Case Study 3 - Spill Detection from Video

# Step One: Defining the Problem

Imagine you run a company that offers specialized on-site janitorial services. A client, an industrial chemical plant, requires a fast response for spills and other health hazards. You realize if you could *automatically* detect spills using the plant's surveillance system, you could mobilize your janitorial team faster.

*Machine learning could be a valuable tool to solve this problem.*

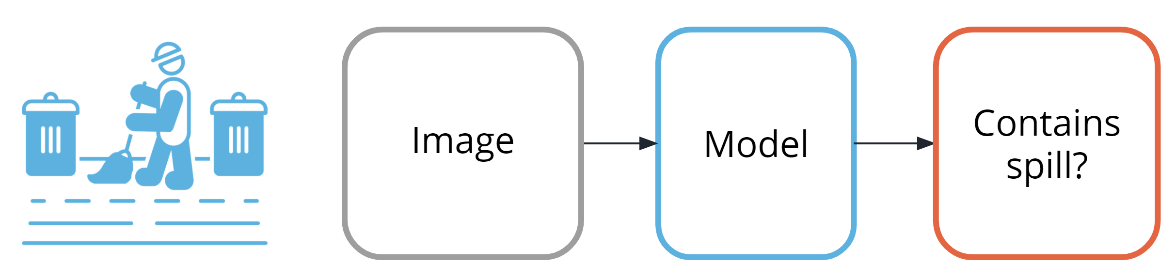


Figure 1: Detecting spills with machine learning

# Step Two: Model Training (and selection)

This task is a supervised classification task, as shown in the following image. As shown in the image above, your goal will be to predict if each image belongs to one of the following classes:

* Contains spill
* Does not contain spill

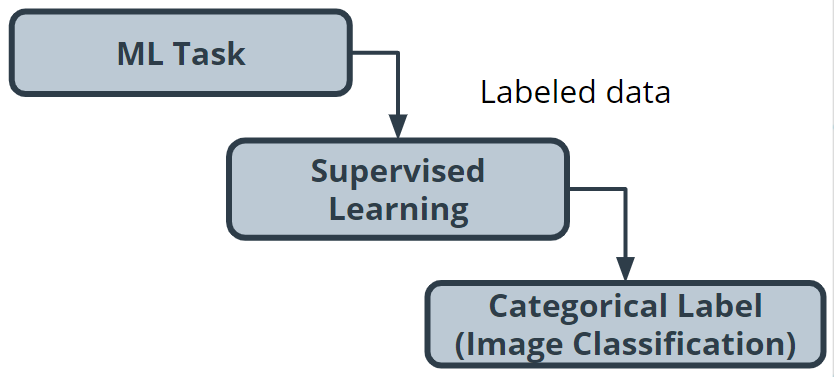


Figure 2: Image classification

# Step Three: Building a Dataset

* Collecting - Using historical data, as well as safely staged spills, you quickly build a collection of images that contain both spills and non-spills in multiple lighting conditions and environments.
* Exploring and cleaning - You go through all the photos to ensure the spill is clearly in the shot. There are Python tools and other techniques available to improve image quality, which you can use later if you determine a need to iterate.
* Data vectorization (converting to numbers) - Many models require numerical data, so all your image data needs to be transformed into a numerical format. Python tools can help you do this automatically.

In the following image, you can see how each pixel in the image on the left can be represented in the image on the right by a number between 0 and 1, with 0 being completely black and 1 being completely white.

