

## CHAPTER 4

# Neural Networks and Image Search

*I envision some years from now that the majority of search queries will be answered without you actually asking. It'll just know this is something that you're going to want to see.*

—Ray Kurzweil, author, director of engineering, Google

It's hard to imagine an industry that relies on images more than the fashion industry. Almost every process, from manufacturing to marketing, revolves around images. This chapter discusses methods for classifying images, developments in neural networks that have been improving these methods, and the basics of how neural networks work. The idea of classifying images was mentioned in Chapter 3. It might not sound like a futuristic or exciting concept, but it is foundational for machines to answer the question, “What is this?” when working with an image.

When might you want to know what is in an image? In retail, the ability for customers to search for a particular garment style on a web site gives them access to the products they are searching for. Even better, the ability to discover products from styled images gives customers the ability to browse inspiration (what to do with a product or how to wear it) and to access the product (which item to buy to achieve the desired look).

## Fashion Industry Images

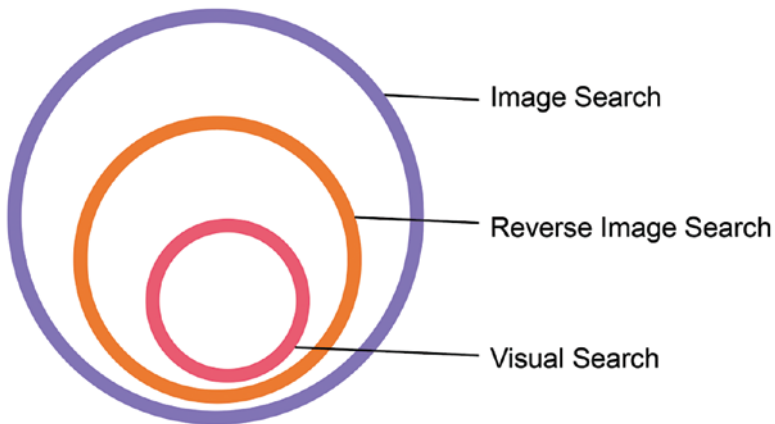
From marketing materials to design tools, images have a huge impact on operations in the fashion industry. Images are highly important in designing, constructing, and selling fashion. The ways images are used in the fashion industry include the following:

- Fashion photography
  - High-fashion images
  - Lookbooks
  - Editorial
- Web imagery
  - Product photography
  - Social media images
- Design drawings
  - Technical flats
  - Tech packs
  - Material references
  - Construction detail references

As you're reading this chapter, can you imagine how having a machine with the ability to identify the content in these images could be helpful?

## Image Search

The possibilities for using image search extend to the many aspects of the fashion industry that rely on images for information. Figure 4-1 shows three types of image-search techniques: image search, reverse image search, and visual search. With given text, a **search engine** is able to return images that have been tagged with matching and related **keywords**.



**Figure 4-1.** A chart of three types of image-searching techniques and their relationship to one another

**Image search** refers to the general topic of finding images, but usually refers to a search process based on a text input. The concept of image retrieval was introduced in the 90s, but wasn't popularized until image search was introduced by Google in 2001, after Jennifer Lopez's green Versace dress sent Internet users into an image search frenzy.

**Reverse image search** is a subset of image search referring to a search query in which an image is used to find another image. A further subset, **visual search**, refers to a process of finding items within an image and searching for those. For example, when searching for an image of a fashion blogger wearing a pair of black pumps, the search results will return the black pumps rather than returning more images that are visually similar to the image of the fashion blogger.

Image search is not a cutting-edge idea and doesn't necessarily include AI. However, in the fashion industry, even basic image search is hardly viable as a search tool. The way that images, from inspiration to process drawings, are organized is more often through folders on desktops rather than in any structured company-wide database.

## Image Tagging

To be able to search images from a text-based description requires **image tagging**. Image tagging is a manual process of using keywords to describe the content of an image. Using neural networks, it is possible to label or tag images with fewer manual processes.

