Exercises (Lecture 2 & 3)

Exercise 1

Sunlight S is given by X = Y = Z = 100, artificial light A is given by X = Y = 100 and Z = 150.

a) Calculate the intensity of the two light sources S and A.

Intensity means the total energy 'emitted' by the three components of the light source: X, Y and Z. Thus, Intensity = X + Y + Z.
I(S) = 300 and I(A) = 350.

b) Calculate the chromaticity values x (= X / (X+Y+Z)) and y (= Y / (X+Y+Z)) and plot these in the chromaticity diagram.

xS=1/3, yS=1/3, xA= 0.286, yA= 0.286

c) What is the estimated hue of the light sources S and A with reference white light B at X=120, Y=100 and Z=100.

The boundary of the chromaticity diagram is where the spectral colors lie. The hue of a color is equivalent to the spectral color that it is closest to. Therefore, we first plot B in the chromaticity diagram (using xB = 0.375, yB = 0.312), then draw a line from the reference light B through A or S onto the chromaticity diagram. Then, the point on the diagram is the hue of the light source.

-

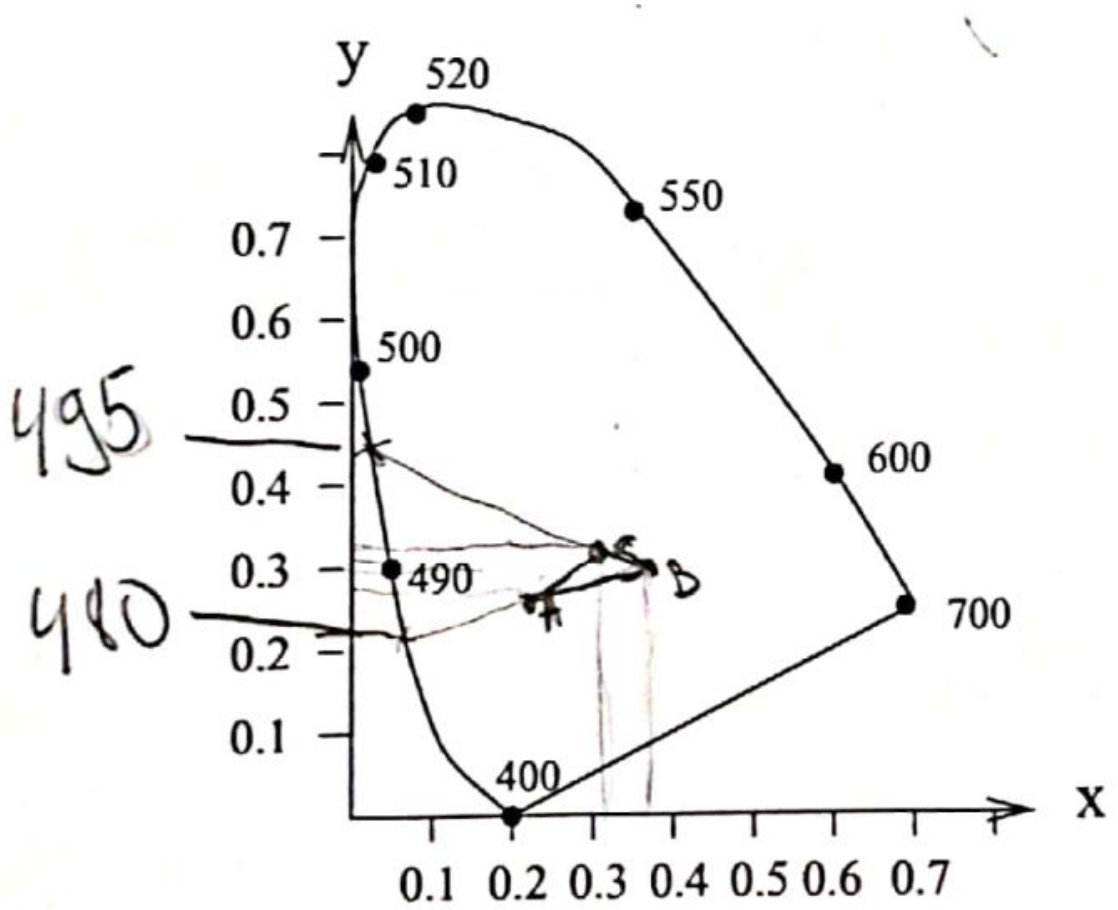


Figure 1: *Chromaticity diagram.*

d) <u>Rank the light sources with respect to their saturation.</u>

Saturation is defined as the 'purity' of a color. Alternatively: it describes how "white" a color is, where a saturation of 0 means the color is fully white. As sunlight is completely white and artificial light isn't, A is more saturated than S.
To compute the actual value, just apply the formula: saturation = (max(X,Y,Z) - min(X,Y,Z)) / (max(X,Y,Z))

s(S)=0, s(A)=⅔, s(B)=⅛ -> rank is A>B>SQ1.c

Hue and saturation are defined in the chromaticity diagram as follows, see Figure 5.b. First a reference white-light point should be defined. This point is defined as a point in the chromaticity diagram that approximately represents the average daylight, for example illuminant C or D65. For a color $G1$, the hue is defined as the wavelength on the spectral curve that intersects the line from reference white through $G1$ to the spectral curve, which is $G2$ at 523 nm. When $||G1||$ is the distance from $P1$ to the white-point, then the saturation is the relative distance to the white-point, given by $\frac{||G1||}{||G2||}$, where $||G2||$ is the distance from $G2$ to the white point. Hence the most saturated colors are the spectral colors.

