Computer Vision 1 March 26th, 2018, 09.00-12.z00

1. Reflection models

    a) The lambertian reflection model describes matte surfaces. This is because the model describes body (diffuse) reflection and not specular reflection.

       It describes an ideal diffuse reflecting surface, because the Image intensity/Brightness is viewing angle independent. We see this because cos \theta is independent of the viewing angle, therefore it describes a matte surface. Which gives of a diffuse reflection.

       It does not describe a glossy surface, because it does not describe how well the surface reflects light in a specular (mirror-like) direction.

    b) The amount of reflected light does not depend on the viewpoint. The surface's luminance is isotropic (uniform in all directions) and the luminous intensity obeys Lambert's cosine law (intensity is proportional to the cosine of the angle $\theta$ between the light source and the surface normal, assuming an ideal diffusely surface).

    c) The maximum intensity of the reflected light depends on the angle $\theta$ and has a maximum for $\theta = 0°$. Cos(0°) = 1 (This is the case when the direction of the light source coincides with the direction of the surface normal.)

    d) The angle between the light source and the surface normal determines the shading (not 100% sure). (<- Yeah I think so: the surface normal depends on the exact shape of the object's surface and it intuitively makes sense that this affects shading)
       value of said region in the previous image. Region

       Intrinsic image decomposition decomposes the image into its shader, reflection and specular components. According to a paper they had given it:
       I = S * R + C, where S = shader, R=reflectance and C = specular

       In the question there is no specular given and in the paper they say R is the albedo, so that leaves cos θ as the answer

    e) If we assume the lambertian reflection model applies then we can compute the light source direction if we know the surface normal, surface albedo and the light source intensity.

f) a = b = c = $\frac{e}{\hat{e}}$ ← The only difference between $R^{\hat{e}}$ and $R^{e}$ is the light source e. Thus if we divide $R^{\hat{e}}$ by $\hat{e}$ and then multiply with e we obtain $R^{e}$.

g) It models the specular highlight in the image. The higher the exponent, the smaller the specular highlight becomes. https://www.scratchapixel.com/lessons/3d-basic-rendering/phong-shader-BRDF

h) From the viewpoint perspective, the shape is circular and the size of s describes the size of the specular reflection. (why circular?) Elliptical is perhaps a better way to phrase it

i) The term which adds the specular highlight to our model is e(cos phi)^s. The color of the highlight is dependent on the light source, because e describes our light source. **NOTE:** e is intensity so I don't think this is correct.

2. Photometric Invariants

a) $\frac{(2cos(\theta)ep_R+e(cos(\phi)^s)-(cos(\theta)ep_B+e(cos(\phi)^s)-(cos(\theta)ep_G+e(cos(\phi)^s)}{(cos(\theta)ep_B+e(cos(\phi)^s)-(cos(\theta)ep_G+e(cos(\phi)^s)} = \frac{2cos(\theta)ep_R-cos(\theta)ep_B-cos(\theta)ep_G}{cos(\theta)ep_B-cos(\theta)ep_G} = \frac{cos(\theta)e(2p_R-p_B-p_G)}{cos(\theta)e(p_B-p_G)} = \frac{2p_R-p_B-p_G}{p_B-p_G}$

b)
$$\frac{R_{x1}G_{x2}}{R_{x2}G_{x1}} = \frac{(\cos\theta_{x1} * e_{x1} * p_{R_{x1}})(\cos\theta_{x2} * e_{x2} * p_{G_{x2}})}{(\cos\theta_{x2} * e_{x2} * p_{R_{x2}})(\cos\theta_{x1} * e_{x1} * p_{G_{x1}})} = \frac{p_{R_{x1}} * p_{G_{x2}}}{p_{R_{x2}} * p_{G_{x1}}}$$
If the object is homogeneously colored, then this becomes 1, as $\frac{p_R * p_G}{p_R * p_G} = 1$.

c) (not sure!!!) I would choose the first model because the second one is equal to 1 for every homogen. colored object, making it useless at discriminating between different objects.