Images on the Internet are displayed by using a text-based address to tell a computer where to find the image. The text can optionally include other descriptive information called **metadata**. The **alt text** is an optional element that is often used in metadata to describe the content of the image. Not all images include it, and the text can be anything. It's up to the programmer to make it a description of the image. The **HTML code** in Listing 4-1 shows the basic **markup**.

**Listing 4-1.** HTML Code That Machines Use to Render an Image on the Web

```

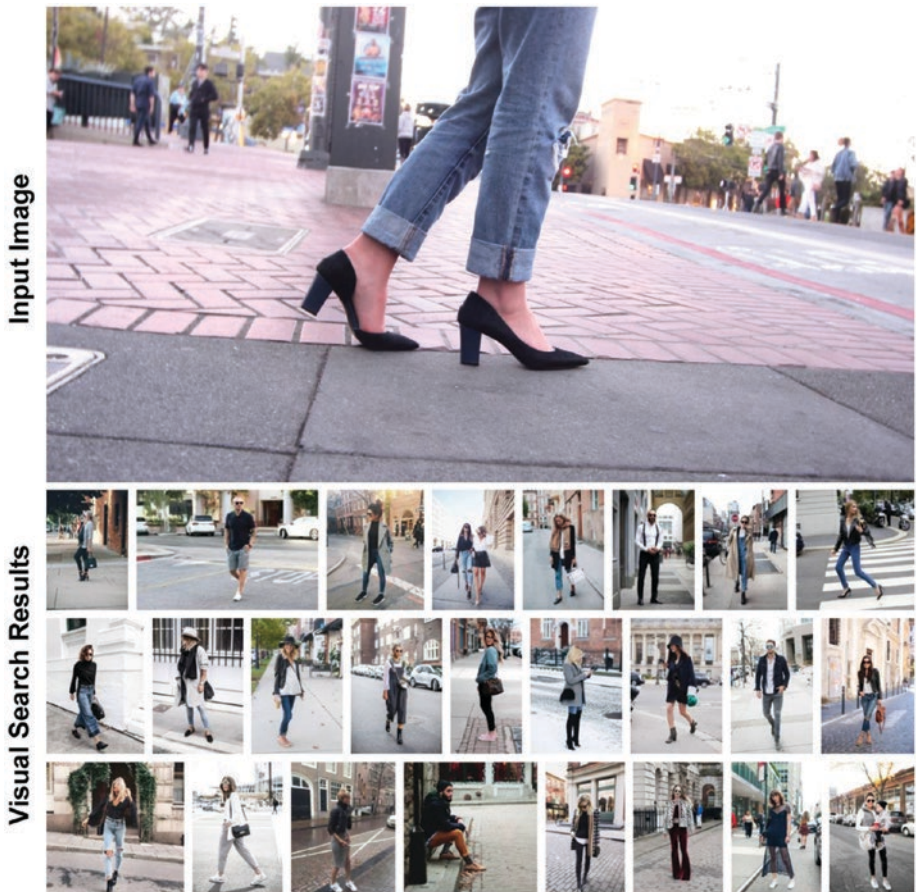
```

In the case of this HTML code, the machine finds the location of the image listed after `src`, short for *source*. Keywords listed in the alt text help machines recognize the image content. In this example, search engines could identify that the image is of the Statue of Liberty, located in New York City, based on the alt text that provides that information. Metadata can be used as the thing being searched, or images could be used as the thing being searched. The way the search process is undertaken may or may not use machine learning.

## Reverse Image Search

Reverse image search is an image-search method popularized by Google in 2011. In this method, an image is used as the search input. It is then analyzed; a search query is created, and results are given to the user.

The query is generated by a combination of factors including the image file name, link text, and text near the image. Figure 4-2 shows results from Google's reverse image search.