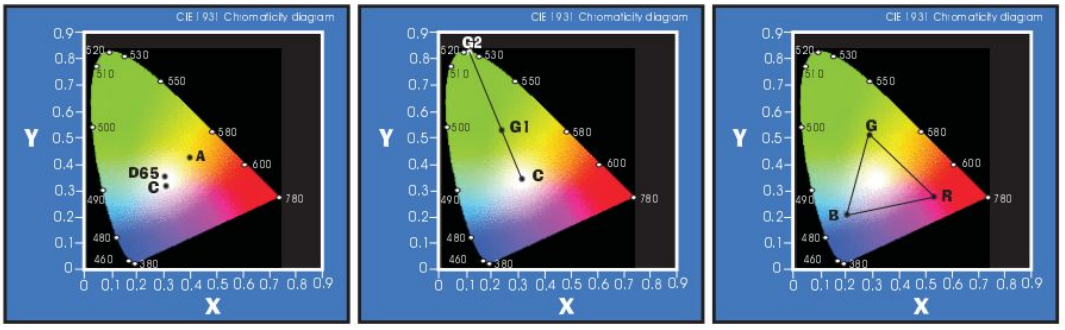
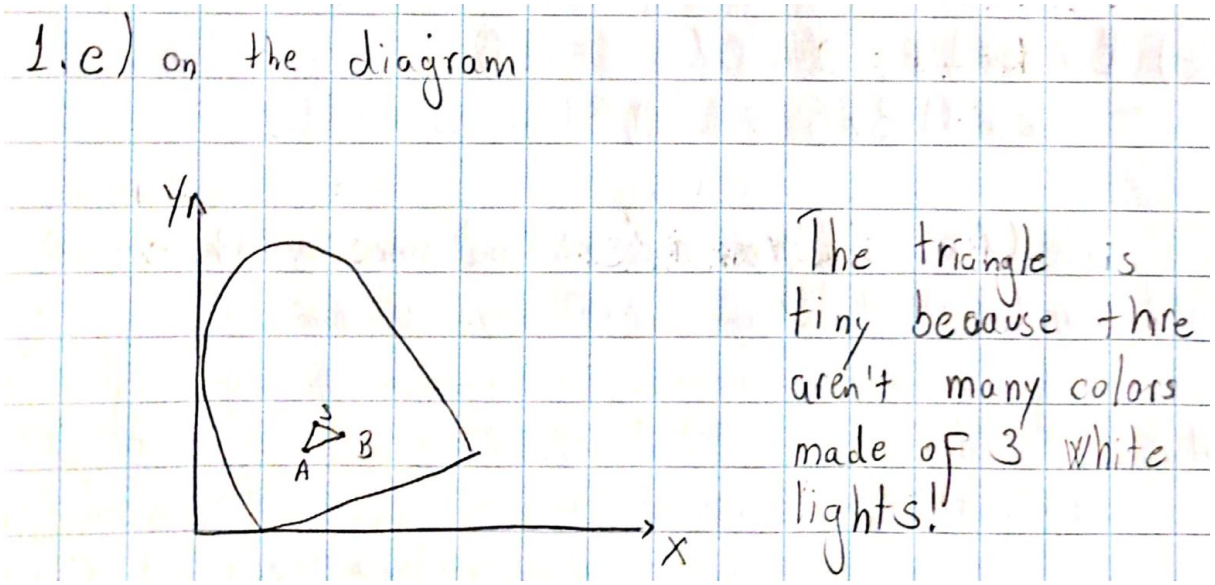


Figure 5: *a. Illuminant A, C and D65 plotted in the xy-plane. b. Definition of hue and saturation under illuminants A and D65. c. Color gamuts.*

For point B = (0.375, 0.3125); white light = (⅓, ⅓)

Saturation = ||G1|| / ||G2|| = sqrt((0.375 - ⅓) ^ 2 + (0.3125 - ⅓) ^ 2) / sqrt((0.375) ^2 + (0.5 - ⅓) ^ 2)) = 0.1135

e) <u>Plot the region of colors which is produced through the mixture of S, A and B.</u>



Exercise 2

$X = \int_{\lambda} K(\lambda)\, \rho(\lambda)\, \bar{x}(\lambda)\, d\lambda$, similar for Y and Z

a) Compute X, Y and Z for a given object color A of 500 nm ($\rho(\lambda 500) = 1$, $\bar{x} = 0.0049$, $\bar{y} = 0.323$ and $\bar{z} = 0.2720$). Further, calculate the chromaticity coordinates of A.

All parts of the formula are given. We end up with X = 0.0049*K, Y = 0.323*K and Z = 0.272*K. Then we calculate the chromaticity coordinates x = 0.008, y = 0.538 and z = 0.45.

b) Plot color A in the chromaticity diagram.

(0.008; 0.538)

c) Given an object color B of 580 nm, find X, Y, Z and x, y, z.

To get X,Y,Z you take the values out of the table that correspond to 580nm and replace them in the formula. Then, having those, you compute x,y,z like this:
x=X/(X+Y+Z),
y=Y/(X+Y+Z),
z=Z/(X+Y+Z)

XB=K0.9163, YB=K0.8700, ZB=K0.0017 and xB=0.512, yB=0.486, zB=0.0

Table 2.1
Tristimulus values of the spectrum colors (CIE 1931 color-matching data, per watt of indicated wavelength)

Wavelength [nm]	$\bar{x}$	$\bar{y}$	$\bar{z}$
400	0.0143	0.0004	0.0679
410	0.0435	0.0012	0.2074
420	0.1344	0.0040	0.6456
430	0.2839	0.0116	1.3856
440	0.3483	0.0230	1.7471
450	0.3362	0.0380	1.7721
460	0.2908	0.0600	1.6692
470	0.1954	0.0910	1.2876
480	0.0956	0.1390	0.8130
490	0.0320	0.2080	0.4652
500	0.0049	0.3230	0.2720
510	0.0093	0.5030	0.1582
520	0.0633	0.7100	0.0782
530	0.1655	0.8620	0.0422
540	0.2904	0.9540	0.0203
550	0.4334	0.9950	0.0087
560	0.5945	0.9950	0.0039
570	0.7621	0.9520	0.0021
580	0.9163	0.8700	0.0017
590	1.0263	0.7570	0.0011
600	1.0622	0.6310	0.0008
610	1.0026	0.5030	0.0003
620	0.8544	0.3810	0.0002
630	0.6424	0.2650	0.0000
640	0.4479	0.1750	0.0000
650	0.2835	0.1070	0.0000
660	0.1649	0.0610	0.0000
670	0.0874	0.0320	0.0000
680	0.0468	0.0170	0.0000
690	0.0227	0.0082	0.0000
700	0.0114	0.0041	0.0000

d) Plot B in the chromaticity diagram.

(0.512, 0.486)

e) Given a color C consisting of the colors A of 500 nm and B of 580 nm. Compute X, Y, Z and x, y, z.

Visually in the chromaticity diagram, when we add up two colors we end up in the middle of the straight line between these two points. The corresponding values are computed as:
$X_c = X_A + X_B$ and likewise for Y and Z. Then we calculate the chromaticity values x, y, z as usual.

