Computer Vision

Lecture 8: Computer vision pipeline

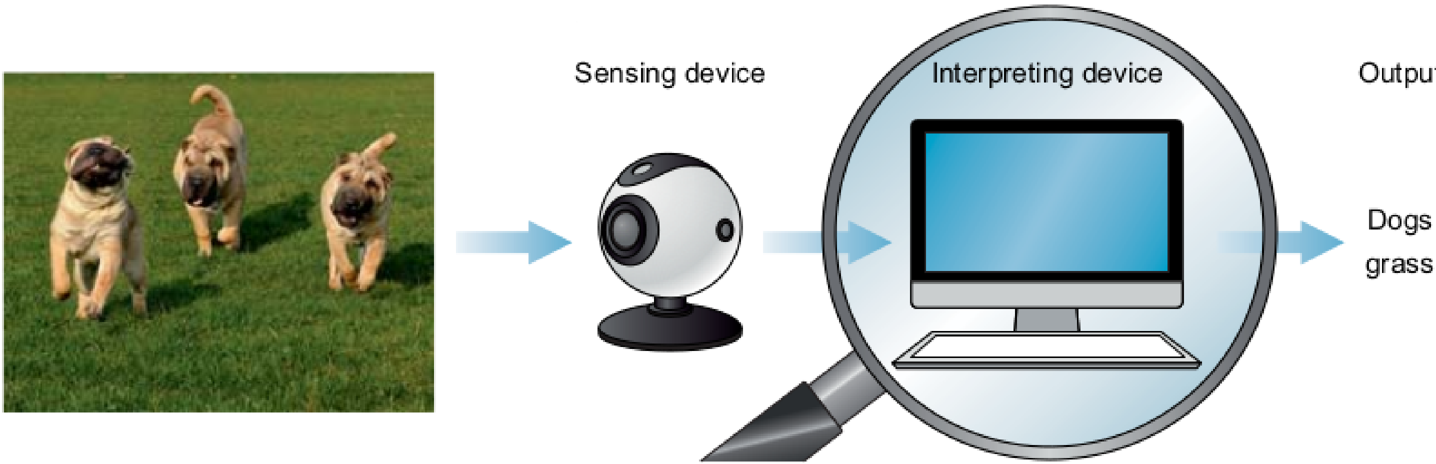
Yuriy Kochura
iuriy.kochura@gmail.com
@y_kochura

Today

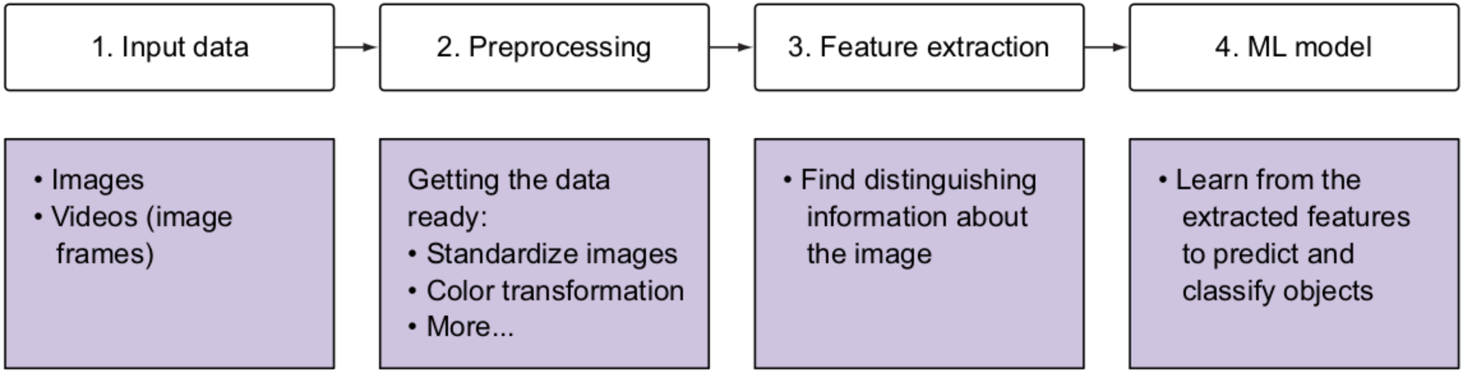
Sequence of distinct steps to process and analyze image data