d) This model is stable as the derivative with respect
$$\sigma_q = \sqrt{(\frac{\partial q}{\partial R}\sigma_R)^2 + (\frac{\partial q}{\partial B}\sigma_B)^2} = \sqrt{(2*4)^2 + (4*4)^2} = 8\sqrt{5}$$
to either parameter never approaches infinity (or negative infinity) as long as the parameters do not approach it either.

e) For a decreasing intensity, G decreases as well. As the intensity approaches zero, so will G. And if G approaches zero, we can see that the derivative approaches infinity. As such, the model is unstable.
$$\sigma_q = \sqrt{(\frac{\partial q}{\partial R}\sigma_R)^2 + (\frac{\partial q}{\partial B}\sigma_B)^2 + (\frac{\partial q}{\partial G}\sigma_G)^2} = \sqrt{(\frac{1}{2G}\sigma_R)^2 + (\frac{1}{2G}\sigma_B)^2 + (-\frac{R+B}{2G^2}\sigma_G)^2} =$$
$$\sqrt{(\frac{4}{2*40})^2 + (\frac{4}{2*40}\sigma_B)^2 + (-4 * \frac{20+60}{2*40^2})^2} = \sqrt{\frac{1}{400} + \frac{1}{400} + \frac{1}{100}} = \frac{1}{10}\sqrt{\frac{3}{2}}$$

3. Filters and Image Features

a) We first need to mirror our convolution filter in its centre before applying it:

$$\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix} \rightarrow \begin{bmatrix}9 & 8 & 7\\6 & 5 & 4\\3 & 2 & 1\end{bmatrix}$$

Then we can convolve, resulting in:

$9a + 8b + 7c + 6d + 5e + 4f + 3g + 2h + 1i$.

b) Convolving an images using filter g results in a sharpening of the image, edges are highlighted.

c) Smoothing along x-axis → 1D gaussian filter

d) h = [1, 1, 1] $*$ [1, 1, 1] $*$ [1, 1, 1]
 = [1, 2, 3, 2, 1] $*$ [1, 1, 1]
 = [1, 3, 6, 7, 6, 3, 1]

e) Edge detection in along x-axis? → 2nd order derivative

f) Gaussian derivative -> can be better for noisy image; blurs edges but removes the noise (noise/edge trade-off)

g) Remember that for convolution, we first have to mirror flip the filters. As such, the x-filter becomes (1 -1) and the y-filter becomes (-1 1)^T.

This table shows the fx of P:

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

This table shows the fy of P:

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 2 | 2 | 2 | 2 | 2 |

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

This table shows the fx of Q:

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 2 | 0 | 0 |
| 0 | 0 | 2 | 0 | 0 |
| 0 | 0 | 2 | 0 | 0 |

This table shows the fy of Q:

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 2 | 2 | 2 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

h)

$$M_P = \begin{pmatrix} 0 & 0 \\ 0 & 20 \end{pmatrix}$$

$$M_Q = \begin{pmatrix} 12 & 4 \\ 4 & 12 \end{pmatrix}$$

i)

$$\det(M_Q - \lambda I) = \det \begin{pmatrix} 12-\lambda & 4 \\ 4 & 12-\lambda \end{pmatrix} = (12-\lambda)^2 - 16 = 144 - 24\lambda + \lambda^2 - 16 = \lambda^2 - 24\lambda + 128 = (\lambda - 16)(\lambda - 8)$$

We thus find the eigenvalues of $\lambda_1 = 16$ and $\lambda_2 = 8$. We can use the eigenvalues to determine if there is a corner in our image. If $\lambda_1 >> \lambda_2$ or $\lambda_2 >> \lambda_1$, then we most likely have an edge. If $\lambda_1 \approx \lambda_2 \approx 0$, then the area most most likely homogeneous. Finally, if $\lambda_1 \approx \lambda_2 >> 0$, then we most likely have found a corner as there is great variance in both the x and y directions. Given that we have found $\lambda_1 = 16$ and $\lambda_2 = 8$, we conclude that it is likely that we have found a corner.

j) $(M - \lambda I)v = 0 \ldots$

If it contains edges at two or more orientations (i.e., a corner), there will be two large eigenvalues (**the eigenvectors will be parallel to the image gradients**).