XA = 0.0049*K, YA = 0.323*K and ZA = 0.272*K
XB=K0.9163, YB=K0.8700, ZB=K0.0017
XC=(0.0049+0.9163=0.9212)*K,

YC=(0.323+0.8700=1.193)*K,
ZC=(0.272+0.0017=0.2737)*K

xC=0.385, yC=0.499, zC=0.115

f) Plot C in the diagram.

(0.385, 0.499)

g) If the white light source K(lambda) varies only in intensity, what would happen with the values X, Y, Z and x, y, z of the colors A, B and C? What would be the consequences?

Chromaticity is defined as color invariant to intensity (so that dark red and bright red are the same value). Thus, varying the white light source has no influence on the values of x, y and z. However, as we can also see from the values obtained for X, Y and Z, decreasing or increasing K changes the brightness of a color.

h) The tri-stimulus values of a lamp L are as follows X = 98.04, Y = 100.00 and Z = 118.12. Compute the chromaticity coordinates x, y and z and plot L in the diagram.

xL=0.31, yL=0.316, zL=0.373.
(0.31, 0.316)

i) Indicate, by three different lines, the colors which are generated by mixture of L with A, B and C respectively.

We can draw a line from L to each of A, B and C. The points on these lines can be created by mixing L with each of the colors.

j) What is the hue (dominant wavelength) of C with L as reference white?

The same procedure as in 1c. This gives around 540 nm.

k) Order the three colors A, B and C with respect to their saturation.

The further away the points are from the reference white color, the more saturated: A=B > C.

l) What are the complementary colors A^c, B^c and C^c for A, B and C respectively with L as reference white? Are these complementary colors pure (wavelength) or a mixture of pure colors?

These are opposite colors reflected at L. Draw a line from A through L of length 2*(A-L). The point that we end up in (i.e. the point on the straight line that is on the opposite side of L from A) is the complementary color A_C.

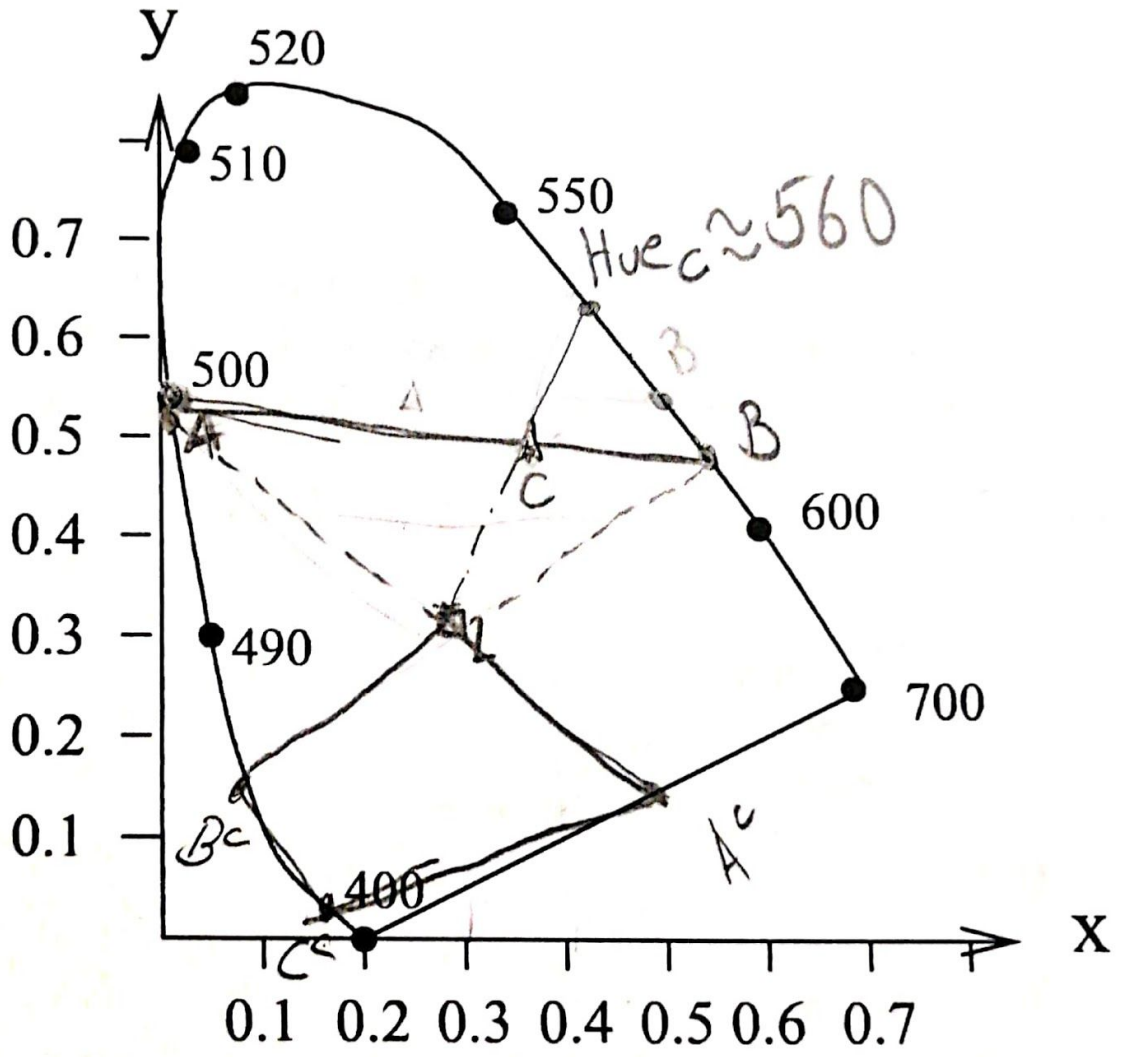


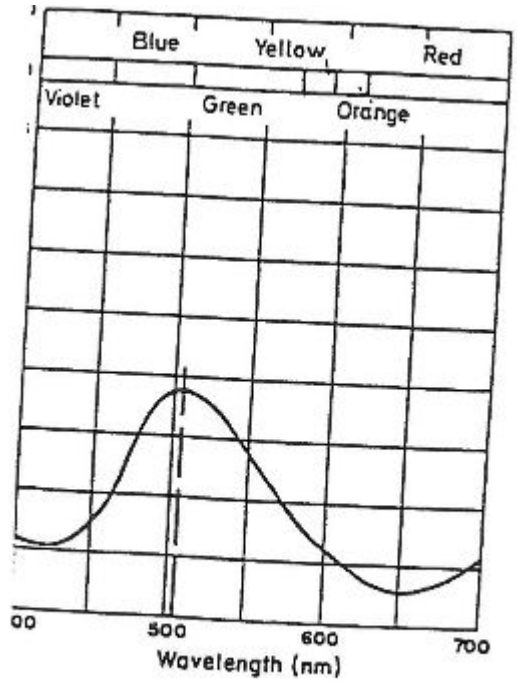
Figure 2: *Color scheme.*

If these colors are on the boundaries of the chromaticity diagram, they are pure colors.

m) Draw the region of colors which are generated by the mixture of A^c, B^c, C^c and L.