- Input data
- Preprocessing
- Feature extraction

Computer vision system



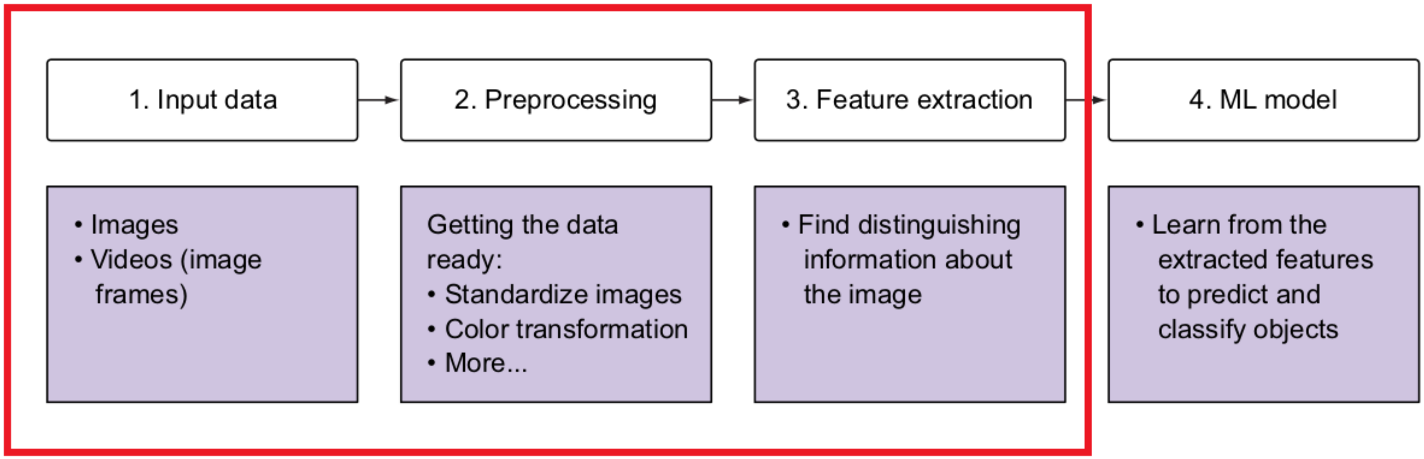
The computer vision pipeline



Credits: Mohamed Elgendy. Deep Learning for Vision Systems, 2020.

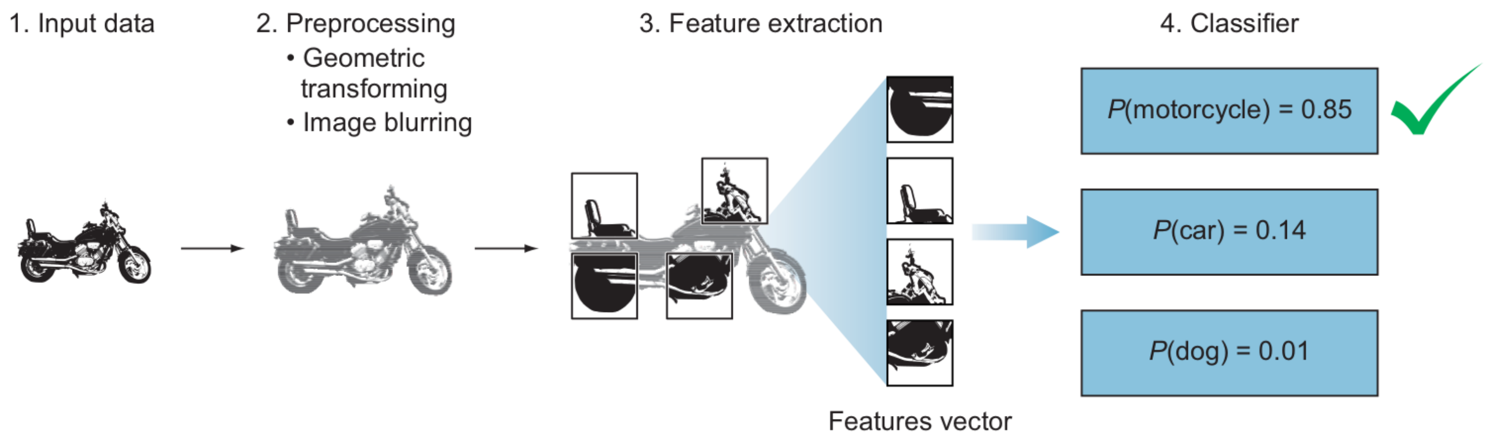
The computer vision pipeline

Today



Credits: Mohamed Elgendy. Deep Learning for Vision Systems, 2020.