Filling in 8 for $\lambda$ in $Qx = \lambda X$ gives

$4x_1 + 4x_2 = 0$ and $4x_1 + 4x_2 = 0$ thus $x_1 = -x_2$

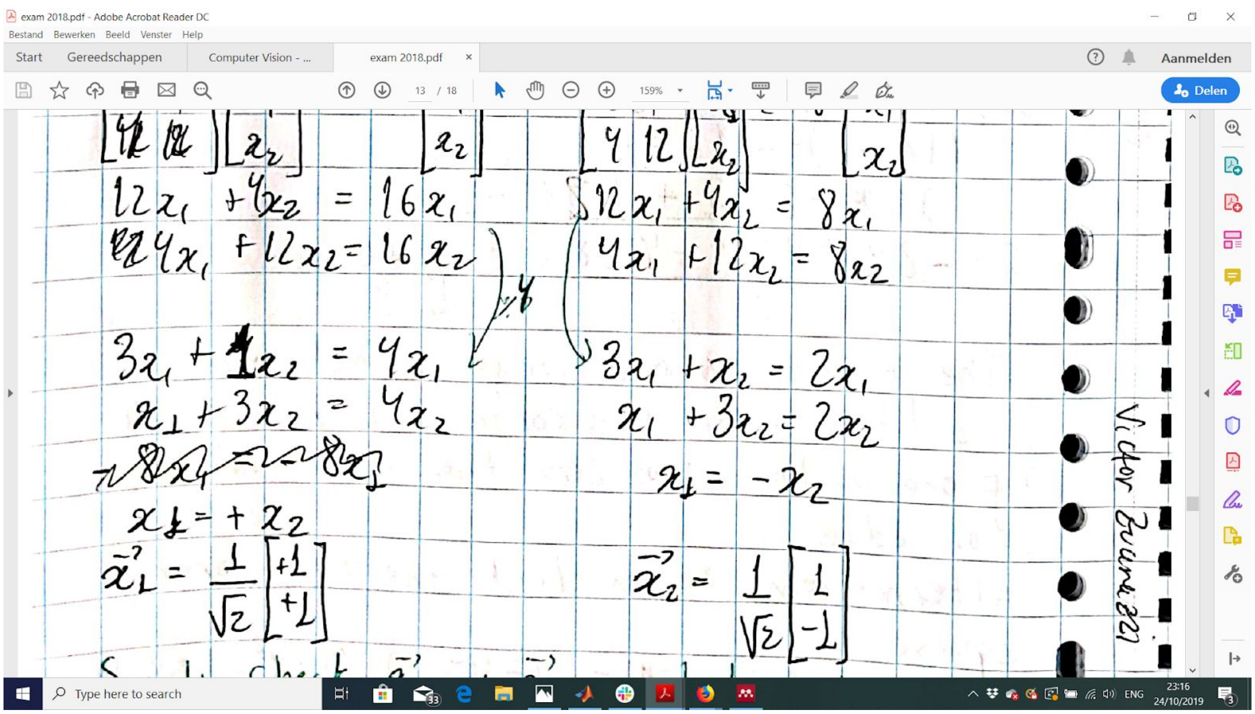
Filling in 16 for $\lambda$ in $Qx = \lambda X$ gives

$4x_1 = 4x_2$ and $4x_1 = 4x_2$ thus $x_1 = x_2$

Therefore, eigenvector for lambda = 8 -> C * [1, -1].T

Sim. eigenvector for lambda = 16 -> C * [1,1].T

(which are indeed orthogonal)