**Figure 4-2.** Reverse image search: an input image and visually similar results from Google

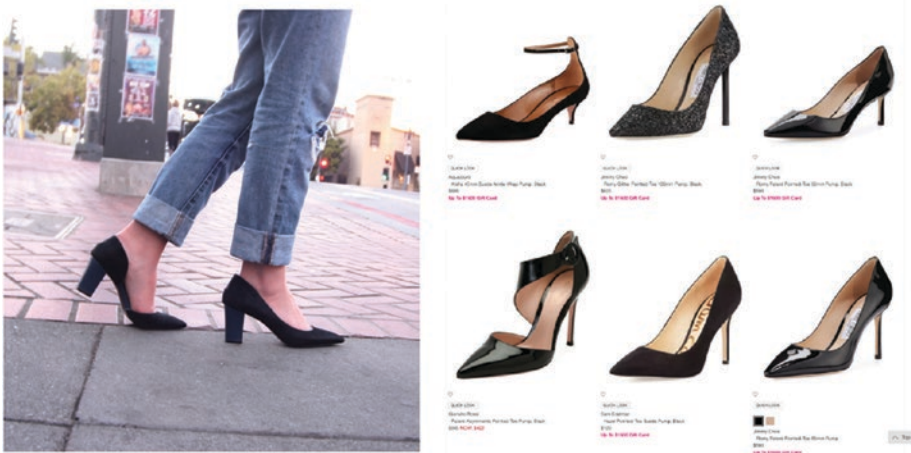
This search method might be used to track down the source of an image, find web sites that an image is posted on, get information about the image, find higher-resolution versions, or access images with similar

content. Before the introduction of this image-search method, screenshots from the Web would remain un-citable artifacts on user desktops.

Reverse image search also used computer vision algorithms for object recognition and to extract other visual information. Computer vision methods are discussed further in Chapter 3.

## Visual Search

Visual search, also implemented through the use of computer vision, provides us with the ability to search through troves of visual data without relying on text. Similar in concept to the challenges faced in natural language processing, the ability to search images that have not been tagged unlocks access to images that would otherwise not be found. Visual search takes an image as an input and returns similar images based on visual characteristics in the image. Figure 4-3 shows an example of visual search results.



**Figure 4-3.** Search results on Neiman Marcus using visual search by Slyce

While reverse image search is optimized for similar images, visual search can be optimized to search for similar items across images. There are a range of ways to create a visual search model.

Neural networks (NNs) are a mathematical or computational tool. Computer vision is a field that applies that tool to image data. The computer vision system is able to recognize objects in the image. The ability to identify that the person is walking and what kind of heels she is wearing is likely a task for machine learning. The key difference here is that computer vision gives the ability to see, while machine learning (namely, neural networks) gives the ability to recognize objects. Computer vision and machine learning can be used to solve problems independently, but more can be accomplished when the two are combined.

### BETTY & RUTH IMAGE SEARCH WORKFLOW

Image search can be useful in unexpected ways. At Betty & Ruth, we used to manage our internal images through a mess of screenshots and desktop folders. We had never thought about using more advanced image search to manage our internal images for design and marketing. Our biggest issue is recalling images (“Where is there a picture of that pink ruffle blouse from last season?”).

We started using online storage services, like Box. They have an image-recognition feature, which has made it easier to find things. It’s not perfect yet, but it helps a lot.

During our fitting sessions and design processes, we upload process images from our phones. The practice sounds basic, but like many other brands we were using old digital cameras before, and it was difficult to manage.

## Neural Networks

*Emotions are enmeshed in the neural networks of reason.*

—Antonio Damasio, neuroscientist

Neural networks are a type of machine learning model. Chapter 1 provides a brief overview of neural networks, which are modeled after early theories on the way the human brain works. Many methods can be used in a neural network, and nuances in data are being used to train these networks.

Neural networks use example data to infer rules for characterizing new data. The main idea behind neural networks is that you give the system many example answers, and then that collection is analyzed to infer patterns. This is an example of a supervised learning training process. Once the model has been trained, it can analyze new data and label it based on the inferences it made in the training process.