The computer vision pipeline



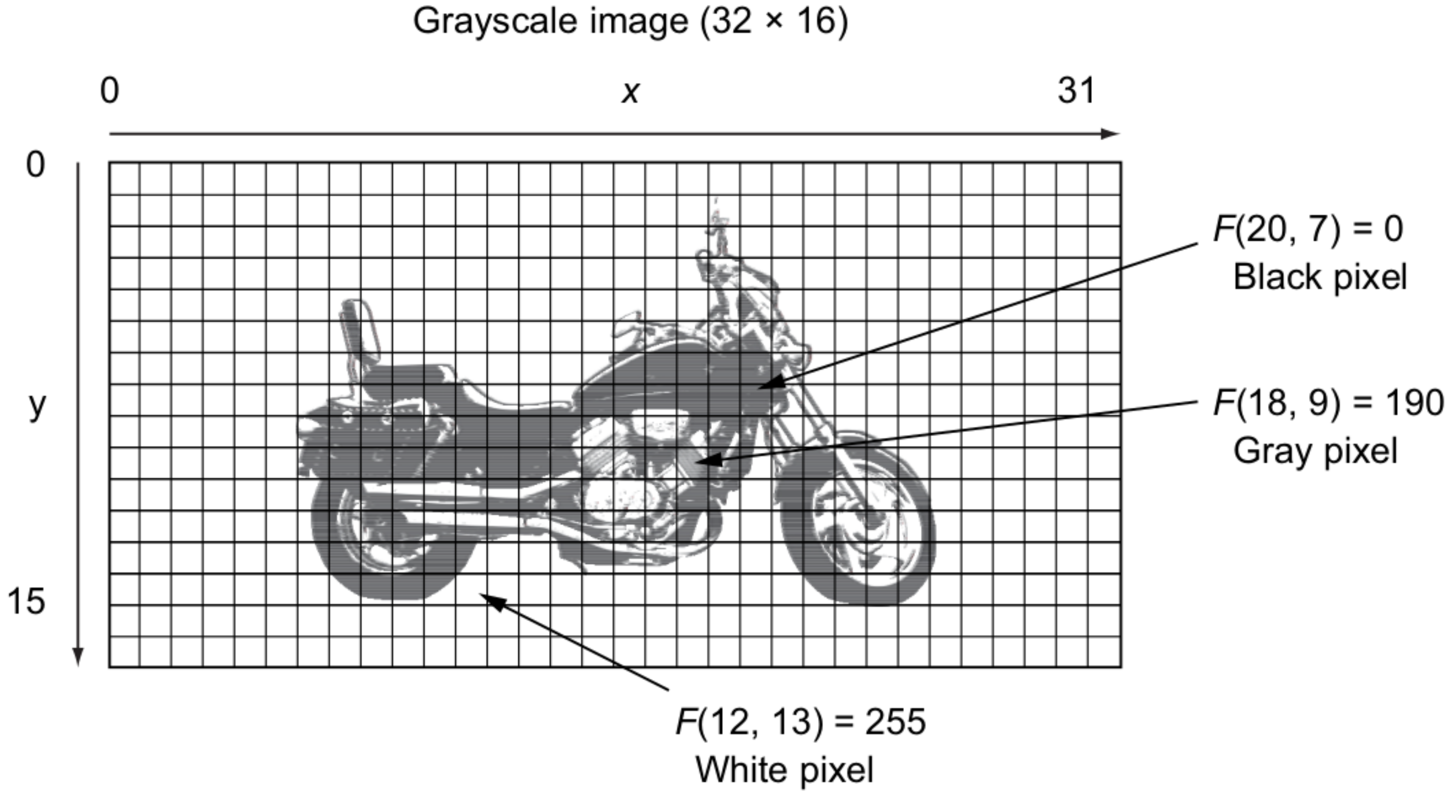
Definition. An image classifier is an algorithm that takes in an image as input and outputs a label or "class" that identifies that image. A class (also called a category) in machine learning is the output category of your data.

—

Credits: Mohamed Elgendy. Deep Learning for Vision Systems, 2020.

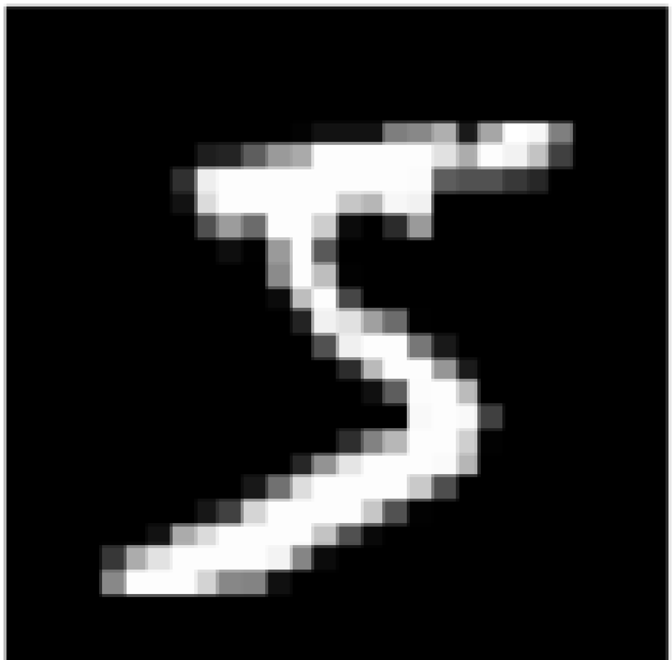
Input data

Image as functions

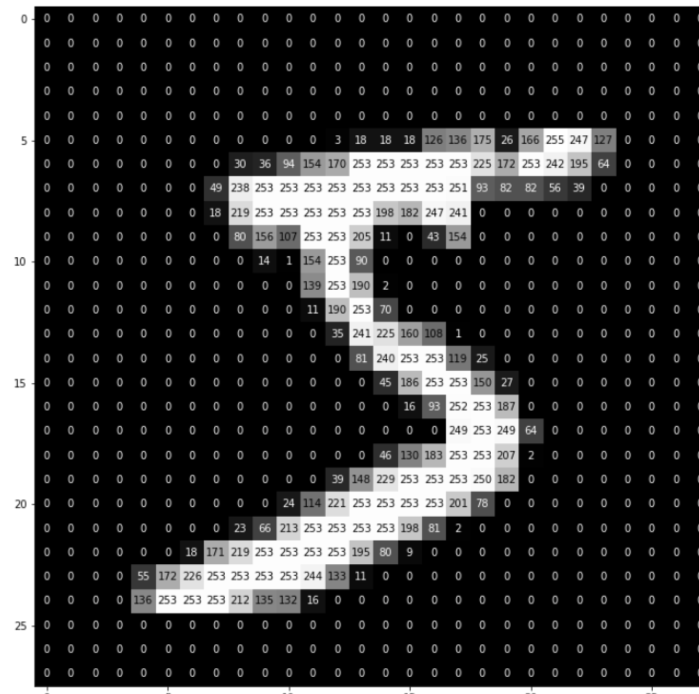


How do computers see images?

Example 1



What we see



What computers see

—

Credits: Mohamed Elgendy. Deep Learning for Vision Systems, 2020.

How do computers see images?

