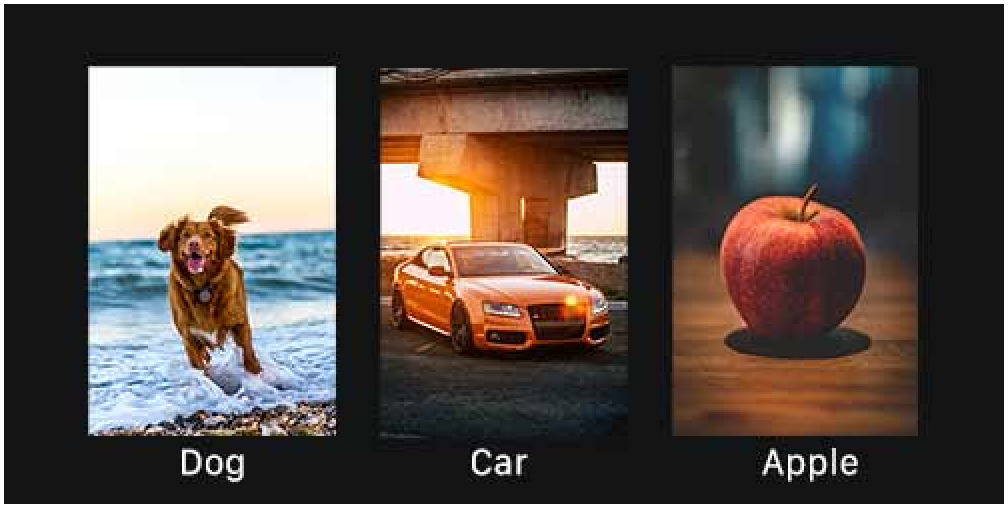
Image Recognition Guide

fritz.ai/image-recognition/

Almost everything you need to know about how image recognition works

Introduction

Image recognition is a computer vision technique that allows machines to interpret and categorize what they “see” in images or videos. Often referred to as “image classification” or “image labeling”, this core task is a foundational component in solving many computer vision-based machine learning problems.



But how does image recognition actually work? What are the different approaches, what are its potential benefits and limitations, and how might you use it in your business?

In this guide, you’ll find answers to all of those questions and more. Whether you’re an experienced machine learning engineer considering implementation, a developer wanting to learn more, or a product manager looking to explore what’s possible with computer vision and image recognition, this guide is for you.

Here’s a look at what we’ll cover:

Part 1: Image recognition – the basics

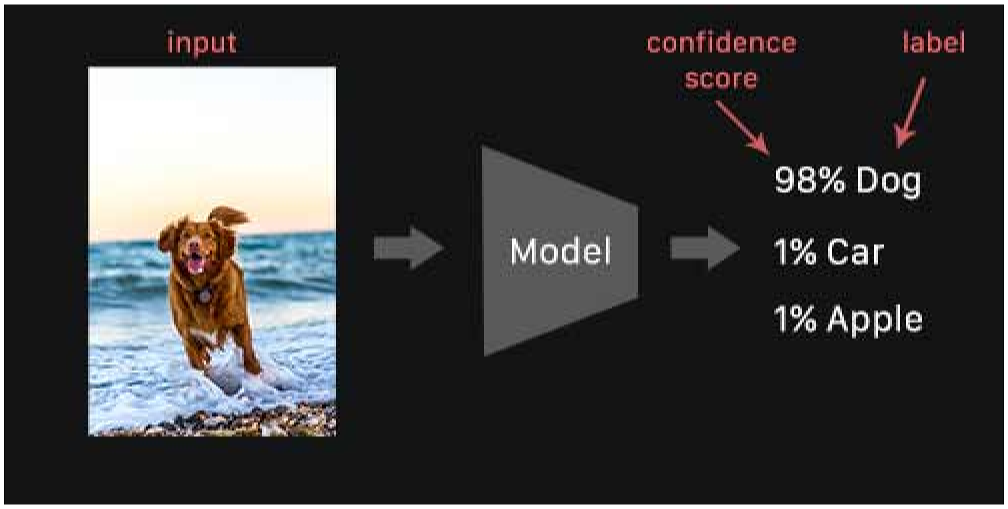
What is image recognition?

Image recognition is a computer vision task that works to identify and categorize various elements of images and/or videos. Image recognition models are trained to take an image as input and output one or more labels describing the image. The set of possible output labels are referred to as **target classes**. Along with a predicted class, image recognition models may also output a confidence score related to how certain the model is that an image belongs to a class.