## Types of Neural Networks

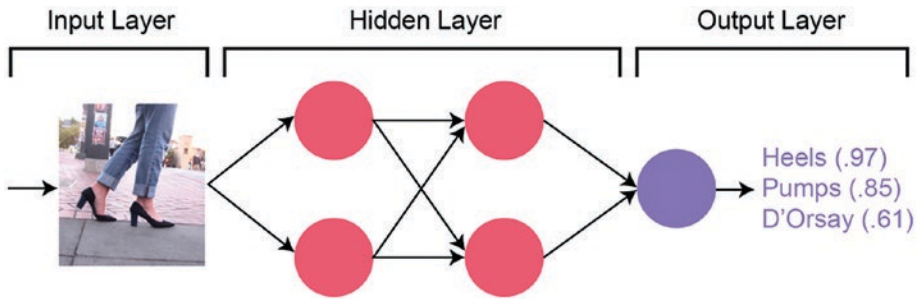
Neural networks are commonly used, but the architecture of the neural network can have a large impact on how effective they are for a given application. Feed-forward neural networks, recurrent neural networks, and convolutional neural networks each have their strengths. Feed-forward NNs are one of the simplest neural networks. Recurrent NNs are useful when the order of the data is important like in language-based applications. Convolutional NNs were inspired by the human visual system and are often used for image-based applications.

## Feed-Forward Neural Networks

A **feed-forward neural network** is the simplest form of neural network. In feed-forward neural networks, the data being passed through the network travels in only one direction. These algorithms take inputs and then generate outputs. Feed-forward NNs allow signals to travel in one direction, from input to output.

Mentioned in Chapter 1, neural networks are typically organized into three basic layers, though many extend well beyond three layers in practice. These basic layers include an input layer, hidden layer, and output layer, as shown in Figure 4-4.





**Figure 4-4.** *The flow of information, input, and output in a simplified illustration of a neural network. There isn't usually a single input node, but rather a collection of images or other types of data.*

## Input Layer

In the first layer, the input layer, no computation is performed. This is the part of the process in which information is passed into the hidden layer. In the example shown in Figure 4-4, the input is an image of a woman from the knees down wearing heels.

## Hidden Layer

The hidden layer is where computation is done. Each hidden layer is only a single layer, but there can be more than one hidden layer.

Each input value, in this case an image, is passed through each node in the hidden layer. Each input is given a **weigh**, or **bias**. Weights refer to the strength of the relationship between nodes in the hidden layer.

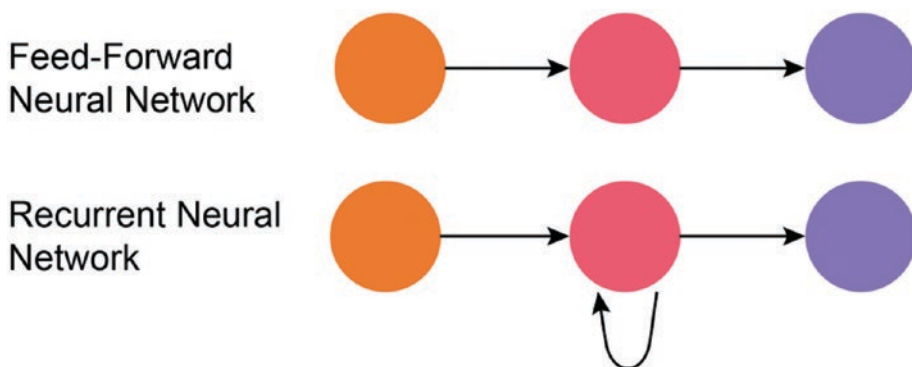
Neural network training is about figuring out what the weights should be. Each input used in the training process contributes to fine-tuning the weights between nodes. As the network is trained, the weights are adjusted based on the neural network's performance. The performance is evaluated by running the model on labeled data that was not used in training and seeing how well the neural network predicts the label.

## Output Layer

In the output layer, the **activation function**, also called the *transfer function*, is triggered. Each node in the output layer will return a yes (1) or a no (0) before continuing information to the next node. The activation function does a lot of math to interpret what happened inside the hidden layers and determine what to do with it. In an image classifier, the output might look like the example in Figure 4-4, which shows three possible categories and the probability that those categories are correct, given the image.