Example 2

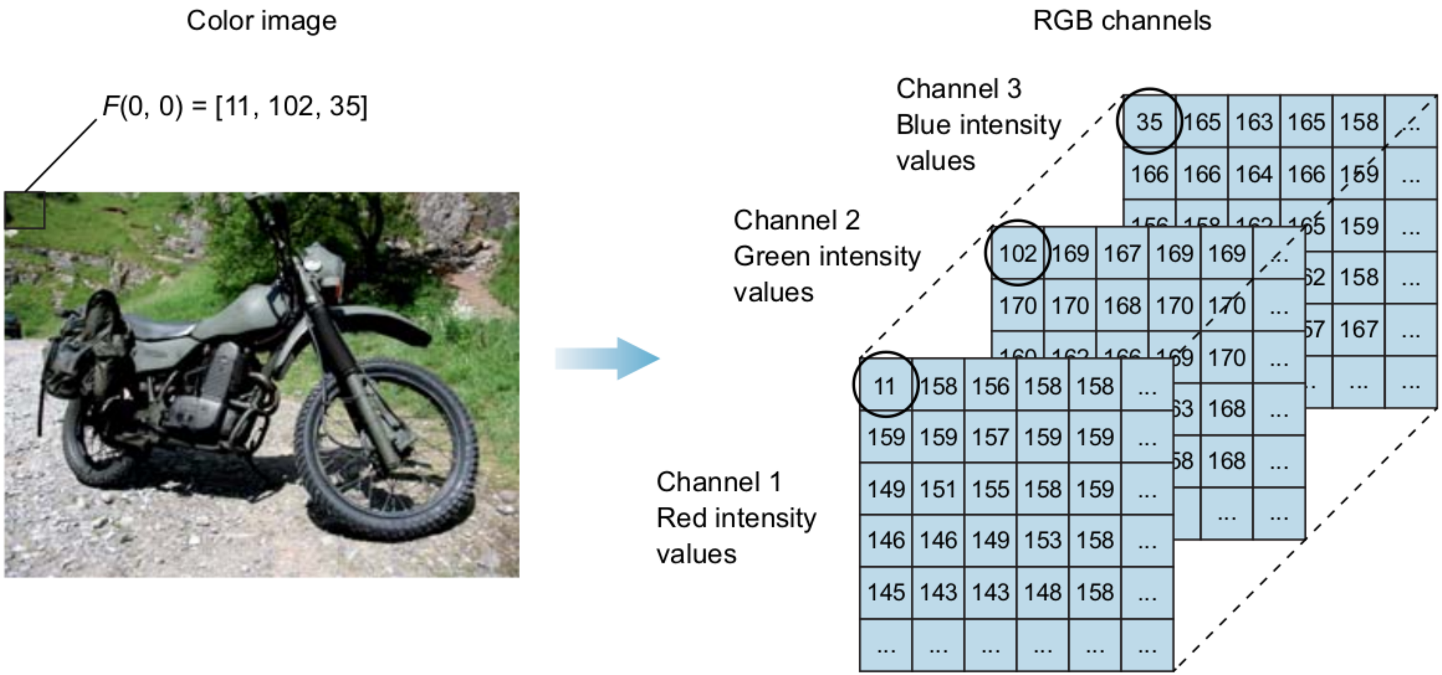
What we see

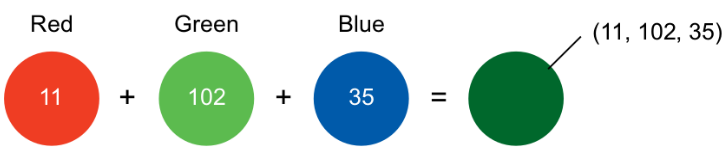
What computers see

08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	91	08
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	46	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	99	30	03	49	13	36	65
52	90	95	23	04	60	11	42	69	24	48	56	01	32	54	71	37	02	34	91
22	31	14	71	51	67	43	59	41	92	34	54	22	40	40	28	44	33	13	80
24	47	32	60	99	03	45	02	44	75	33	53	78	36	64	20	35	09	12	80
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
47	24	20	68	02	62	12	20	95	63	94	39	63	04	49	91	44	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	94	83	14	88	34	89	63	72
21	36	23	09	75	00	74	44	20	45	35	14	00	41	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	42	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	34	29	85	57
84	56	00	48	35	71	89	07	05	44	44	37	44	40	21	58	51	54	17	58
19	80	61	68	05	94	47	49	28	73	92	13	86	52	17	77	04	09	55	40
04	52	08	83	97	35	99	14	07	97	57	32	16	26	26	79	33	27	98	44
04	36	68	81	57	62	20	72	03	16	33	67	46	55	12	32	43	93	53	69
04	42	14	73	38	25	39	11	24	94	72	18	06	46	29	32	40	62	74	36
20	49	34	41	72	30	23	88	34	62	99	69	82	47	59	85	74	04	34	24
20	23	35	29	78	31	90	01	74	31	49	71	48	86	81	14	23	57	05	54
01	70	54	71	83	51	54	49	16	92	33	48	61	43	51	01	89	19	67	48

—
Credits: Mohamed Elgendy. Deep Learning for Vision Systems, 2020.

Color images





Summary

How do computers see color?