For instance, if you wanted to build an image recognition model that automatically determined whether or not a dog was in a given image, the pipeline would, broadly speaking, look like this:

- Image recognition model trained on images that have been labeled as “dog” or “not dog”
- Model input: Image or video frame
- Model output: Class name (i.e. dog) with a confidence score that indicates the likelihood of that image containing that class of object.

Here’s what this looks like in practice:



Modes and types of image recognition

Image recognition is a broad and wide-ranging computer vision task that's related to the more general problem of pattern recognition. As such, there are a number of key distinctions that need to be made when considering what solution is best for the problem you're facing.

Broadly speaking, we can break image recognition into two separate problems: **single** and **multiclass recognition**. In single class image recognition, models predict only one label per image. If you're training a dog or cat recognition model, a picture with a dog and a cat will still only be assigned a single label. In cases where only two classes are involved (dog; no dog), we refer to these models as **binary classifiers**.

Multiclass recognition models can assign several labels to an image. An image with a cat and a dog can have one label for each. Multiclass models typically output a confidence score for each possible class, describing the probability that the image belongs to that class.

While there are a number of traditional statistical approaches to image recognition (linear classifiers, Bayesian classification, support vector machines, decision trees, etc.), this guide will focus on image recognition techniques that employ neural networks, as those have become the state-of-the-art approaches to image recognition.

Why is image recognition important?

Image recognition is one of the most foundational and widely-applicable computer vision tasks.

Recognizing image patterns and extracting features is a building block of other, more complex computer vision techniques (i.e. object detection, image segmentation, etc.), but it also has numerous standalone applications that make it an essential machine learning task.

Image recognition's broad and highly-generalizable functionality can enable a number of transformative user experiences, including but not limited to:

- Automated image organization
- User-generated content moderation
- Enhanced visual search
- Automated photo and video tagging
- Interactive marketing/Creative campaigns

Of course, this isn't an exhaustive list, but it includes some of the primary ways in which image recognition is shaping our future.

Part 2: How does image recognition work?

Now that we know a bit about what image recognition is, the distinctions between different types of image recognition, and what it can be used for, let's explore in more depth how it actually works.

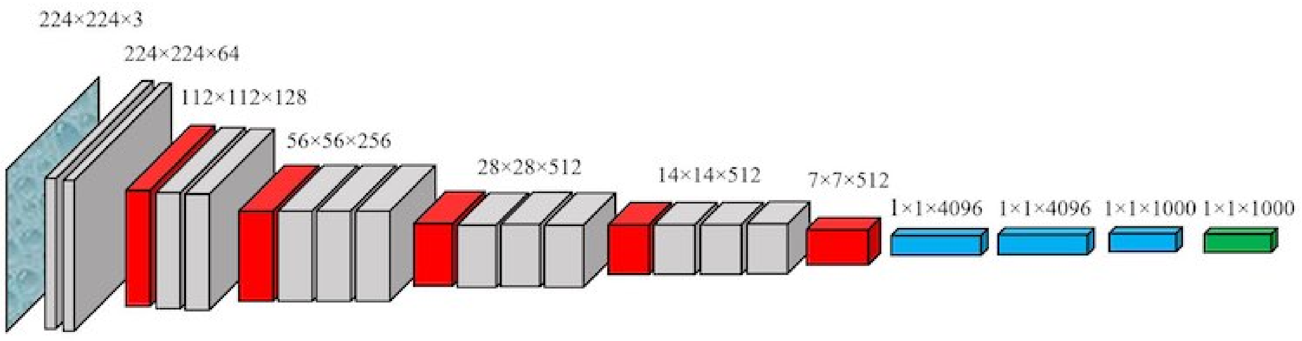
In this section, we'll look at several deep learning-based approaches to image recognition and assess their advantages and limitations. While there are a number of traditional methods—including the ones mentioned above—for the purposes of this overview, we're going to look at the approaches that use neural networks, which have become the state-of-the-art methods for image recognition.

Popular image recognition benchmark datasets include CIFAR, ImageNet, COCO, and Open Images. Though many of these datasets are used in academic research contexts, they aren't always representative of images found in the wild. As such, you should always be careful when generalizing models trained on them. For example, a full 3% of images within the COCO dataset contains a toilet.

Basic structure

In general, deep learning architectures suitable for image recognition are based on variations of convolutional neural networks (CNNs). For a gentle introduction to CNNs, check out this overview.