## Recurrent Neural Networks

**Recurrent neural networks (RNNs)** are particularly useful when it comes to **sequential data**. Arranging data in order is important for applications like natural language processing and speech recognition. Recurrent networks can have multiple hidden layers. While feed-forward neural networks can also have multiple layers, they allow signal to travel in only one direction, from input to output. Figure 4-5 shows a simple recurrent network cycling through the hidden layer, where the data is processed multiple times using the same function and parameters.

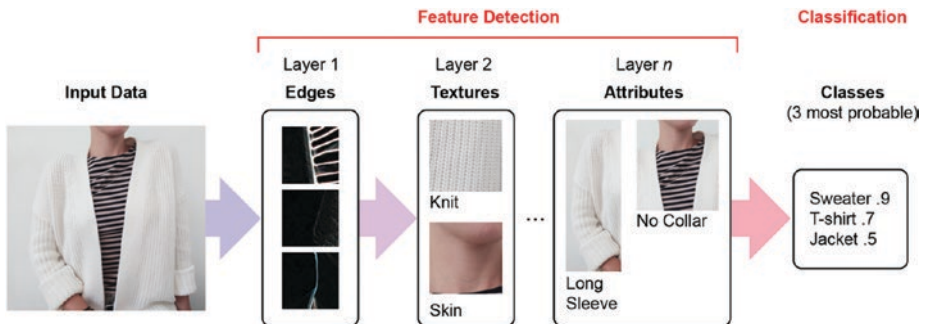


**Figure 4-5.** The basic difference between a feed-forward neural network structure and a recurrent neural network

## Convolutional Neural Networks

**Convolutional neural networks (CNNs)** are better suited for working with images. In CNNs, the neural network will find features in a large dataset and use those to determine what is in the image. The design of CNNs was inspired by the visual cortex system specifically for image-based tasks.

It can be difficult to understand what is going on in the hidden layers of any neural network, but in a CNN, there are two major parts: feature extraction and classification. Features of an image are being detected, narrowing down what is contained in that image. An image with a sweater might be classified by characteristics such as knit structure, the presence of skin, sleeves, and collar. A hypothetical, simplified example of this is shown in Figure 4-6.



**Figure 4-6.** A hypothetical, simplified example of how a CNN extracts features to classify an image of a garment

A CNN can have tens or hundreds of hidden layers. Each layer can detect different features within an image, increasing complexity with each layer, as in the image shown.

## Training Neural Networks

After a neural network is set up, it needs to be trained. **Training** is an important concept in machine learning and is broadly applied to machine learning models, not just neural networks. By sending training data through the model, it “learns” to generate results.

There are two main approaches to training neural networks: supervised and unsupervised learning.

### Supervised Learning

In **supervised learning**, the inputs and desired outputs are provided. For example, an image of a dress is used as an input, and the word “dress” is used as an output. (This example would be applicable in classifying large sets of images.) With the already labeled data, the network can process and then compare the results.

Wherever there is an error (for example, if the machine returned “skirt” instead of “dress”), the error is sent back through the system and used to improve the results. This process is called **backpropagation**. Backpropagation compares training results to the manually labeled results and feeds them back through the network to improve accuracy. The system can adjust the weights at each node accordingly to correct the error. Training is complete when changing the weights at each node no longer produces a better result.

This is a common strategy for training neural networks. Supervised training of neural networks relies heavily on the quality of the training data. The networks cannot learn without high-quality, accurately labeled data.

### Unsupervised Learning

Another strategy used in training is **unsupervised learning**. In this case, the network is not told the correct or desired output and must decide for itself which features to use to classify data and self-organize. This behavior

is commonly called **adaption**. The reason that unsupervised learning is a goal is because there is an ever-increasing amount of easily accessible unlabeled data, whereas creating labeled data can be a time-consuming and costly human task. However, it is a much more challenging approach.

We won't discuss unsupervised learning techniques in great detail in this book, but it's important to know that this is a field of study that has the potential to address a major pain point in machine learning: manually labeling huge datasets.

## Training Data

The training process requires large datasets in order for the neural network to find patterns across images. A number of datasets have been created and made publicly available in the research community. While in many industries, information is closed, machine learning has evolved so rapidly and effectively in part because of the sharing of valuable information and resources like these training libraries.