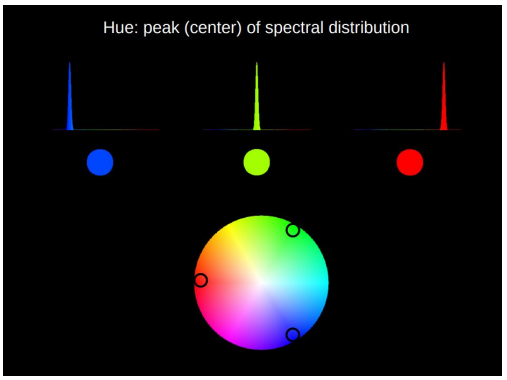
Draw lines between A^c, B^c, C^c, and L; the colors within this area are those that can be generated by mixing them.

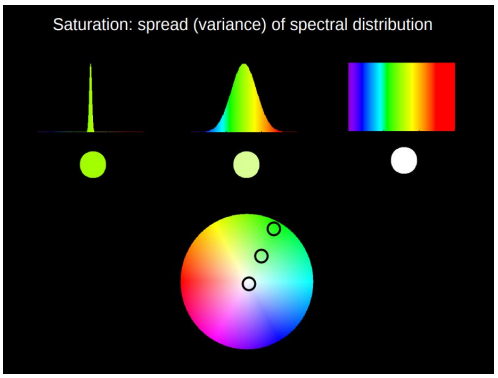
n) Given is a color with a spectral power distribution given in figure A2. Estimate the hue (dominant wavelength) and describe the amount of saturation and intensity. What should be the approximated position of this color in the diagram?

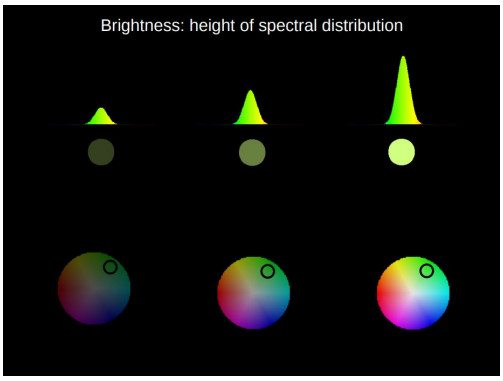


1.2

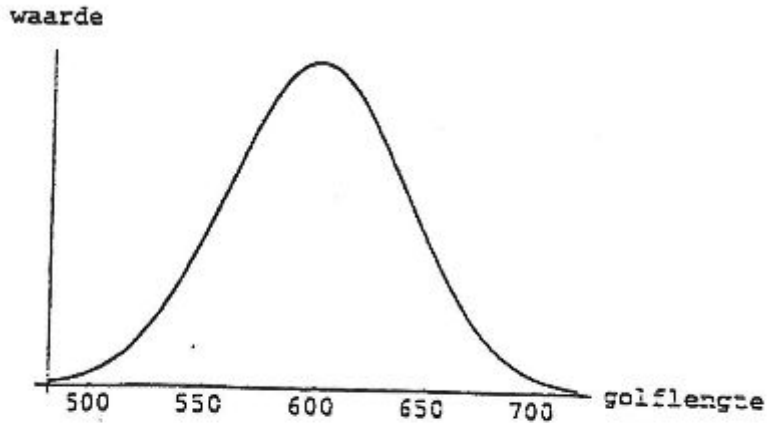
Hue: dominant wavelength, just over 500; corresponding color is green.
Saturation: purity of the color (less the spread/ variance of the curve, more saturated the color - consider white light: horizontal line over the spectral distribution graph- thus very less saturation).
The given distribution is unimodal (one peak), so the saturation would be somewhat high.
Intensity: area under the curve.
Below figures explain the concepts better:







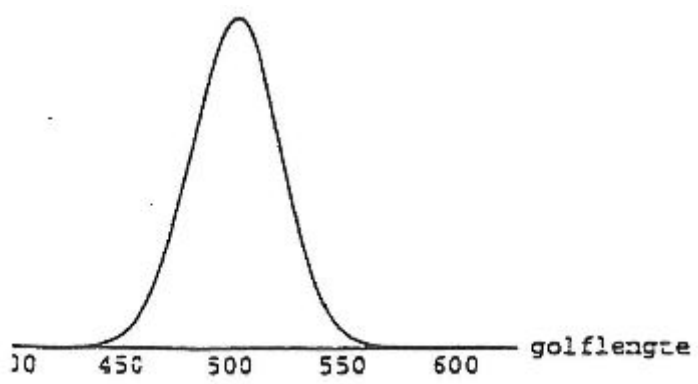
o) <u>See n), figure A3</u>



A.3

Hue: just over 600; corresponding color is orange. Saturation: the distribution is a bit wider than that in A2, so the saturation is lower. Intensity: the area under the curve is larger than in A2, so the intensity is higher.

