

SSP 100 - 10

How Computers See

10 / 20 / 2026

Instructor: Michael Eckmann

Today's Topics

- Questions/Comments?
- Results of survey
- Review of some recent topics
- In-class exercise on Histograms and distance between them

Survey results

- First let's look at the results of the survey specifically on the topic of how comfortable you each are on the various technical topics taught.
- Thank you to AJ for using time yesterday giving the survey and for you all for responding so thoughtfully to it.

Survey results

- This is not too surprising as I gave you an in-class exercise for doing convolution on an image for edge detection so that the process is understood.
- Whereas I didn't give exercises on Histograms, and later material.

Plan for today

- Let's do a review of some of the recent technical topics that you aren't as comfortable with and let me fit them in an overview within the history of CV.
- Then at end of class you can work on an in-class exercise on histograms, L1 and L2 distance.

Representations

- Rather than use the raw pixels of the images themselves to achieve a task, we create representations of the images that are then used for achieving the task.
- Examples:
 - Histogram of color information to represent an image - to be used for content based image retrieval
 - Histogram of oriented gradients (HOG) - can be used to detect people in images among other things
 - Canny edges to detect boundaries of main objects in an image

Representations

- Any representation choice is a summary of the image data and necessarily ignores some content of the image
- Examples:
 - Color histogram ignores where the pixel colors are located in the image --- only keeps counts of pixels of different colors
 - Canny edges ignores the original image pixels --- only keeps connect edges of high contrast representing borders

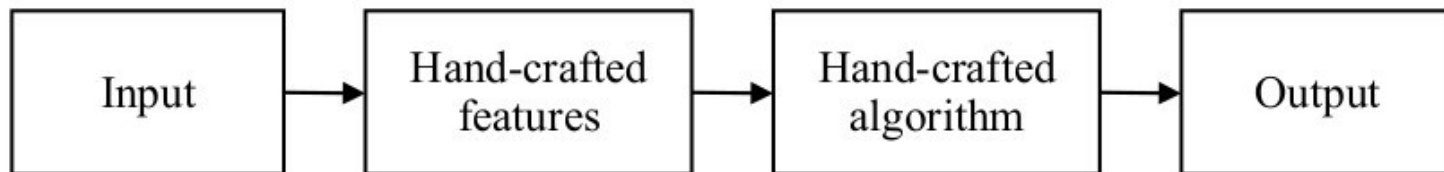
Hand-crafted features vs. learned features

- Color Histograms, HOG and Canny edges are all hand-crafted features.
- In the past decade or so typically we let CNNs learn from experience of a dataset what are good features for various tasks.
- For example: for person detection HOG hand-crafted features, as well as CNN learned features performs best for accuracy of person detection.
- The addition of the HOG hand-crafted features also speeds up the learning (the CNN doesn't need to learn those high quality HOG features, it only learns additional features to get better accuracy.) This is an interesting case of a hybrid approach. Let me point out a couple of recent papers.

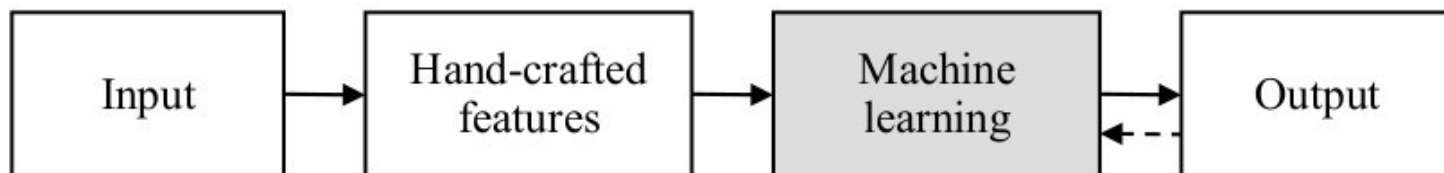
Implementations

- Let me show the color histogram programs that I did a while ago determine similarity of images
- Let me also show you a program that computes HOG features and uses an Machine Learning algorithm that was trained on standing people for person detection.

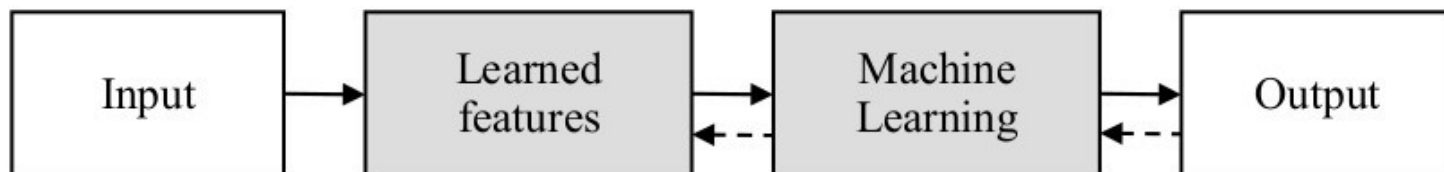
History of computer vision



(a) Traditional vision pipeline



(b) Classic machine learning pipeline



(c) Deep learning pipeline

Types of vision systems

- The color histogram program for content based image retrieval uses hand-crafted features and a hand-crafted algorithm.
- The HOG person detector uses hand-crafted features and a pre-trained machine-learning algorithm.
- A CNN for person detection would be an example using the deep learning pipeline.
- A CNN with HOG for person detection would be a hybrid (hand-crafted features with learned features) approach.