## Standardized Datasets

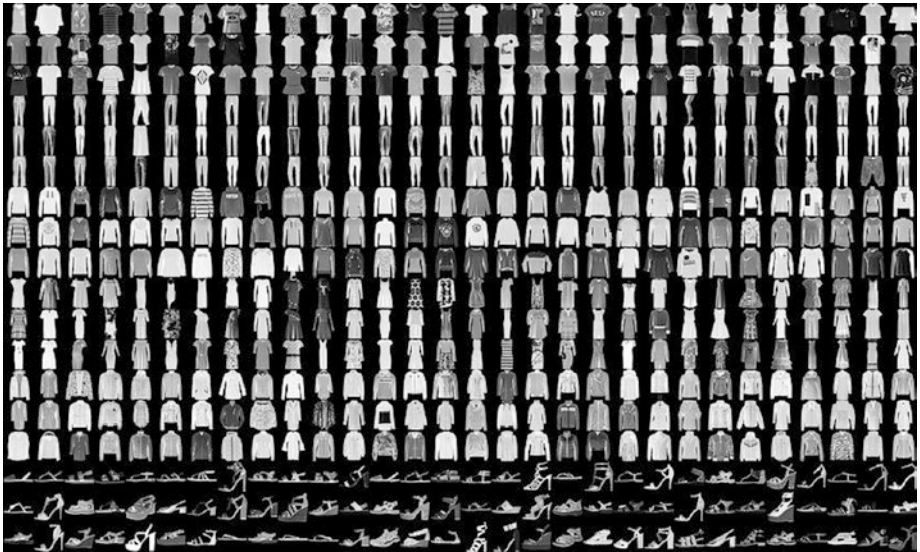
Using standardized datasets helps reduce the variables and isolate problems in designing neural networks and other models. However, standardized datasets introduce other challenges such as perpetuating bias across multiple systems.

One of the most commonly used training datasets is the **Modified National Institute of Standards and Technology (MNIST)** database. The MNIST database is a set of 60,000 images of handwritten characters, sampled in Figure 4-7.



*Figure 4-7. Handwritten examples from the MNIST dataset*

Relevantly, a more effective training dataset called **Fashion-MNIST** was introduced by Zalando in 2017. Zalando is a German-based e-commerce retailer that specializes in fashion and beauty products. The Fashion-MNIST dataset contains 60,000 garment images (instead of handwritten characters) as the training data and is said to be more representative of modern computer vision tasks. The dataset offers greater variance and complexity of images compared to the MNIST database. Figure 4-8 shows a sample of images from Fashion-MNIST.



**Figure 4-8.** A sample from the Fashion-MNIST database

Like the MNIST dataset, Fashion-MNIST contains ten categories of images. In this case, instead of numbers 0–9, they are T-Shirt/Top, Trouser, Pullover, Dress, Coat, Sandals, Shirt, Sneaker, Bag, and Ankle Boots. Rather than the simple features of numbers (lines, curves, and loops), the fashion dataset represents more-complex features (necklines, sleeves, and much more).

In this field, new datasets and tools are made available all the time, improving the work of researchers training new models and applying existing models.

With these improvements, obstacles are also sometimes introduced. Examples of the ways neural networks can be exploited are also sometimes revealed.

## Adversarial Examples

*In order to be irreplaceable, one must always be different.*

—Coco Chanel, fashion designer

While emerging technologies are exciting, they commonly require solving problems that don't exist elsewhere. With any new technology, it's just as important to address weaknesses and threats associated with the new technology. Machine learning does have a long history, but in many cases the research is still nascent, and security exploits are commonly found. While this is arguably true of most of computing, in certain applications, security vulnerabilities could pose a danger to humans.