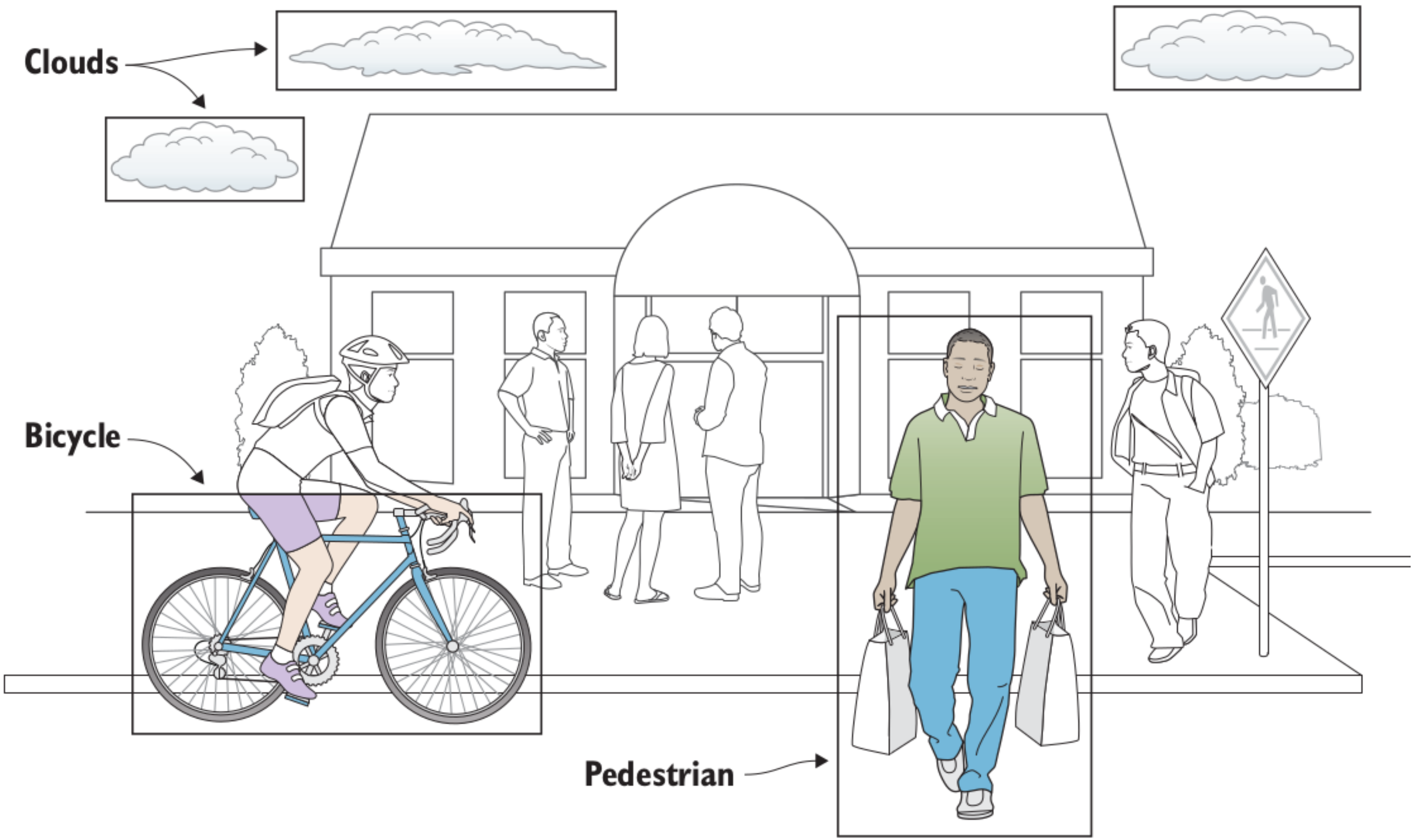
Computers see an image as matrices. Grayscale images have one channel (gray); thus, we can represent grayscale images in a 2D matrix, where each element represents the intensity of brightness in that particular pixel. Remember, 0 means black and 255 means white. Grayscale images have one channel, whereas color images have three channels: red, green, and blue. We can represent color images in a 3D matrix where the depth is three.

We've also seen how images can be treated as functions of space. This concept allows us to operate on images mathematically and change or extract information from them. Treating images as functions is the basis of many image-processing techniques, such as converting color to grayscale or scaling an image. Each of these steps is just operating mathematical equations to transform an image pixel by pixel.

- Grayscale: $f(x, y)$ gives the intensity at position (x, y)
- Color image: $f(x, y) = [\ \text{red}\ (x, y), \text{green}\ (x, y), \text{blue}\ (x, y)\ ]$

Preprocessing

Image preprocessing: Converting color images to grayscale



When is color important?

Converting an image to grayscale might not be a good decision for some problems. There are a number of applications for which color is very important: for example, building a diagnostic system to identify red skin rashes in medical images. This application relies heavily on the intensity of the red color in the skin. Removing colors from the image will make it harder to solve this problem. In general, color images provide very helpful information in many medical applications.

Another example of the importance of color in images is lane-detection applications in a self-driving car, where the car has to identify the difference between yellow and white lines, because they are treated differently. Grayscale images do not provide enough information to distinguish between the yellow and white lines.