Review of Histograms

- A histogram can be thought of as an array of counts.
- Each bin represents something and the value in a bin is a count of how many items belong in that bin.
- Non-image example on the board of bins being height ranges of people and values a count of the people for each height range.

Review of Histograms

- We can compute a gray level histogram of 256 bins and the value in each bin is a count of how many pixels have that bin gray value.
- We can compute a gray level histogram of say 8 bins where each bin is a consecutive range of 32 gray values.
- Example on the board.

Histogram

- If we are considering a histogram of an RGB image, we could have over 16 million bins, 1 bin for each different RGB value.
- But just like gray value 26 and 27 look about the same brightness to our eyes, say $\text{RGB}=(120,30,20)$ looks about the same color to us as $\text{RGB}=(121,28,22)$.
- Example of 16 bins on the next slide (R and B divided into 2 groups and G divided into 4 groups)

Histogram

- For color histograms, since we have multiple channels, we need a way to combine ranges of all the channels into a bin. Consider these 16 bins for a color image histogram:
- Bin0 those with R in 0-127, G in 0-63, B in 0-127
- Bin1 those with R in 0-127, G in 0-63, B in 128-255
- Bin2 those with R in 0-127, G in 64-127, B in 0-127
- Bin3 those with R in 0-127, G in 64-127, B in 128-255
- Bin4 those with R in 0-127, G in 128-191, B in 0-127
- Bin5 those with R in 0-127, G in 128-191, B in 128-255
- Bin6 those with R in 0-127, G in 192-255, B in 0-127
- Bin7 those with R in 0-127, G in 192-255, B in 128-255
- Bin8 those with R in 128-255, G in 0-63, B in 0-127
- Bin9 those with R in 128-255, G in 0-63, B in 128-255
- Bin10 those with R in 128-255, G in 64-127, B in 0-127
- Bin11 those with R in 128-255, G in 64-127, B in 128-255
- Bin12 those with R in 128-255, G in 128-191, B in 0-127
- Bin13 those with R in 128-255, G in 128-191, B in 128-255
- Bin14 those with R in 128-255, G in 192-255, B in 0-127
- Bin15 those with R in 128-255, G in 192-255, B in 128-255

Review of Histograms

- Computing distance between histograms.
- L1 and L2 distances on next slides

Distance measures

- the L1 distance (aka Manhattan or Taxicab) is the sum of the absolute values of the differences of each dimension
- the L2 distance (aka Euclidean) is the square root of the sum of squares of the differences of each dimension
- These are easy to see in 2 dimensions.
- Say we have $p = (p_1, p_2)$ and $q = (q_1, q_2)$ then the L1 distance between them is: $\text{abs}(p_1 - q_1) + \text{abs}(p_2 - q_2)$
- The L2 distance between them is: $\text{sqrt}((p_1 - q_1)^2 + (p_2 - q_2)^2)$

Distance measures

- In general for n dimensions,
- if $p=(p_1, p_2, \dots, p_n)$ and $q = (q_1, q_2, \dots, q_n)$
- the L1 distance between them is:

$$\text{abs}(p_1-q_1) + \text{abs}(p_2-q_2) + \dots + \text{abs}(p_n-q_n)$$

- The L2 distance between them is:

$$\text{sqrt}((p_1-q_1)^2 + (p_2-q_2)^2 + \dots + (p_n-q_n)^2)$$

Distance measures

- An example where the vectors have 5 values (5 dimensions)
- $p = (5,3,2,6,9)$ and $q=(4,4,4,7,6)$ and $r = (10,6,4,5,0)$
- Let's see whether p is closer to q or r using L1
- L1 distance between p and q = $|5-4| + |3-4| + |2-4| + |6-7| + |9-6| = 1+1+2+1+3 = 8$
- L1 distance between p and r = $|5-10| + |3-6| + |2-4| + |6-5| + |9-0| = 5+3+2+1+9 = 20$
- So p is closer to q than r when considering L1 distance
- In other words we determined that p is more similar to q than it is to r

Review of Histograms

- Normalizing histograms before computing distance to be able to compare images of different sizes more fairly.

Review of Histograms

- In class exercises on computing histograms and determining distance between histograms.