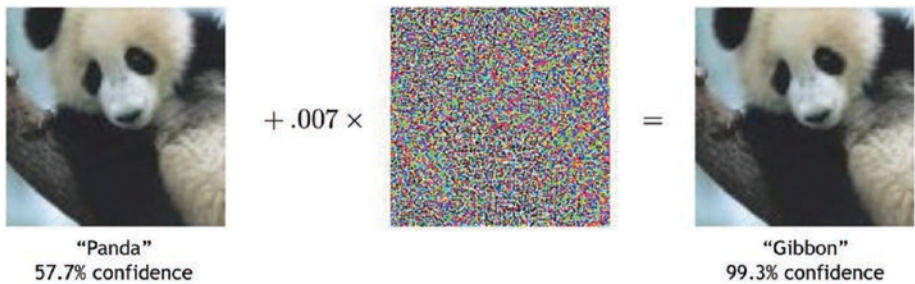
**Adversarial examples** are an example of possible security exploits in machine learning. An adversarial example is sort of like an optical illusion for computers that causes the computer to misinterpret something. These examples are intentionally created by an attacker to trick a machine into making a mistake.

Adversarial examples demonstrate that minor changes, which are imperceptible to humans, can dramatically change the results given by a machine. These examples can apply to numerous mediums, particularly images and three-dimensional objects.

## Adversarial Image Overlays

There are some well-known adversarial examples. In their paper, "Explaining and Harnessing Adversarial Examples," Ian Goodfellow et al. show an image of a panda that with a few minor changes imperceptible to humans is classified as a gibbon. This example is shown in Figure 4-9. The image in the middle is output by the adversarial example. When overlaid on the image of the panda, the resulting image to the right is incorrectly classified as a gibbon.





**Figure 4-9.** A small change that is imperceptible to humans can cause a machine learning model to confidently output the wrong label

## Adversarial Additions

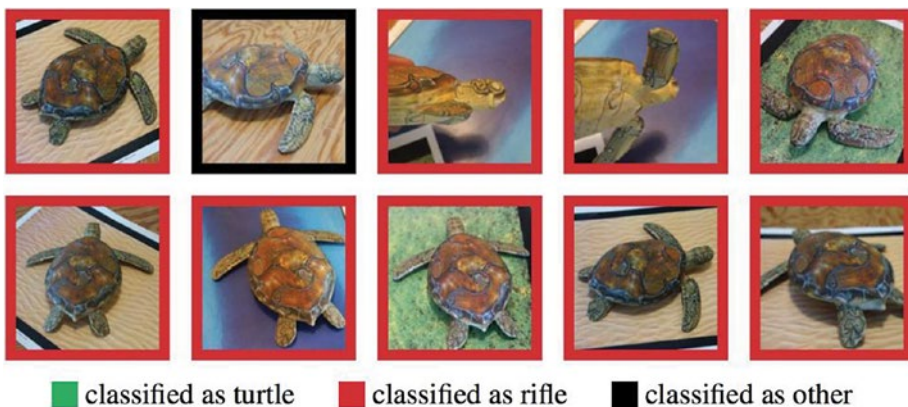
Tom Brown et al. in December 2017 published an example of being able to create a unique *adversarial patch* that, when present in an image, prompts the machine to ignore other objects in the image and classify the image as a toaster. This patch is able to trick a machine into believing that the image contains a toaster in a variety of environments. The use of this patch is shown in Figure 4-10.



**Figure 4-10.** The addition of an adversarial patch to a scene with a banana is able to trick the model into falsely labeling it as a toaster

## Adversarial Objects

As a final example, a 3D object can look like a turtle but be classified as a gun by a machine from every angle. While some of the examples shown using images are not always as robust under transformations like cropping and rotating, 3D adversarial examples are. Anish Athalye et al. demonstrated a reproducible method for creating these 3D examples in their paper, “Synthesizing Robust Adversarial Examples,” in October 2017. Their 3D-printed adversarial turtle can be seen in the images in Figure 4-11.



**Figure 4-11.** *The model classifies images of a turtle as a rifle with high confidence from almost every angle*

## Possible Implications

Is there a harmful way in which these exploits can be used in the fashion industry? We might not be able to imagine this possibility yet. We can only guess: what if someone created a neural network for spotting knock-offs? Another person could create images using these examples to reclassify fakes as real designer goods, or vice versa.