Nearly all image recognition models begin with an encoder. Encoders are made up of blocks of layers that learn statistical patterns in the pixels of images that correspond to the labels they're attempting to predict. High performing encoder designs featuring many narrowing blocks stacked on top of each other provide the "deep" in "deep neural networks". The specific arrangement of these blocks and different layer types they're constructed from will be covered in later sections.



The encoder is then typically connected to a **fully connected** or **dense** layer that outputs confidence scores for each possible label. It's important to note here that image recognition models output a confidence score for every label and input image. In the case of single-class image recognition, we get a single prediction by choosing the label with the highest confidence score. In the case of multi-class recognition, final labels are assigned only if the confidence score for each label is over a particular threshold.

Finally, a note about accuracy. Most image recognition models are benchmarked using common accuracy metrics on common datasets. **Top-1 accuracy** refers to the fraction of images for which the model output class with the highest confidence score is equal to the true label of the image. **Top-5 accuracy** refers to the fraction of images for which the true label falls in the set of model outputs with the top 5 highest confidence scores.

Model architecture overview

Many neural network architectures exist for image recognition. Given the simplicity of the task, it's common for new neural network architectures to be tested on image recognition problems and then applied to other areas, like object detection or image segmentation. This section will cover a few major neural network architectures developed over the years.

AlexNet

AlexNet, named after its creator, was a deep neural network that won the ImageNet classification challenge in 2012 by a huge margin. Though it wasn't the first convolution neural network to be used for image recognition or even win this particular challenge, it's widely credited with sparking a resurgence of interest in using deep convolutional neural networks to solve computer vision problems. The network, however, is relatively large, with over 60 million parameters and many internal connections, thanks to dense layers that make the network quite slow to run in practice.

VGGNet

Two years after AlexNet, researchers from the Visual Geometry Group (VGG) at Oxford University developed a new neural network architecture dubbed VGGNet. VGGNet has more convolution blocks than AlexNet, making it "deeper", and it comes in 16 and 19 layer varieties, referred to as VGG16 and VGG19, respectively.

The deeper network structure improved accuracy but also doubled its size and increased runtimes compared to AlexNet. Despite the size, VGG architectures remain a popular choice for server-side computer vision models due to their usefulness in transfer learning. VGG architectures have also been found to learn hierarchical elements of images like texture and content, making them popular choices for training style transfer models.

Inception

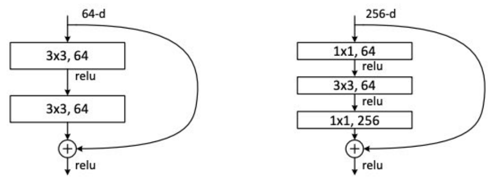
The Inception architecture, also referred to as GoogLeNet, was developed to solve some of the performance problems with VGG networks. Though accurate, VGG networks are very large and require huge amounts of compute and memory due to their many densely connected layers.

The Inception architecture solves this problem by introducing a block of layers that approximates these dense connections with more sparse, computationally-efficient calculations. Inception networks were able to achieve comparable accuracy to VGG using only one tenth the number of parameters.

ResNet

The success of AlexNet and VGGNet opened the floodgates of deep learning research. As architectures got larger and networks got deeper, however, problems started to arise during training. When networks got too deep, training could become unstable and break down completely.

ResNets, short for **residual networks**, solved this problem with a clever bit of architecture. Blocks of layers are split into two paths, with one undergoing more operations than the other, before both are merged back together. In this way, some paths through the network are deep while others are not, making the training process much more stable over all. The most common variant of ResNet is ResNet50, containing 50 layers, but larger variants can have over 100 layers. The residual blocks have also made their way into many other architectures that don’t explicitly bear the ResNet name.



SqueezeNet

Even the smallest network architecture discussed thus far still has millions of parameters and occupies dozens or hundreds of megabytes of space. SqueezeNet was designed to prioritize speed and size while, quite astoundingly, giving up little ground in accuracy.

Despite being 50 to 500X smaller than AlexNet (depending on the level of compression), SqueezeNet achieves similar levels of accuracy as AlexNet. This feat is possible thanks to a combination of residual-like layer blocks and careful attention to the size and shape of convolutions. SqueezeNet is a great choice for anyone training a model with limited compute resources or for deployment on embedded or edge devices.

