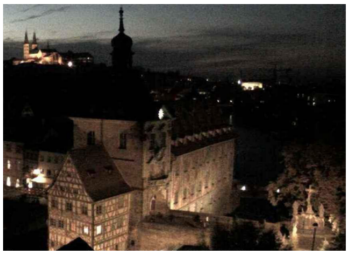
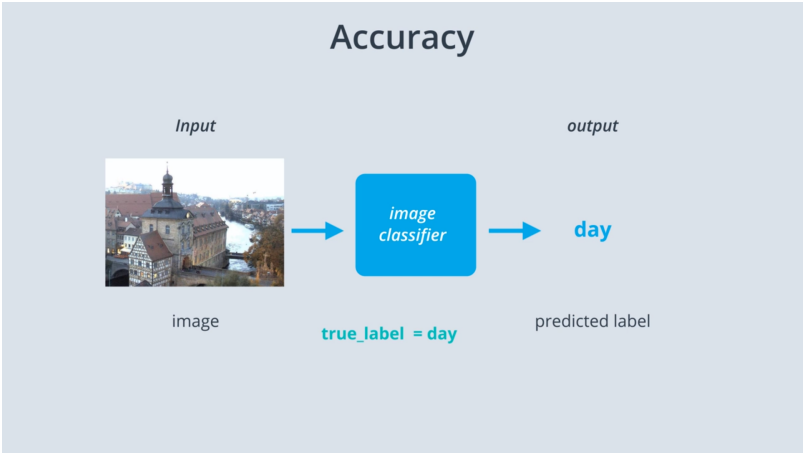
Day and Night Classification

