# Machine Learning

## Different Data Formats

Edgar F. Roman-Rangel.
edgar.roman@itam.mx

Digital Systems Department.
Instituto Tecnológico Autónomo de México, ITAM.

May 28th, 2021.

# Outline

Data formats

Text

Images

# Intro

So far:

- Supervised and non-supervised ML.
- Classification, regression, clustering, dimensionality reduction.
- Always assuming we already got numeric data: vectors.

What about other types of data? E.g.,

- Text,
- Images or video,
- Audio,
- Radio frequencies,
- Etc.

Data must be converted into a numeric descriptor, i.e., vector.

# Different formats

@ - 64   01000000
A - 65   01000001
B - 66   01000010

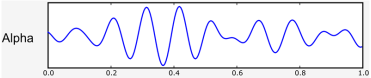
Most types of data can be understood as either:

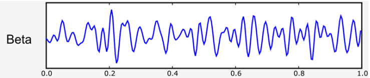
## Static data

Vectors, as we already know.

## Sequential data

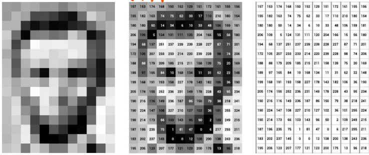
Text, sound, voltage, etc.




## Spatial data

0-255

Images. 

Or a combination of both, e.g., video.

# Exploit other formats

Design filters that extract relevant statistics, create vectors.

# Outline

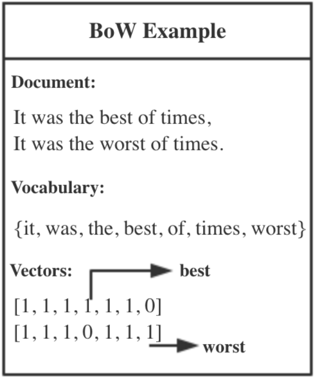
Data formats

Text

Images

# BoW

Bag-of-words (BoW): vector that counts frequency of words.

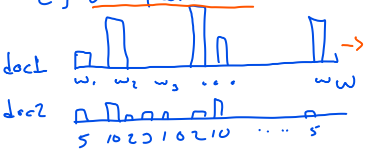



Empezamos con N documentos

1. Definir diccionario común a los N documentos

2.- Para cada doc, creamos un vector de frecuencias de las palabras del doc

Ej; W palabras




$\rightarrow w_i = \frac{w_i}{\sum w}$

Idea: documents of similar topic, have similar word distribution.

# Considerations

- Put all characters in lowercase.
- Remove punctuation and special characters.
- Remove numbers.
- Remove stopwords (articles, prepositions, etc).
- Use lematization or stemming.

# T~~D~~F-IDF

Term frequency - inverse document frequency (tf-idf): used for weighting each term with a inverse frequency with respect to documents: terms appearing in all documents are of low relevance.

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

**TF-IDF**

Term $x$ within document $y$

$tf_{x,y}$ = frequency of $x$ in $y$

$df_x$ = number of documents containing $x$

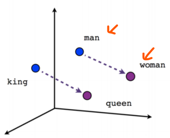
$N$ = total number of documents

# Embeddings

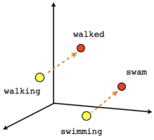
$w_1$ 000001 ← Diccionario del corpus
$w_2$ 000010
$w_3$ 001000 →
⋮
$w_W$ 1000000

Starting from one-hot encoding vectors, find rich dense representation that capture semantic context of words.

