where the model will run in the wild on real-world images (as opposed to the testing environment where the model is developed).
- PyTorch - a popular open source deep learning framework developed by Facebook. It has a focus on accelerating the path from research prototyping to production deployment.
- RasPi - the Raspberry Pi is an inexpensive Linux-based microcomputer with a broad ecosystem of compatible peripherals.
- Realtime - when a model needs to be run in a specified amount of time, for example in a mobile augmented reality application it needs to provide its predictions in less time than the desired frame rate so that it can keep up with the incoming images. Doing batch prediction to take advantage of parallel processing doesn’t help here because as soon as the next image comes in the previous prediction is no longer as relevant.
- Recall - A measure of performance for a prediction system. Recall is used to assess whether a prediction system is guessing enough. True positives / All possible true positives.
- Region Attribute - additional properties beyond the class name and location that can be added to bounding boxes and polygons in

some annotation tools. They can be thought of as metadata at the object (rather than the image) level.

- Regression - a model that predicts one or multiple real numbers (such as an age, year, or pixel location) where "how close" a prediction is to the ground truth is measurable (as opposed to classification where the prediction is either right or wrong).
- Regularization - A technique to reduce bias in machine learning models. Machine learning models have a tendency to overfit to training data. Regularization introduces a penalty for heavily weighting features, forcing a machine learning system to formulate a flexible algorithm.
- **Remap** - changing class composition after the annotation task. Sometimes your ontology needs to evolve as you learn more about the problem and your process matures.
- Repo - short for repository. A repo is a version-controlled storage place for things like code that allows for change-tracking and team collaboration.
- Requirements - the third-party code needed to replicate your environment. These could be libraries, frameworks, or drivers. For Python projects, the requirements are stored in a requirements.txt file. For Docker containers, they are defined by the Dockerfile.
- **Resolution** - the number of pixels in an image (defined by its width multiplied by its height). The standard unit of measure is megapixels (millions of pixels).
- Runtime Environment - Where machine learning code is being executed. CPU, GPU, VPU (Vision Processing Unit), or TPU.
- **SageMaker** - Amazon AWS' machine learning platform encompassing tools for outsourced annotation, model training, and deployment.
- Segmentation - a type of model that classifies each individual pixel used when the exact outlines of objects are needed.
- Self Adversarial Training - a technique where the model strategically starves itself of the information it is most reliant on to force itself to learn other ways to make predictions. For example, if it detects it is mostly relying on cat ears to identify cats it will turn off the parts of the input feeding into those neurons to force it to also learn other ways to identify cats (like its paws and tail).
- Session - A TensorFlow Session allocates resources for a machine to execute a TensorFlow defined neural network graph structure. Sessions are deprecated in TensorFlow 2.
- **Split** - segregating subsets of your data and delineating them for different purposes. Usually we create three splits: train (given to your model to mimic), valid (used for evaluation during training), and test (held back until the very end to determine how well your model will generalize).

- SSD - single shot detector. A model that only does a single pass to both localize and classify objects. The YOLO family famously plays off of this concept in its name: You Only Look Once.
- State of the Art - a model that is currently top of its class, performing better on the benchmark dataset than any other previously known model.
- Subjective - opposite of objective; performance that is observed intuitively but not necessarily able to be measured. For example, in language modeling, it's common for models' metrics to be similar but for one of their outputs to be subjectively better as judged by a human reader.
- **Synthetic Data** - images that are created rather than collected. There are several strategies for creating more training data including using 3D models, GAN synthesis, and context augmentation.
- Tensor - a (possibly multi-dimensional) array of numbers of a given type with a defined size. Because they have a defined size and shape it makes it possible to optimize and parallelize operations on them with hardware accelerators.
- Tensor Core - NVIDIA's brand name for the part of their GPUs that is specifically optimized for deep learning (and especially mixed-precision neural networks).
- Tensorboard - a tool used to track and visualize training metrics including graphs of common statistics like loss and mean average precision originally developed for Tensorflow but now compatible with other frameworks like PyTorch.
- Tensorflow - Google's popular open source deep learning framework.
- Tensorflow Lite - model serialization for Tensorflow models to optimize them to run on mobile and edge devices.
- TensorRT - NVIDIA's framework agnostic inference optimization tooling. Helps to optimize models for deployment on NVIDIA-powered edge-devices.
- **Test Set Bleed** - an issue that occurs when data from your test set leaks into your training set. This is bad because it defeats the purpose of your hold out set; you no longer have any way of judging how well your model will generalize to predict on images it has never seen before.
- TFJS - Tooling that enable (some) Tensorflow-trained models to perform inference in the web browser with Javascript, WebAssembly, and WebGPU.
- **TFRecord** - a binary data format compatible with Tensorflow. In the object detection API all of the images and annotations are stored in a single file.
- **Tile** - splitting an image into a grid of smaller images and running them through a model independently to boost the effective

resolution. This can be an effective strategy to improve model accuracy while still fitting into the available memory (at the expense of having to run the model several times per image).

- TPU - Tensor Processing Unit. Google's hardware accelerator for performing operations on tensors. It is much faster than a GPU for some workloads. Most often they are run on Google Cloud or Google Colab but there is also an edge-TPU that can be deployed in the field.
- Tradeoff - when two competing concerns pull you in opposite directions. For example, there is often a tradeoff between speed and accuracy (where there is a continuum from fast and inaccurate to slow and accurate with a continuum of valid choices in between depending on the needs of your specific problem).
- Train - the process iteratively of adjusting your model's parameters to converge on the weights that optimally mimic the training data.
- Transfer Learning - using pre-trained weights to bootstrap your model's learning. You are "transferring" the knowledge learned on another dataset and then "fine-tuning" it to learn about your new domain.
- Tune - adjusting hyperparameters to find the optimal settings to get the best model.
- **Two Stage Detector** - a category of (typically older generation) object detection models that first localize, then classify. As opposed to single shot detectors which do both tasks in one pass.
- **Validate** - during the training process of a neural network, the validation set is used to assess how well the model is generalizing. These examples are not used to calculate the gradient; they are the ones used to calculate your metrics and see how well they are improving over time.
- Version - a point in time snapshot of your dataset. By keeping track of exactly which images, preprocessing, and augmentation steps were used in each iteration of your model you maintain the ability to reproduce the results and scientifically test across various models and frameworks while remaining confident that the results are attributable to the model changes and not due to a bug in the data pipeline.
- Weights - the parameters of your model that neurons use to determine whether or not to fire. The optimal values are learned via backpropagation during training and can then be serialized and deployed for inference.
- Workflow - The process you follow. This will be some combination of manual steps, custom code, and third party tools in any number of environments. A **computer vision platform** can help you set up an optimal workflow.

- **XML** - a hierarchical data format (HTML, the markup language defining the layout and content of the page you are currently reading, is a subset of XML). In computer vision XML is most commonly used with the Pascal VOC XML annotation format.
- YAML - a markup language originally invented by Yahoo that is now commonly used as a format for configuration files (notably in **YOLOv5's YAML configuration**).
- **YOLO** - You Only Look Once, a family of single-shot learner object detection models providing state of the art results for object detection as of fall 2020.