4. Object Classification, Detection and Performance

   a) I = [0, 1, 0, 0, 1, 0, 0, 1, 0].T

      X =[1, 0, 1, 0, 1, 0, 1, 0, 1].T

      M * I = [2, 2, 0.5, 1] -> apply softmax: [0.386, 0.386, 0.086, 0.142]

M * X = [0.5, 2, 0, 4.8] -> apply softmax: [0.01, 0.05, 0.007, 0.92]

Image I is hard to classify as it has the same score for the first 2 classes. Image x likely belongs to class 4.

b) High # of parameters, pixel-space too sensitive to translations/rotations etc.

c) ((32-5)/1)+1 = 28, 10 neurons, so 28x28x10 ;((image_size-filter_size)/stride)+1 is the formula for computing the amount of neurons needed for 2d, this times the depth gives the answer above.

**d) Standard CNN**: Convolutional layer, max pooling layer, fully connected layer (ReLU)?

**AlexNet**: ~~Dropout layer,~~ ~~dense layers~~ + standard CNN layers + ReLU + ?Overlapping max pooling?

[A nice comparison between different CNN's]

e) **Max-Pooling**: with max pooling an image is divided into an x amount of regions, depending on the defined size of the region, where each region gets assigned the assigned the max value of said region in the previous image. Region

| | | | |
|---|---|---|---|
| 16 | 22 | 3 | 22 |
| **92** | 1 | **67** | 3 |
| 2 | 68 | 12 | 6 |
| **79** | 24 | 2 | **14** |

Will become:

| | |
|---|---|
| 92 | 67 |
| 79 | 14 |

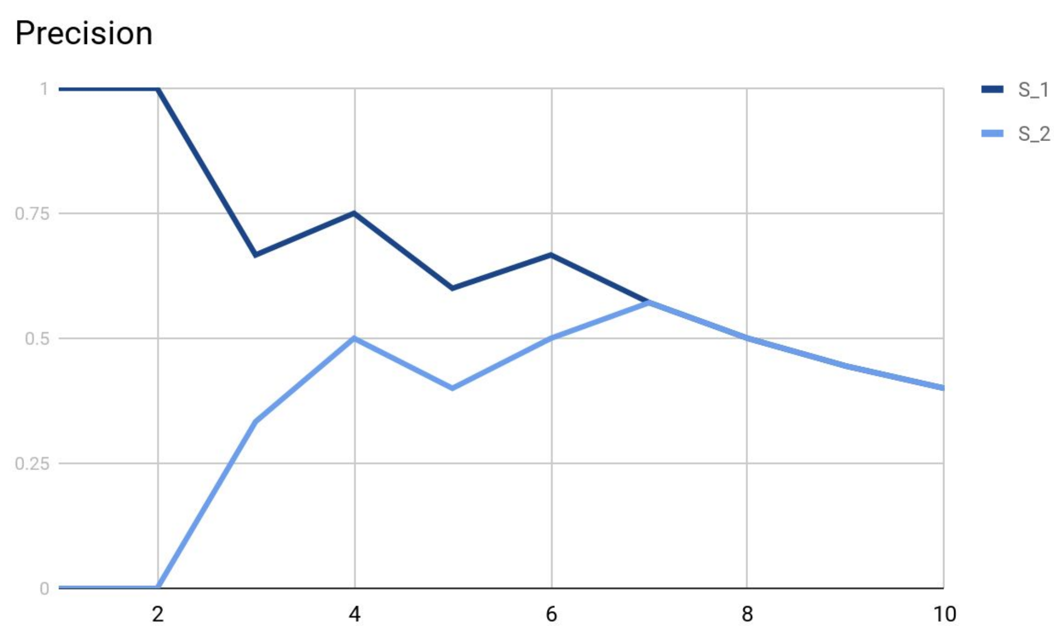
It is used for reducing the dimensionality, also introduces a (slight) invariance to translations