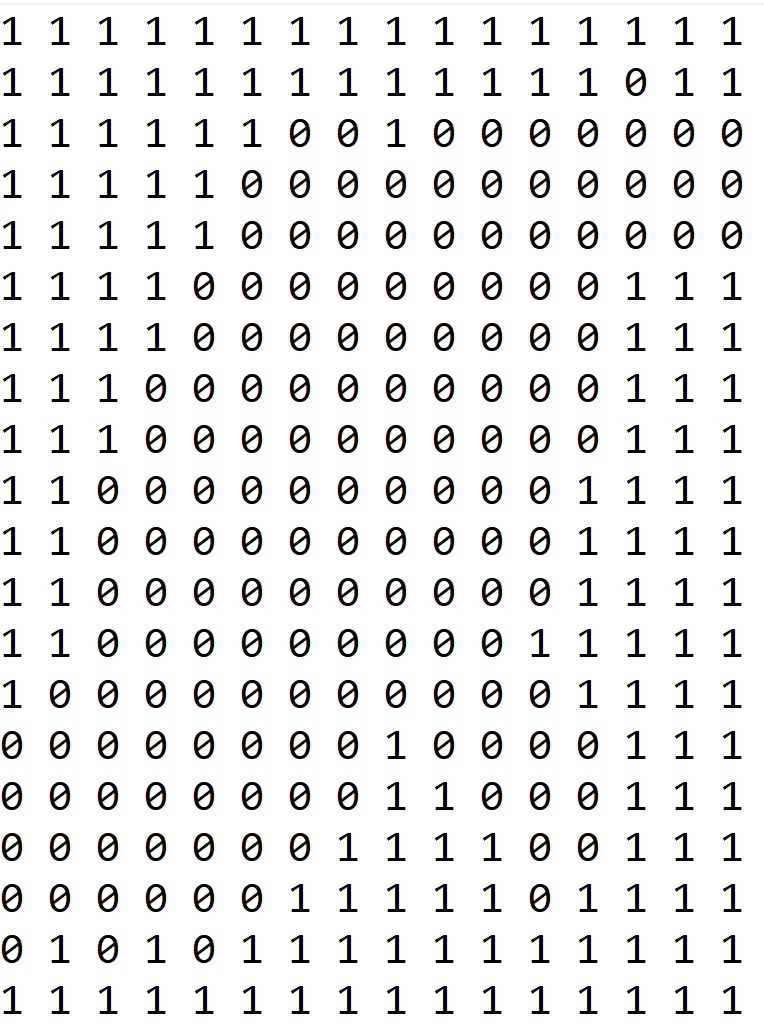


Figure 3: Numeric representation of chemical spill image

Figure 4: Chemical spill image

|  |
| --- |
|  |

* Split the data - You split your image data into a training dataset and a test dataset.

# Step Four: Model Training

Traditionally, solving this problem would require hand-engineering features on top of the underlying pixels (for example, locations of prominent edges and corners in the image), and then training a model on these features.

Today, deep neural networks are the most common tool used for solving this kind of problem. Many deep neural network models are structured to learn the features on top of the underlying pixels so you don’t have to learn them. You’ll have a chance to take a deeper look at this in the next lesson, so we’ll keep things high-level for now.

#### CNN (convolutional neural network)

Neural networks are beyond the scope of this lesson, but you can think of them as a collection of very simple models connected together. These simple models are called *neurons*, and the connections between these models are trainable model parameters called *weights.*

Convolutional neural networks are a special type of neural network particularly good at processing images.

# Step Five: Model Evaluation

As you saw in the last example, there are many different statistical metrics you can use to evaluate your model. As you gain more experience in machine learning, you will learn how to research which metrics can help you evaluate your model most effectively. Here's a list of common metrics:

|  |  |  |
| --- | --- | --- |
| Accuracy | False positive rate | Precision |
| Confusion matrix | False negative rate | Recall |
| F1 Score | Log Loss | ROC curve |
|  | Negative predictive value | Specificity |

In cases such as this, accuracy might not be the best evaluation mechanism.

***Why not?*** You realize the model will see the '**Does not contain spill'** class almost all the time, so any model that just predicts “**no spill**” most of the time will seem pretty accurate.

What you really care about is an evaluation tool that rarely misses a real spill.

After doing some internet sleuthing, you realize this is a common problem and that Precision and Recall will be effective. You can think of *precision* as answering the question, "Of all predictions of a spill, how many were right?" and *recall* as answering the question, "Of all actual spills, how many did we detect?"

Manual evaluation plays an important role. You are unsure if your staged spills are sufficiently realistic compared to actual spills. To get a better sense how well your model performs with actual spills, you find additional examples from historical records. This allows you to confirm that your model is performing satisfactorily.

# Step Six: Model Inference

The model can be deployed on a system that enables you to run machine learning workloads such as AWS Panorama.

Thankfully, most of the time, the results will be from the class '**Does not contain spill.'**

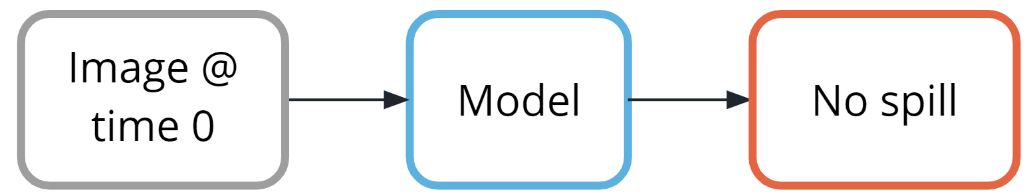


Figure 5: No spill detected

But, when the class '**Contains** **spill'** is detected, a simple paging system could alert the team to respond.

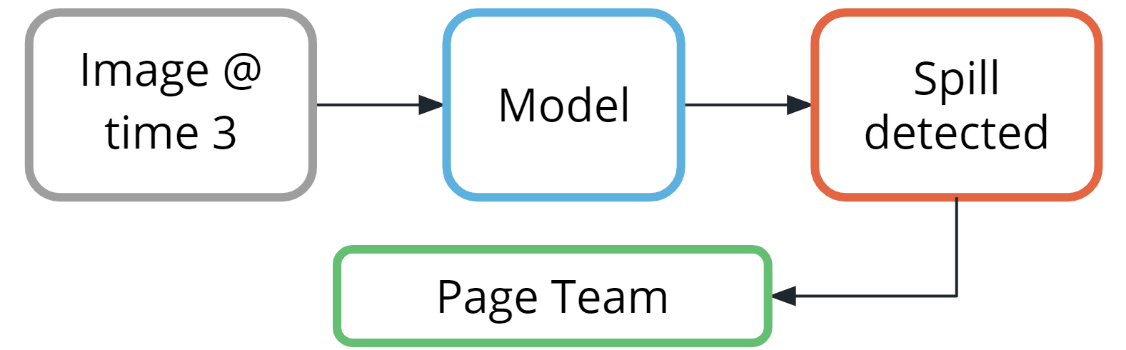


Figure 6: Spill detected