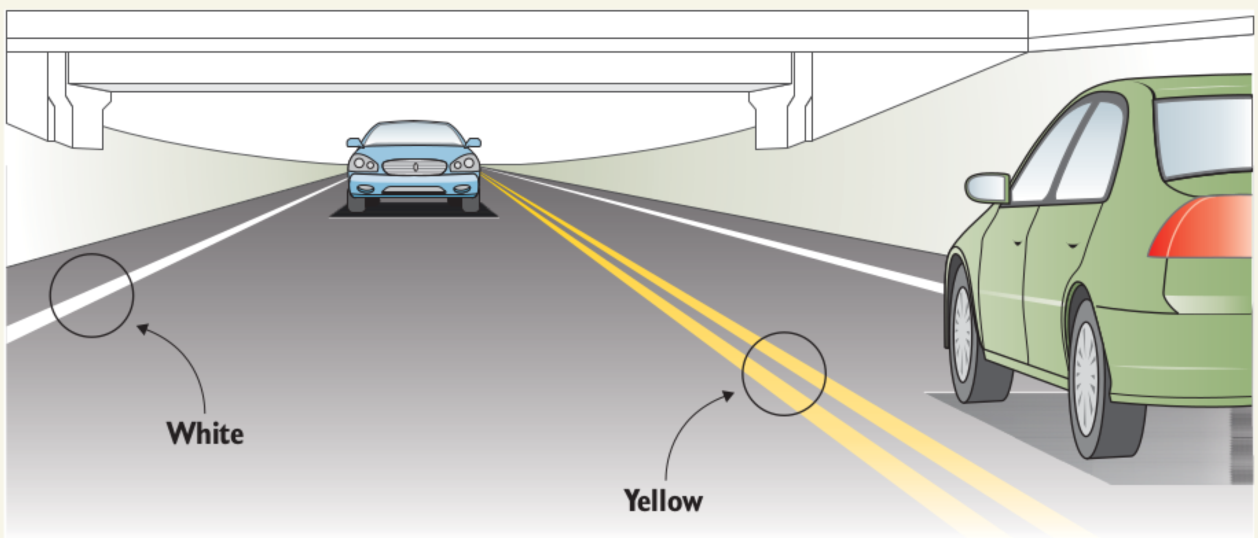
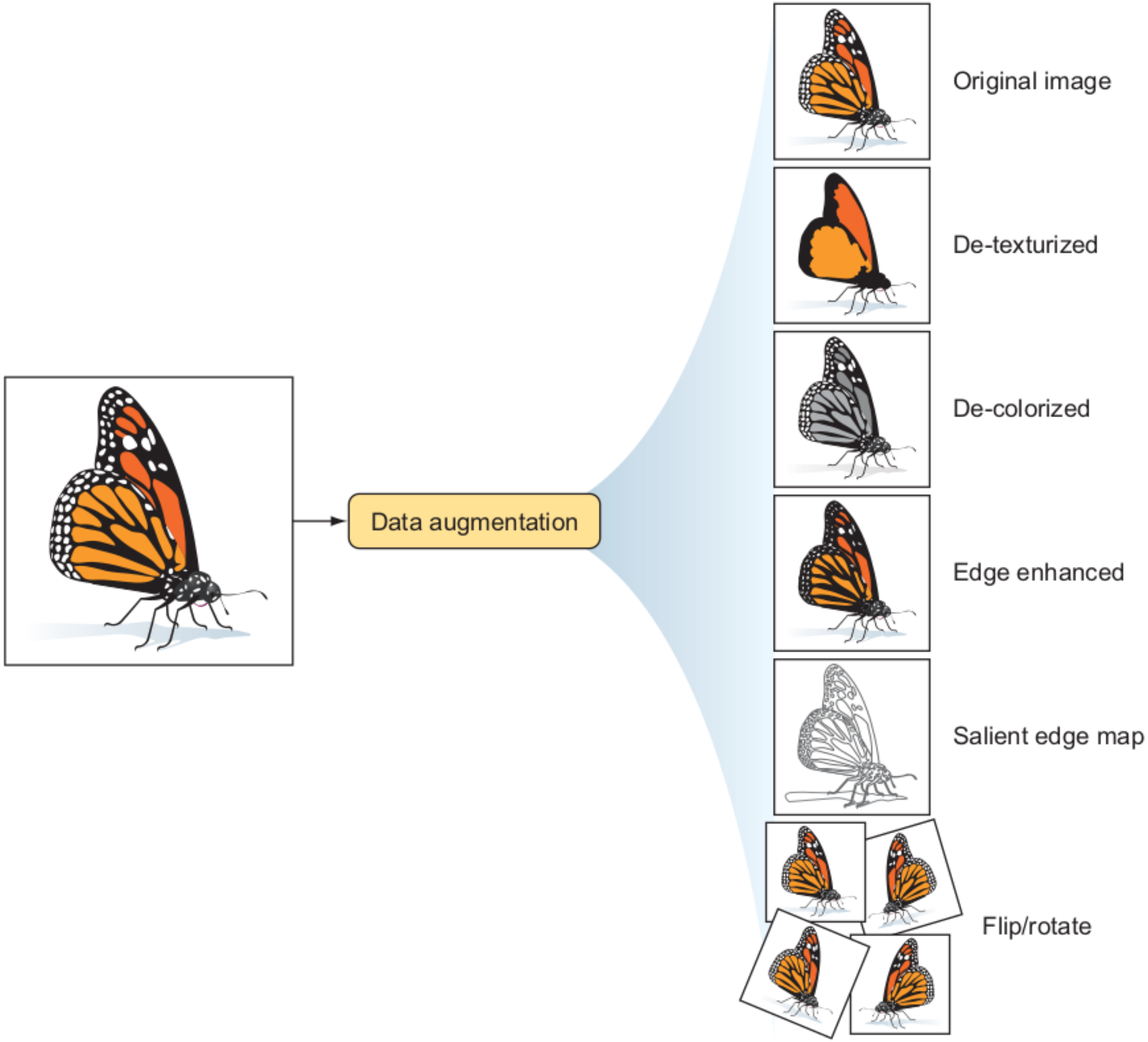


Image preprocessing: Standardizing images

One important constraint that exists in some ML algorithms, such as CNNs, is the need to resize the images in your dataset to unified dimensions. This implies that your images must be preprocessed and scaled to have identical widths and heights before being fed to the learning algorithm.