p) <u>See n) figure A4</u>

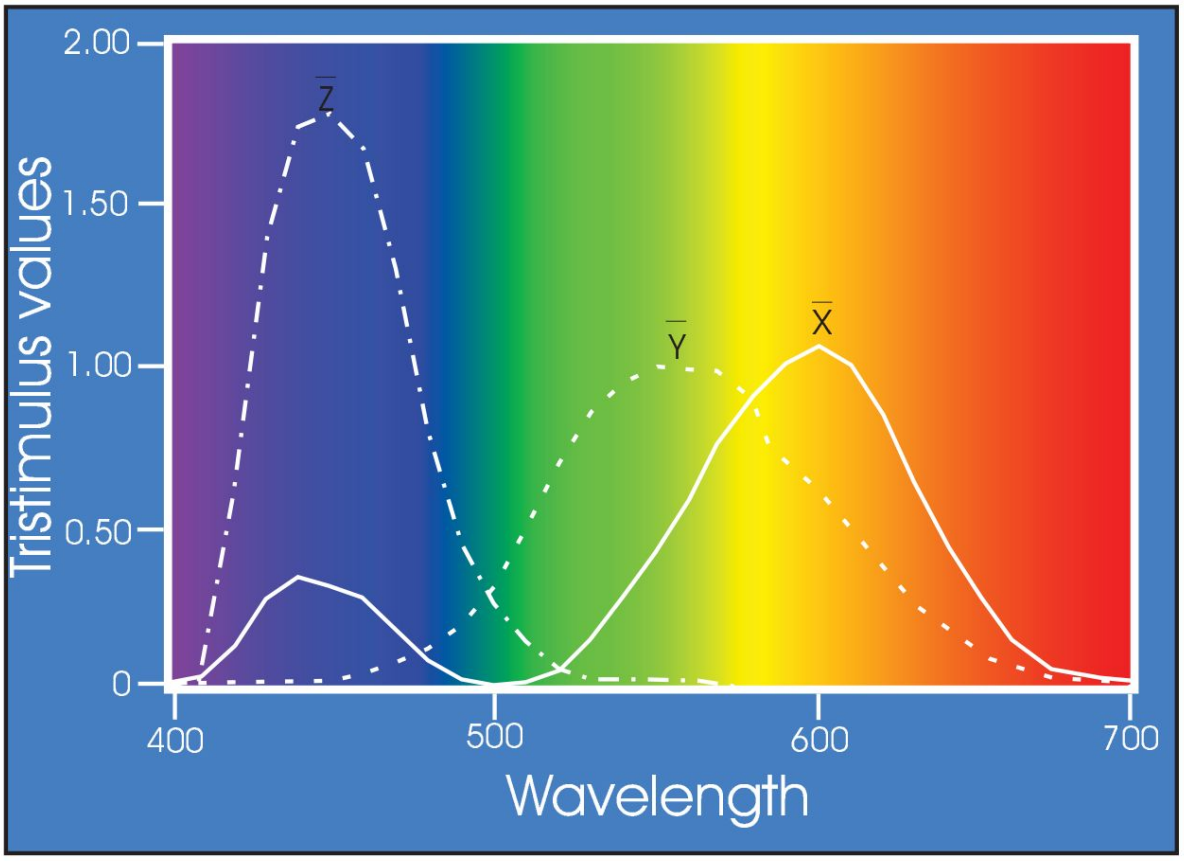


Hue: just over 500, color green. Saturation: the distribution is very narrow so highest saturation of all three. Intensity: area under the curve is quite small, so intensity is lower.
Exercises (Lectures)Computer Vision 1, Master AIExercise 1. Image FilteringBelow are four types of filters.3x3 uniform (box) filter:T=1111111115x5 uniform (box) filter:U=1111111111111111111111111Another 3x3 filterV=-101-101-101The Laplacian using the following matrix values:W=1-21-24-21-21•(a) Which of the following would make an image blurrier, a 3x3 or a 5x5uniform filter? Why?•(b) What kind of image features are detected by the 3x3 edge filtersVandW?•(c) You wish to transform an image by applying a 3x3 uniform filter fol-lowed by the 3x3 Laplacian filterW. Show that this can be implementedby a single a 5x5 filter and calculate the elements of this filter.Exercise 2. Color ConstancyColor constancy is an important issue when recognizing an object independentof the color of the light source. Two simple color constancy algorithms are based1
Exercises (Lectures)Computer Vision 1, Master AIExercise 1. Image FilteringBelow are four types of filters.3x3 uniform (box) filter:T=1111111115x5 uniform (box) filter:U=1111111111111111111111111Another 3x3 filterV=-101-101-101The Laplacian using the following matrix values:W=1-21-24-21-21•(a) Which of the following would make an image blurrier, a 3x3 or a 5x5uniform filter? Why?•(b) What kind of image features are detected by the 3x3 edge filtersVandW?•(c) You wish to transform an image by applying a 3x3 uniform filter fol-lowed by the 3x3 Laplacian filterW. Show that this can be implementedby a single a 5x5 filter and calculate the elements of this filter.Exercise 2. Color ConstancyColor constancy is an important issue when recognizing an object independentof the color of the light source. Two simple color constancy algorithms are based1
Exercises (Lectures)Computer Vision 1, Master AIExercise 1. Image FilteringBelow are four types of filters.3x3 uniform (box) filter:T=1111111115x5 uniform (box) filter:U=1111111111111111111111111Another 3x3 filterV=-101-101-101The Laplacian using the following matrix values:W=1-21-24-21-21•(a) Which of the following would make an image blurrier, a 3x3 or a 5x5uniform filter? Why?•(b) What kind of image features are detected by the 3x3 edge filtersVandW?•(c) You wish to transform an image by applying a 3x3 uniform filter fol-lowed by the 3x3 Laplacian filterW. Show that this can be implementedby a single a 5x5 filter and calculate the elements of this filter.Exercise 2. Color ConstancyColor constancy is an important issue when recognizing an object independentof the color of the light source. Two simple color constancy algorithms are based1

q) For which of the three spectra will a human perceive the highest intensity? Explain your answer.

To determine this, we would need to overlap the spectral power distribution with the distribution that humans can perceive. Whichever spectral power distribution has the largest area under the curve that still falls within the human-perceived distribution has the highest perceived intensity. ~~From these figures I think that that would be figure A4.~~
A3 seems to have the most overlap, it peaks around 600 nm, around the same peak as x, the other two peak at around 500nm, where the human eye is not so sensitive.