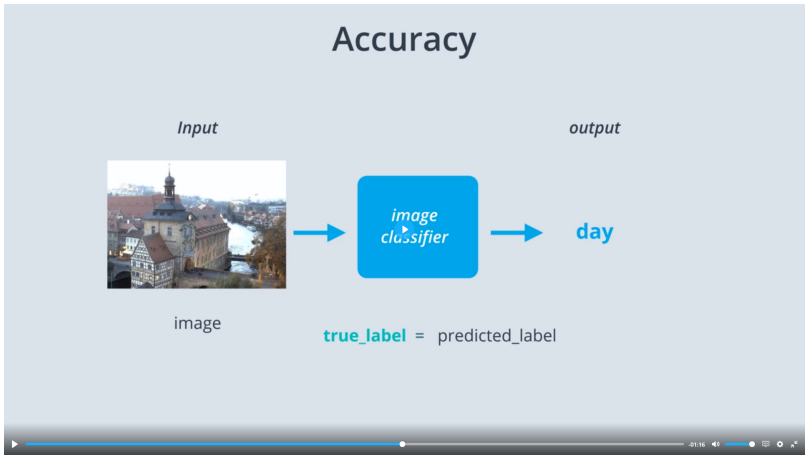
day

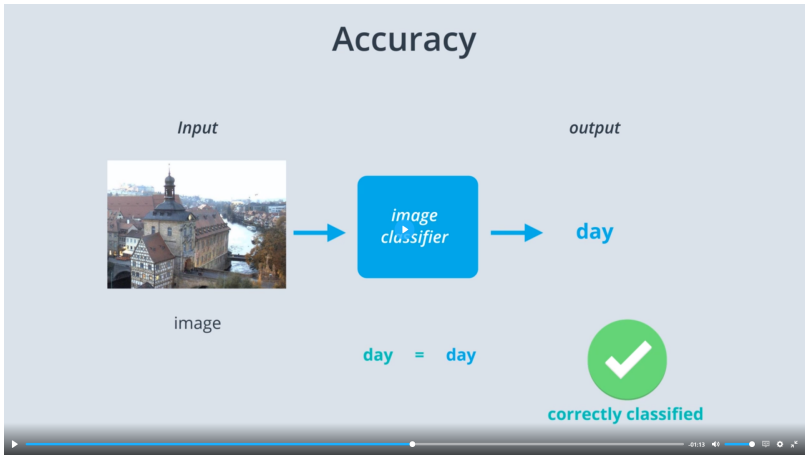


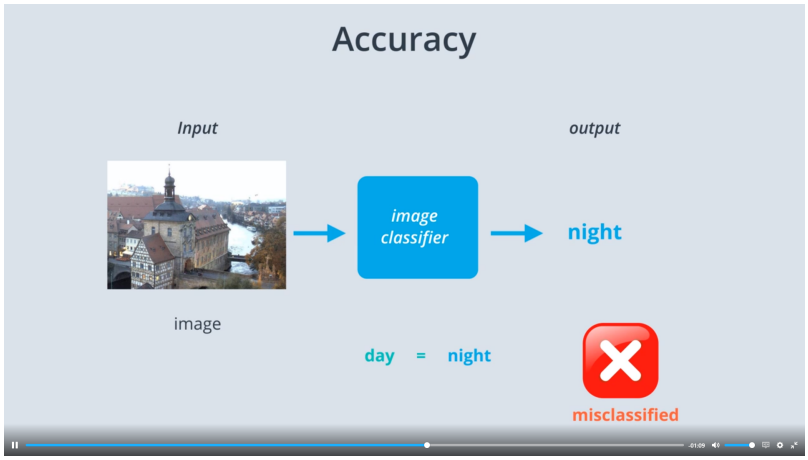
night

- **Visualizing the image data you're working with is the *first step in identifying any patterns in image data and being able to make predictions about the data!***
- label data and accuracy
  - *You* can tell if an image is night or day, but a computer cannot unless we tell it explicitly with a label!
  - This becomes especially important when we are testing the accuracy of a classification model.
  - A classifier takes in an image as input and should output a predicted_label that tells us the predicted class of that image. Now, when we load in data, like you've seen, we load in what are called the true_labels which are the *correct* labels for the image.
  - To check the accuracy of a classification model, we compare the predicted and true labels. If the true and predicted labels match, then we've classified the image correctly! Sometimes the labels do not match, which means we've misclassified an image.









- Accuracy = Total number of correctly classified images / Total number of images
- Numerical labels:
  - t’s good practice to use numerical labels instead of strings or categorical labels. They're easier to track and compare. So, for our day and night, binary class example, instead of "day" and "night" labels we’ll use the numerical labels: 0 for night and 1 for day.
  - Okay, now you’re familiar with the day and night image data AND you know what a label is and why we use them; you’re ready for the next steps. We’ll be building a classification pipeline from start to end!
  - Let’s first brainstorm what steps we’ll take to classify these images.

- FEATURES:
    - There are lots of measurable traits that distinguish these images, and these measurable traits are referred to as **features**.
    - A feature a measurable component of an image or object that is, ideally, unique and recognizable under varying conditions - like under varying light or camera angle.
    - Standardizing Output:
        - Categorical values are typically text values that represent various traits about an image. A couple examples are:
        - An "animal" variable with the values: "cat," "tiger," "hippopotamus," and "dog."
        - A "color" variable with the values: "red," "green," and "blue."
        - . Many machine learning algorithms do not; they require that all output be numerical. Numbers are easily compared and stored in memory, and for this reason, we often have to convert categorical values into **numerical labels**. There are two main approaches that you'll come across:
        - Integer encoding
            - Integer encoding means to assign each category value an integer value. So, day = 1 and night = 0. This is a nice way to separate binary data, and it's what we'll do for our day and night images.
        - One hot-encoding
            - One-hot encoding is often used when there are more than 2 values to separate. A one-hot label is a 1D list that's the length of the number of classes. Say we are looking at the animal variable with the values: "cat," "tiger," "hippopotamus," and "dog." There are 4 classes in this category and so our one-hot labels will be a list of length four. The list will be all 0's and one 1; the 1 indicates which class a certain image is.