$V_{king} - V_{man} + V_{woman} = V_{queen}$

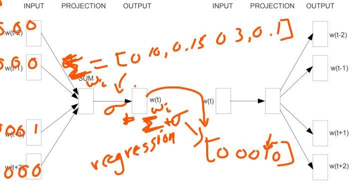



Male-Female




Verb tense

001000
010000
000001
010000
000010




$\Sigma$ = [0 10, 0.15 0 3, 0.1]

regression

[0 0 0 1 0]

CBOW

Skip-gram

# Outline

Data formats

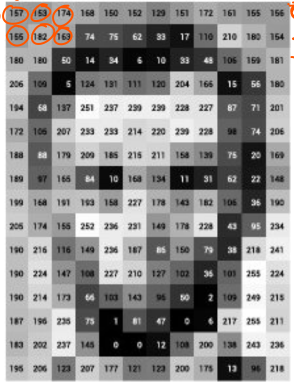
Text

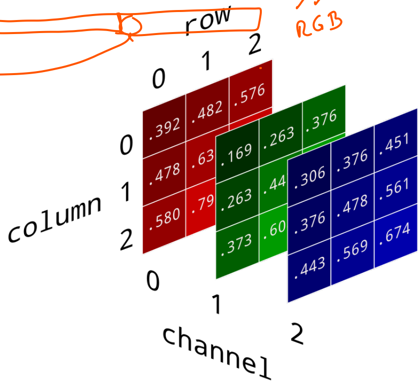
Images

# Image formation

Each pixel indicates a relative amount of light captured by a sensor.

• Enteros 0, 255
• reales 0, 1





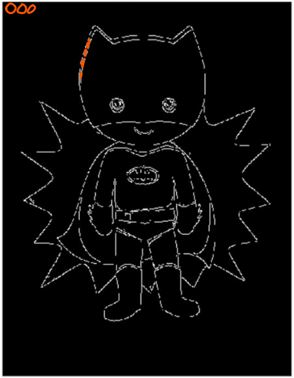
# Derivatives in 2D

Edge detector: $\frac{\partial I}{\partial x}$ , $\frac{\partial I}{\partial y}$

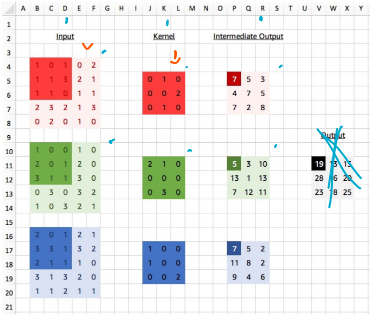
111

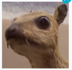
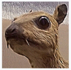
Original Image

Edge Image



# Convolution




| | | |
|---|---|---|
| Identity | $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ |  |
| Edge detection | $\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$ |  |
| | $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ |  |
| | $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ |  |
| Sharpen | $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ |  |
| Box blur (normalized) | $\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ |  |
| Gaussian blur 3 × 3 (approximation) | $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ |  |

# HOG

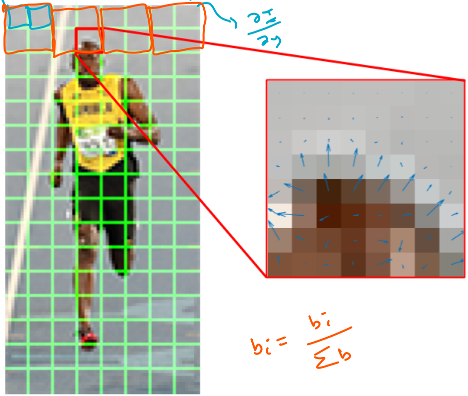
Histogram-of-Oriented-Gradients.

histogram

64 pix

128 pix

$\frac{\partial I}{\partial x}$ $\frac{\partial I}{\partial y}$



$b_i = \frac{b_i}{\Sigma b}$

$mag = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$

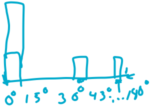
$\theta = \arctan\left(\frac{\partial y}{\partial x}\right)$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | 3 | 4 | 4 | 3 | 4 | 2 | 2 |
| 5 | 11 | 17 | 13 | 7 | 9 | 3 | 4 |
| 11 | 21 | 23 | 27 | 22 | 17 | 4 | 6 |
| 23 | 99 | 165 | 135 | 85 | 32 | 26 | 2 |
| 91 | 155 | 133 | 136 | 144 | 152 | 57 | 28 |
| 98 | 196 | 76 | 38 | 26 | 60 | 170 | 51 |
| 165 | 60 | 60 | 27 | 77 | 85 | 43 | 136 |
| 71 | 13 | 34 | 23 | 108 | 27 | 48 | 110 |