Credits: Mohamed Elgendy. Deep Learning for Vision Systems, 2020.

Image preprocessing: Data augmentation



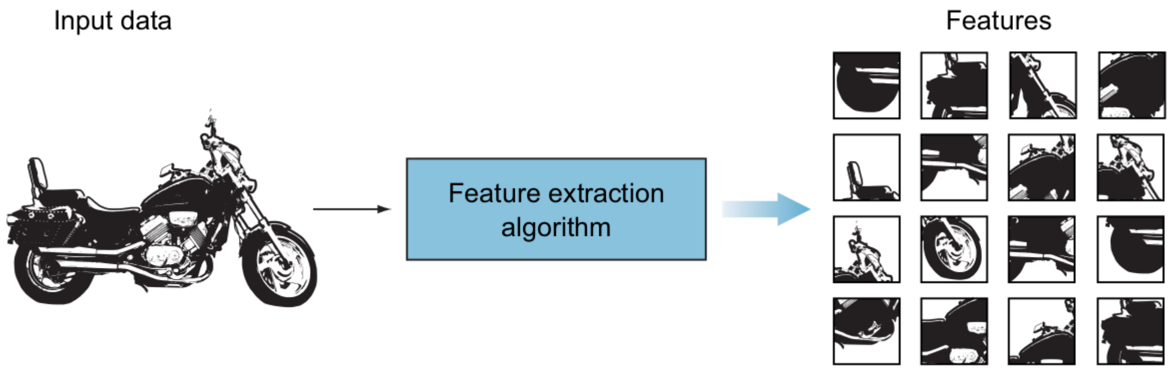
—
Credits: Mohamed Elgendy. Deep Learning for Vision Systems, 2020.

Feature extraction

Definition. A feature in machine learning is an individual measurable property or characteristic of an observed phenomenon. Features are the input that you feed to your ML model to output a prediction or classification.

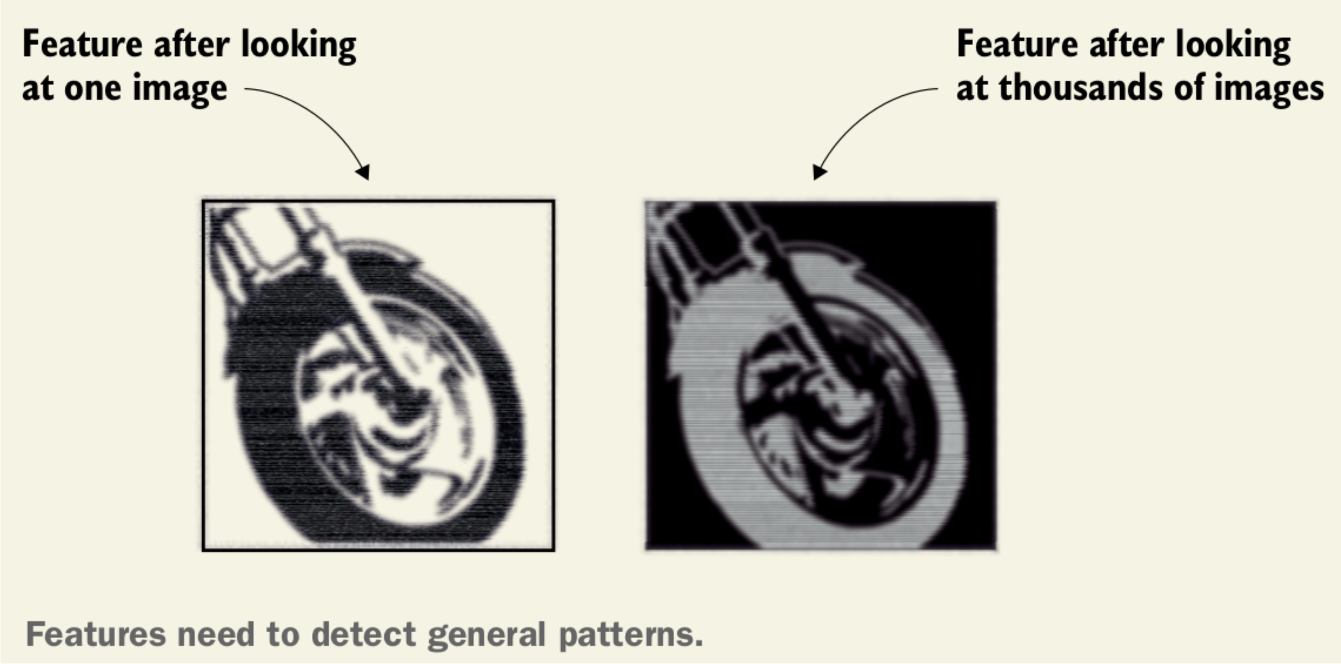
Suppose you want to predict the price of a house: your input features (properties) might include square_foot, number_of_rooms, bathrooms, and so on, and the model will output the predicted price based on the values of your features. Selecting good features that clearly distinguish your objects increases the predictive power of ML algorithms.

What is a feature in computer vision?