Exercise 3:



R=I*k_R cosθ, G=I*k_G cosθ and B=I*k_B cosθ
cos θ = ~n (surface normal) · ~l (direction light source)

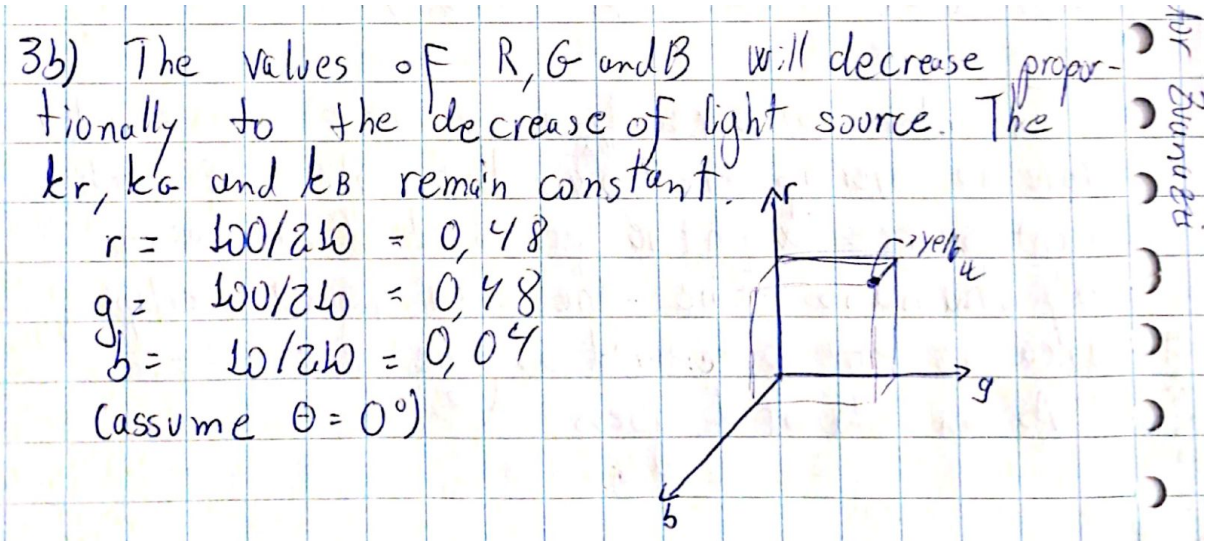
a) Assume that the surface is flat and homogeneously colored. Explain why the intensity is higher when the surface normal coincides with the direction of the light source than observed under an angle with respect to the direction of the light source.

The intensity is dependent on cos θ: since cos(theta) is a dot product between two vectors, that dot product is at maximum value when the vectors coincide. This also corresponds to maximizing cos theta, the angle between the vectors. Cos function peaks at 1 when theta is 0.

b) Assume that the color of the surface is yellow i.e. R=100, G=100, and B=10. Explain what will happen with the values R, G and B if (only) the intensity of the light source will diminish. Plot the positions of the colors in the RGB-color space.

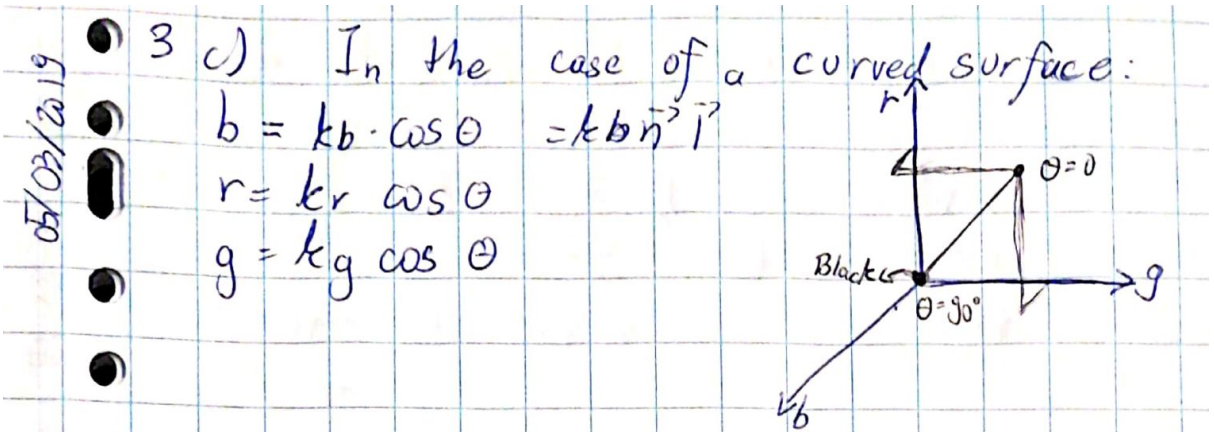
R, G and B values changes evenly: If you only want to change intensity but leave hue and saturation the same, the only way is to proportionally increase R,G and B, a.k.a. R, G and B R *= a, B *= a, G*= a (If the light is twice as intense the values are R= 200, G = 200, B= 20). Any modification different than this will also affect hue and saturation.

The values of R, G and B will decrease proportionally to the decrease of the light source. The k_R, k_G and k_B remain constant:
r = 100/210 = 0.48
g = 100/210 = 0.48
b = 10/210 = 0.05
(assume theta = 0)



c)
In case of a curved (not flat) surface, indicate where the colors will be positioned in the RGB-color space. Explain your answer.

If the surface is curved, the reflected colors will be proportional to cos θ. They will range from Black to the color of the object (yellow in this case):
r = k_R cos θ
b = k_B cos θ
g = k_G cos θ



Recap on color invariant:

What is an invariant? It is an equation that, given a reflectance model, captures some key properties of the object (color coefficients: k_r, k_g, k_b) but is invariant to the environment (light source color, light source direction, view point). Color invariance is interested in capturing the real color of the object.

There is no such a thing as color invariant! There is, however, color invariance to view point, light source direction/intensity, object shape and so on (Zizek dixit).

To see that perhaps it is useful to replace cos fθ with $dot(I_d, S_n)$. Where I_d is the light source direction, S_n is the surface normal and $dot(I_S, S_n)$ is the dot product between them (both vectors have norm = 1 thus dot product = cosine).