**Gradient Magnitude**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 80 | 36 | 5 | 10 | 0 | 64 | 90 | 73 |
| 37 | 9 | 9 | 179 | 78 | 27 | 169 | 166 |
| 87 | 136 | 173 | 39 | 102 | 163 | 152 | 176 |
| 76 | 13 | 1 | 168 | 159 | 22 | 125 | 143 |
| 120 | 70 | 14 | 150 | 145 | 144 | 145 | 143 |
| 58 | 86 | 119 | 98 | 100 | 101 | 133 | 113 |
| 30 | 65 | 157 | 75 | 78 | 165 | 145 | 124 |
| 11 | 170 | 91 | 4 | 110 | 17 | 133 | 110 |

**Gradient Direction**

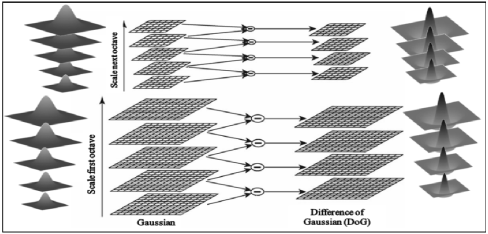


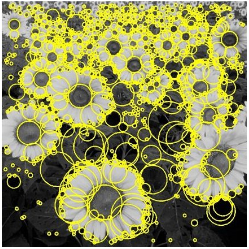

9-bins

$8 \times 16 \times 9 = 1152\text{-D}$

# Local image descriptor

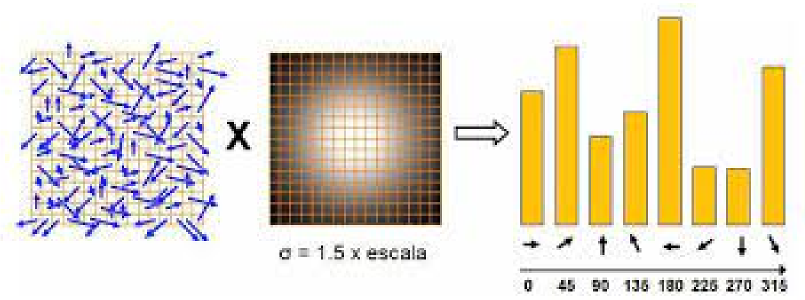
Detect points of interest (PoI): corners of blobs.



# SIFT

Scale-Invariant Feature Transform



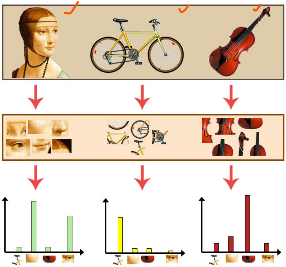

Each local descriptor is a 128-D vector. There are as many local descriptors as PoI's were detected.

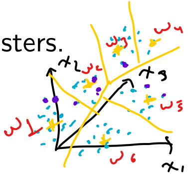
# BoVW

Count the frequency of *visual words* (types of local descriptors).

Descriptors are vectors in $\mathbb{R}^N$, let's map them to $\mathbb{Z}$.

1. Grab a set of local descriptors.
2. Use a clustering algorithm to group them in $D$ clusters.
3. Label each descriptor with the index of its cluster.
4. Create a $D$-dimensional vector of visual words.






img1 = $\omega_1$ $\omega_2$ $\omega_3$ $\omega_4$ $\omega_5$ $\omega_6$

img2 =

# Q&A

Thank you!

edgar.roman@itam.mx