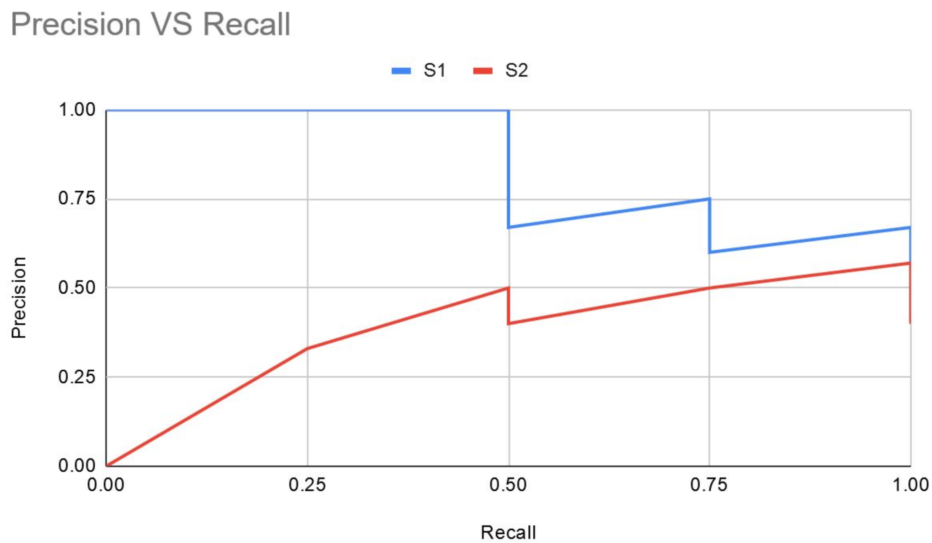
f) (roughly) Initialize window position, scale, aspect ratio. Slide the window across the whole image + changing aspect ratios and scale. Classify the patches selected by the window at each location.

g) **Disadvantage**: The box might not fit the object, Multiple boxes around the same object, the number of classes is large.  Many boxes, many scales, many box ratios, many classes = computationally expensive..!

   **Advantage**: Given you have an infinitely powerful computer the algorithm will find all the possible boxes where all the defined classes might be located (within a reasonable timespan).

h) anFrom the graph below we can conclude that both recognition systems converge to the same badness at recognition, but the S_1 system has the best top 6 performance.






You cannot estimate posterior probabilities for this ECOC model.

2016 exam

1.

a) The color of the reflected light is R,G,B (same color as incoming light ???)

b) The amount of reflected light does not depend on the viewpoint. The surface's luminance is isotropic (uniform in all directions) and the luminous intensity obeys

c) The maximum intensity of the reflected light depends on the angle $\theta$ and has a maximum for $\theta = 0°$. Cos(0°) = 1

d) $$X = \cos(\theta)\sum_{\lambda} e(\lambda)p(\lambda)x(\lambda)$$
$$= 6e(\lambda)(0.4334 + 0.5945 + 0.7621 + 0.9163 + 1.0263 + 1.0622)$$
*cos(theta), and not 6e(\lambda)

X = 4.7948, Y = 5.2, Z = 0.0183

x = 0.4211, y=0.5767, z=0.0022

e) Same as d)

X = 0.6167, Y = 0.2293, Z = 0

x = 0.7290, y=0.2710

f) They have different hues, because the chromaticity values do not lay on one line w.r.t. The chromaticity of the reference white light. Their saturation most likely differs as well, as their distance from the white light are also different.

2.

a. Second order = [1 -2 1] third order = [-1 3 -3 1] (using zero padding and doing convolutions on the correlation filter?)

b. [1 1] with a higher alpha meaning a smaller standard deviation. (Value of alpha = 1 ??? ) I think ¼ such that the sum of the filter is 1 and hence does not change the intensity of the output

c.

d.

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |

e. blobs/dots (corners?)

f. Two of the following

| | |
|---|---|
| -1 | 1 |
| 1 | -1 |

g.

h.
i.
j.
k. Illumination and orientation

3.

a.

4.

a.