If we consider the reflectance model given by question 3g, we have the following:

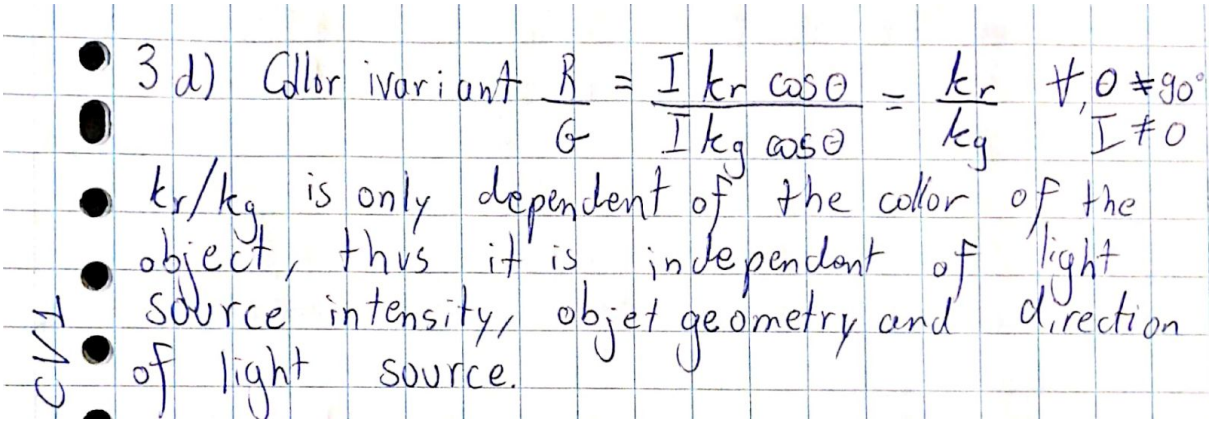
$$\frac{R}{G} = \frac{Ik_r cos(\theta) + Ik_s cos^n(\alpha)}{Ik_g cos(\theta) + Ik_s cos^n(\alpha)} = \frac{k_r cos(\theta) + k_s cos^n(\alpha)}{k_g cos(\theta) + Ik_s cos^n(\alpha)} = \frac{k_r dot(I_d, S_n) + k_s dot(V_p, S_n)^n}{k_g dot(I_d, S_n) + k_s dot(V_p, S_n)^n}$$

It is easy to see that the equation takes more arguments than the ones dependent on surface properties (k_r, k_g, k_b and S_n), namely I_d (Light source direction) and V_p (View point). That is not a color invariant to light source direction and view point, but it is a color invariant to light source intensity (the I dropped, assuming I != 0).

d)

<u>A simple color invariant is given by R/G. Prove that R/G is independent of the (intensity) light source I, object geometry and the direction of the light source.</u>

R/G = (I k_R cosθ) / (I k_G cosθ) = k_R / k_G

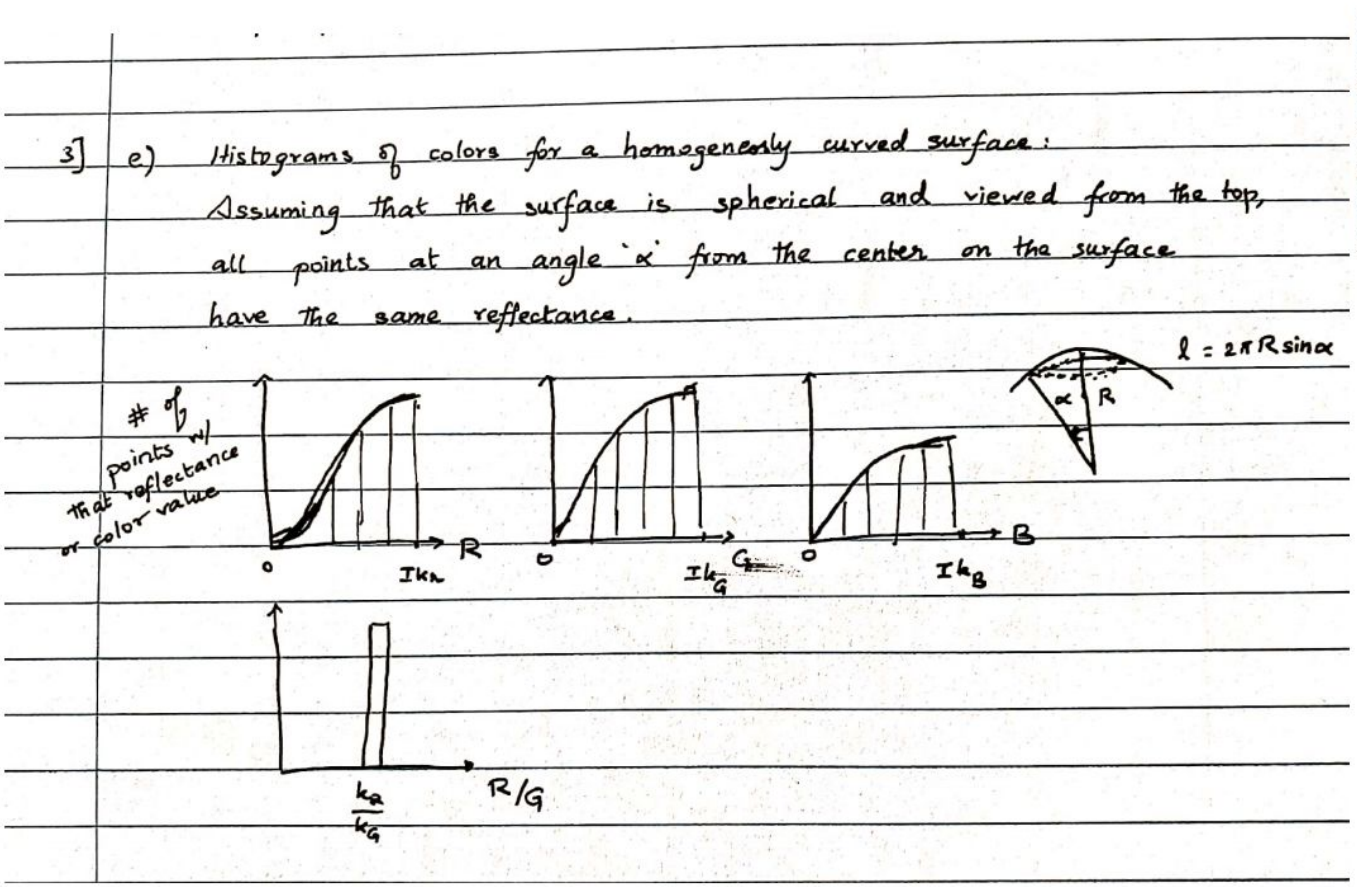


R/G is invariant because it only depends on the ~~reluctance~~ reflectance properties of the object (k_R and k_G) and not on the properties of the light source (I and direction). In this case the invariance due to incoming light intensity and direction, because the specular reflection ($k_s dot(V_p, S_n)^n$) is not taken into account.

e)
<u>The values of R, G and B will vary for a curved surface. Give the approximated shapes of the histograms for a homogeneously (curved) surface for R/G.</u>

Still no idea about the histogram and color model.