MobileNet

The MobileNet architectures were developed by Google with the explicit purpose of identifying neural networks suitable for mobile devices such as smartphones or tablets. They're typically larger than SqueezeNet, but achieve higher accuracy.

MobileNet architectures brought two important innovations to network designs: **depthwise separable convolutions** and a hyperparameter known as a **width multiplier**. Depthwise separable convolutions are a replacement for traditional convolution layers, having fewer parameters and being more computationally efficient. The width multiplier is a parameter that controls how many parameters are used for each convolution layer. This allows for the creation of multiple networks along a tradeoff curve of size and speed versus accuracy. A continuum of models can be created with the same basic architecture so that more powerful devices can receive larger, more accurate models, while less powerful devices can use smaller, less accurate models.

Neural Architecture Search

For much of the last decade, new state-of-the-art results were accompanied by a new network architecture with its own clever name. In certain cases, it's clear that some level of intuitive deduction can lead a person to a neural network architecture that accomplishes a specific goal.

As with many tasks that rely on human intuition and experimentation, however, someone eventually asked if a machine could do it better. Neural architecture search (NAS) uses optimization techniques to automate the process of neural network design. Given a goal (e.g model accuracy) and constraints (network size or runtime), these methods rearrange composible blocks of layers to form new architectures never before tested. Though NAS has found new architectures that beat out their human-designed peers, the process is incredibly computationally expensive, as each new variant needs to be trained.

It's estimated that some papers released by Google would cost millions of dollars to replicate due to the compute required. For all this effort, it has been shown that random architecture search produces results that are at least competitive with NAS.

One final fact to keep in mind is that the network architectures discovered by all of these techniques typically don't look anything like those designed by humans. For all the intuition that has gone into bespoke architectures, it doesn't appear that there's any universal truth in them. They simply work and, in many cases, that's enough.

How image recognition works on the edge

If your use case requires that image recognition work in real-time, without internet connectivity, or on private data, you might be considering running your image recognition model directly on an edge device like a mobile phone or IoT board.

In those cases, you’ll need to choose specific model architectures to make sure everything runs smoothly on these lower power devices. Here are a few tips and tricks to ensure your models are ready for edge deployment:

- Prune your network to include fewer convolution blocks. Most papers use network architectures that are not constrained by compute or memory resources. This leads to networks with far more layers and parameters than are required to generate good predictions.
- Add a width multiplier to your model so you can adjust the number of parameters in your network to meet your computation and memory constraints. The number of filters in a convolution layer, for example, greatly impacts the overall size of your model. Many papers and open-source implementations will treat this number as a fixed constant, but most of these models were never intended for mobile use. Adding a parameter that multiplies the base number of filters by a constant fraction allows you to modulate the model architecture to fit the constraints of your device. For some tasks, you can create much, much smaller networks that perform just as well as large ones.
- Shrink models with quantization, but beware of accuracy drops. Quantizing model weights can save a bunch of space, often reducing the size of a model by a factor of 4 or more. However, accuracy will suffer. Make sure you test quantized models rigorously to determine if they meet your needs.
- Input and output sizes can be smaller than you think! If you’re designing a photo organization app, it’s tempting to think that your image recognition model needs to be able to accept full resolution photos as an input. In most cases, edge devices won’t have nearly enough processing power to handle this. Instead, it’s common to train image recognition models at modest resolutions, then downscale input images at runtime.

To see just how small you can make these networks with good results, check out this post on creating a tiny image recognition model for mobile devices.

Back to top

Part 3: Use cases and applications

In this section, we'll provide an overview of real-world use cases for image recognition. We've mentioned several of them in previous sections, but here we'll dive a bit deeper and explore the impact this computer vision technique can have across industries.