A common “what if” scenario that causes alarm in self-driving cars is the threat of stop sign that has been perturbed, or slightly modified to deceive the machine’s system. If a computer vision system in a self-driving car does not perceive a stop sign because it has an adversarial sticker placed on it, the car might not stop. These are the kinds of examples that give people pause when thinking about implementing new technologies on a massive scale.

## Summary

Images are used frequently in fashion. Historically, discovering these images has been based on text queries. New methods for finding images are emerging because of developments in neural networks. We can now understand much more about the content of images we use every day in fashion.

Neural networks offer a way to automate the process of understanding, “What is this?” in visual data. Convolutional neural networks are especially useful in this scenario. They were designed with the task of processing visual data in mind.

Adversarial examples show vulnerabilities in neural networks and how they can be exploited and manipulated to output the incorrect results. While for some applications this might be less problematic, in high-value or high-risk situations, they can go so far as to pose a threat to human life. It’s crucial to consider, what can go wrong if I make an incorrect guess?

## Terminology from This Chapter

**Activation function**—Defines the output of an individual node. In neural networks (NNs), this is also commonly referred to as the *transfer function*.

**Adaption**—A method in unsupervised learning in which the network decides for itself what features to use to classify data and self-organize.

**Adversarial examples**—Exploits that are able to confuse a neural network or other machine learning model, thereby getting that model to confidently output the wrong answer.

**Alt text**—Metadata often used in HTML markup to describe the contents of an image. It is often implemented by the engineer writing the code, and can be important for search engine discovery.

**Backpropagation**—Uses errors comparing training results to the correct results and feeds them back through the network. The model becomes more accurate by feeding learned information back through the system.

**Bias**—In neural networks, a bias is a number that represents the strength of the relationship between nodes in the hidden layer. Adjusting the bias, or weight, is a critical part of the training process. See also *weight*.

**Convolutional neural networks (CNNs)**—A type of neural network designed for interpreting visual data.

**Fashion MNIST**—The commonly-used successor to the MNIST dataset, Fashion MNIST is composed of 70,000 labeled fashion images.

**Feed-forward neural networks**—The simplest form of artificial neural networks, in which information flows in one direction only and never goes backward.

**HTML code**—A standard markup language used for creating web pages and applications to render UI components.

**Image search**—A more general term referring to the discovery of images based on a search query. Usually, image search refers to a text-based query, and other terms, like reverse image search or visual search, are used to describe image-based methods.

**Image tagging**— A process of describing what is in an image by using keywords.

**Keywords**—Also referred to as *index terms*, these capture the essence of an image or document and make it searchable, particularly on the Internet.

**Markup**—In computer processing, a system for annotating text that modifies how the text is displayed. HTML is a commonly used markup

language. A markup language is used to indicate sections of a text document (including headings), images, and styling differences.

**Metadata**—Data that refers to other pieces of data. Metadata could include information like author, title, description, and location. It may be hidden from the user, but machine readable for discovery.

**Modified National Institute of Standards and Technology (MNIST)**—The MNIST dataset is commonly used for training and testing image-based machine learning models. It is a databased composed of 70,000 hand-written characters, numbers from 0-9.

**Recurrent neural networks (RNNs)**—Useful when arranging data in order is important; that is, for natural language processing and speech recognition.

**Reverse image search**—A search method that uses a user input image to find similar images.

**Search engines**—Find items like documents or images that are related to a user's input by using keywords. Search engines are used both locally and on the Internet.

**Sequential data**—Data that requires a special order to make sense. A sentence is an example of information that, taken out of sequence, can lose all meaning.

**Supervised learning**—A method of learning in which a model is trained on labeled training data.

**Training**—The process in which a model learns from training data.

**Unsupervised learning**—Uses unlabeled data to train neural networks and other machine learning models.

**Visual search**—A search tool for finding objects within an image that are returning results similar to a given object. This process differs from reverse image search in that the results are not related to the image as a whole, but rather related to the objects it contains.

**Weight**—In neural networks, a weight, or bias, refers to the strength of the relationship between nodes in the hidden layer. See also *bias*.