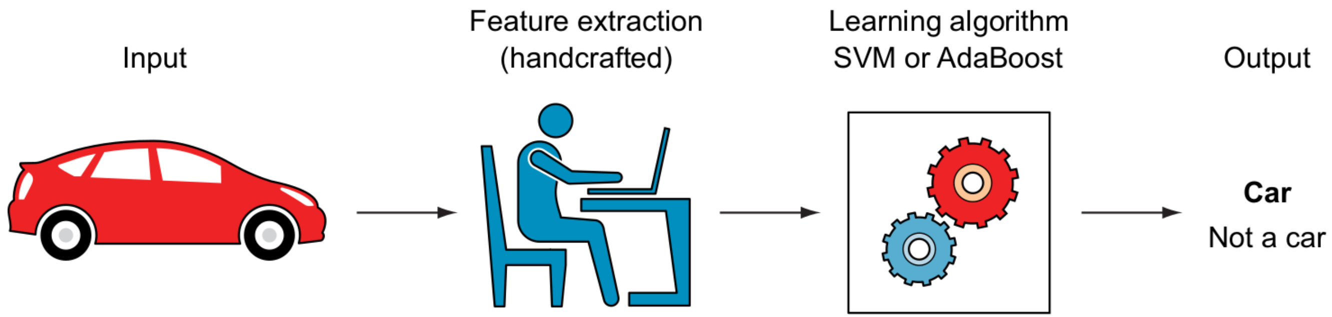


Example input image fed to a feature-extraction algorithm to find patterns within the image and create the feature vector

Feature generalizability

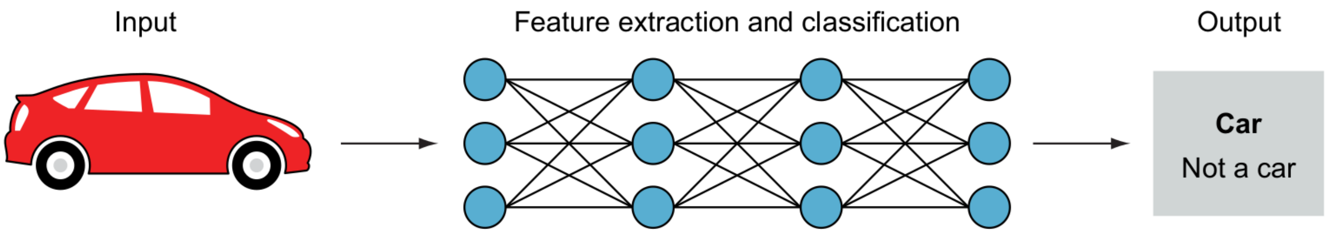


Extracting features: handcrafted



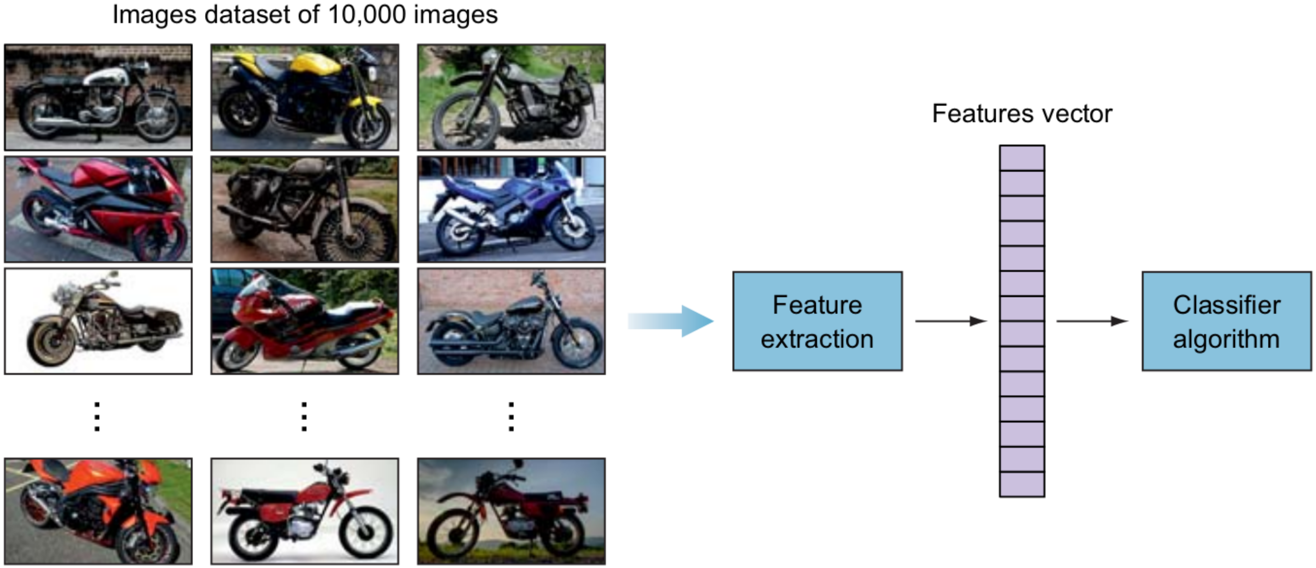
Traditional machine learning algorithms require handcrafted feature extraction

Extracting features: automatic extracting



A deep neural network passes the input image through its layers to automatically extract features and classify the object. No handcrafted features are needed

Extracting features: automatic extracting



Extracting and consolidating features from thousands of images in one feature vector to be fed to the classifier

Demo

Image Preprocessing

The end