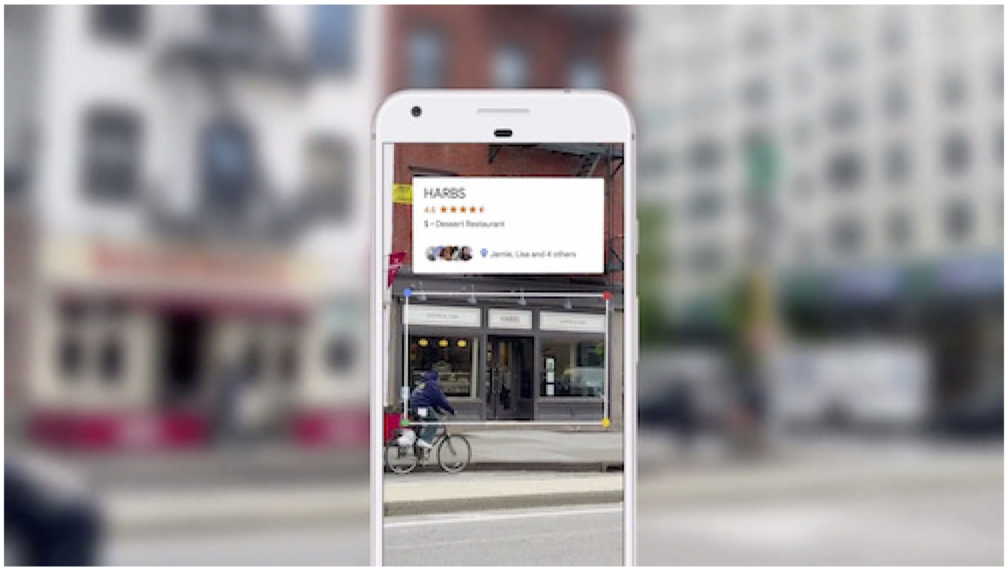
Specifically, we'll examine how image recognition can be used in the following areas:

- Visual search
- Image organization
- Content moderation
- Accessibility

Visual search

Broadly speaking, visual search is the process of using real-world images to produce more reliable, accurate online searches. Visual search allows retailers to suggest items that thematically, stylistically, or otherwise relate to a given shopper's behaviors and interests.

Using a deep learning approach to image recognition allows retailers to more efficiently understand the content and context of these images, thus allowing for the return of highly-personalized and responsive lists of related results.



These techniques are already being used by major retailers (eBay, ASOS, Neimann Marcus), tech giants (Google Lens), and social media companies (Pinterest Lens), and though these approaches are still (relatively speaking) in their infancy, the results are compelling:

- 55% of consumers say Visual Search is instrumental in developing their style and taste.
- The "Global Visual Search Market" is estimated to surpass $14.7 billion by 2023.
- When shopping online for clothing or furniture, more than 85% of respondents respectively put more weight on visual info than text info.

Image organization

With modern smartphone camera technology, it’s become incredibly easy and fast to snap countless photos and capture high-quality videos. However, with higher volumes of content, another challenge arises—creating smarter, more efficient ways to organize that content.

With ML-powered image recognition, photos and captured video can more easily and efficiently be organized into categories that can lead to better accessibility, improved search and discovery, seamless content sharing, and more.

Google Photos already employs this functionality, helping users organize photos by places, objects within those photos, people, and more—all without requiring any manual tagging.

Many of the current applications of automated image organization (including Google Photos and Facebook), also employ facial recognition, which is a specific task within the image recognition domain.

Content moderation

Many of the most dynamic social media and content sharing communities exist because of reliable and authentic streams of user-generated content (USG). But when a high volume of USG is a necessary component of a given platform or community, a particular challenge presents itself—verifying and moderating that content to ensure it adheres to platform/community standards.

To better exemplify the importance of this need for content moderation, let’s consider the example of One Bite, an online community that relies heavily on user-generated pizza reviews to power its network of more than 100,000 restaurants and establishments across the United States. Manually reviewing this volume of USG is unrealistic and would cause large bottlenecks of content queued for release.

To ensure that the content being submitted from users across the country actually contains reviews of pizza, the One Bite team turned to on-device image recognition to help automate the content moderation process. To submit a review, users must take and submit an accompanying photo of their pie. Any irregularities (or any images that don’t include a pizza) are then passed along for human review.

This kind of automated content moderation can be an essential tool in more effectively ensuring that community spaces are focused, safe, and fulfilling their intended purposes—all of which is made possible with the help of AI-powered image recognition.

Accessibility

One of the more promising applications of automated image recognition is in creating visual content that’s more accessible to individuals with visual impairments. Providing alternative sensory information (sound or touch, generally) is one way to create more accessible applications and experiences using image recognition.

Facebook was an early adopter of this technological advancement. In 2016, they introduced automatic alternative text to their mobile app, which uses deep learning-based image recognition to allow users with visual impairments to hear a list of items that may be shown in a given photo.

Similarly, apps like Aipoly and Seeing AI employ AI-powered image recognition tools that help users find common objects, translate text into speech, describe scenes, and more.

And because there's a need for real-time processing and usability in areas without reliable internet connections, these apps (and others like it) rely on on-device image recognition to create authentically accessible experiences.