How about we do this?
Consider a finite set of points on the surface of the object (sufficient to cover the entire surface). Calculate the values of R, G, B and R/G at all these points. Collect and plot separate histogram of each of the channels (R, G, B and R/G).be R/G will be a single peak whereas the other curves will be distributed.



Chat:
Source?

----- histogram is the other way around (freq. decreases as r or g or b values increase)

----- or uniform for all bins except the highest value which has half the frequency (all other thetas are seen twice) - if viewpoint is at the same position as the direction of light

He explained that today in the tutorial. At least that is what I understood from it.
Oh I thought it was something from the slides.
I haven't seen anything on the slides that explains that. =/
Yeah me neither. Slides are bad
I think they should make all pos-docs go to his classes. To learn how not to give a class.
Lmao
Can you see my name? I just see a bunch of anonymous animals =/
No. You're an anonymous skunk :P
I was called worse things before…
Hahaha. I just googled quokka and it's amazing
It is so cute!
Don't forget to smile! :)
I feel that is an appropriate use of this google sheet!
LOOOL

f)
Consider the same surface. Assume that the surface is glossy (instead of matte). The reflection model is now given by:
$R = I k_R \cos \theta + I k_s \cos^n \alpha,$
$G = I k_G \cos \theta + I k_s \cos^n \alpha,$
$B = I k_B \cos \theta + I k_s \cos^n \alpha.$
... where:

- k_s is the specular reflection coefficient,
- $\cos^n$ depends on the glossiness,
- α depends on the viewing condition.

Plot the colors of the homogeneously colored (shiny) surface in RGB - and rgb-color space.

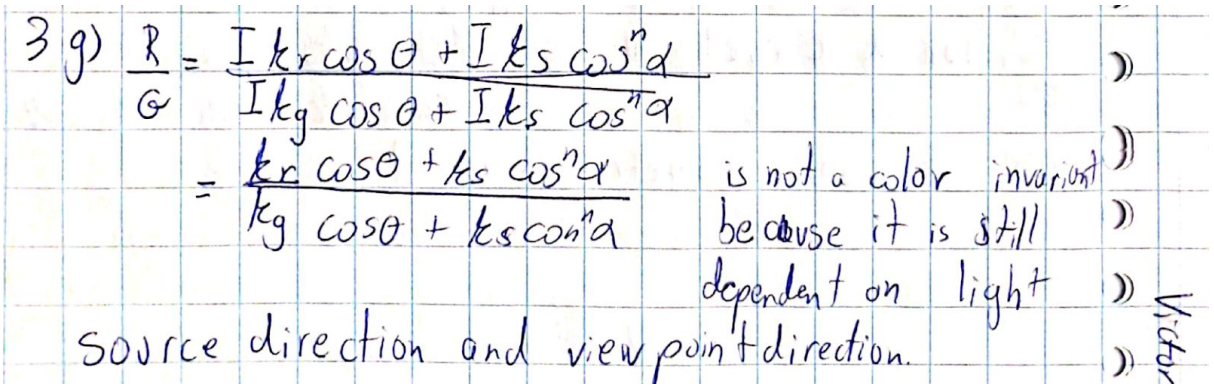
(R–G)/(R−B)=(kR–kG)(kR−kB) ← why are we calculating (R-G)/(R-B)?
How to solve this exercise?

(There are references to the rgb-color space in Pages 405 and 415 of "Lectures 2-3 Image Formation - Color Shape and Texture.pdf".)

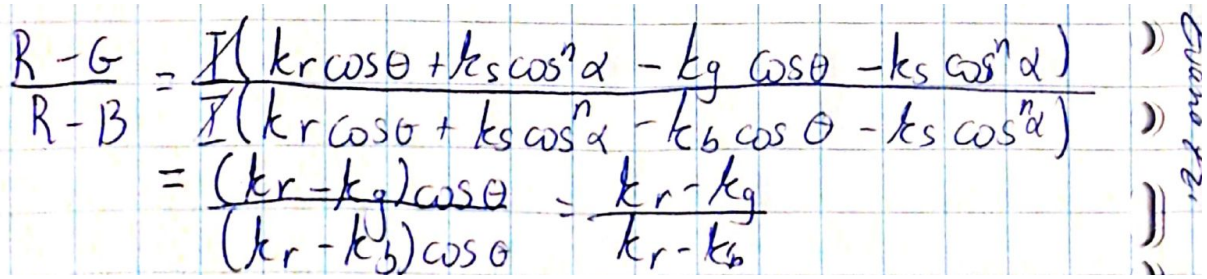
g)
Prove that R/G is not a color invariant for shiny surfaces. Prove that (R−G)/(R−B) is a color invariant for shiny surfaces.

$R/G = (R = I k_R \cos \theta + I k_s \cos^n \alpha) / (G = I k_G \cos \theta + I k_s \cos^n \alpha)$
$= (R = k_R \cos \theta + k_s \cos^n \alpha) / (G = k_G \cos \theta + k_s \cos^n \alpha)$

It is not invariant to light source direction and view point. In other words, it still depends on theta (light source direction) and alpha (view point).



(R−G)/(R−B) = (I k_R cos θ - I k_G cos θ) / (I k_R cos θ - I k_B cos θ)
= (I theta (k_R - k_G)) / (I theta (k_R - k_B))
= (k_R - k_G) / (k_R - k_B)



This only depends on the reflectance properties k, and is therefore color invariant, as it no longer depends on light source intensity I or reflection angle theta.

Exercises (Lecture 3)

Exercise 1. Image Filtering

a)
Which of the following would make an image blurrier, a 3x3 or a5x5 uniform filter? Why?

3x3 uniform (box) filter:

$$T = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

5x5 uniform (box) filter:

$$U = \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

A 5x5 uniform filter. Average over a larger area.

b)
What kind of image features are detected by the 3x3 edge filters V and W?

Another 3x3 filter

$$V = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

The Laplacian using the following matrix values:

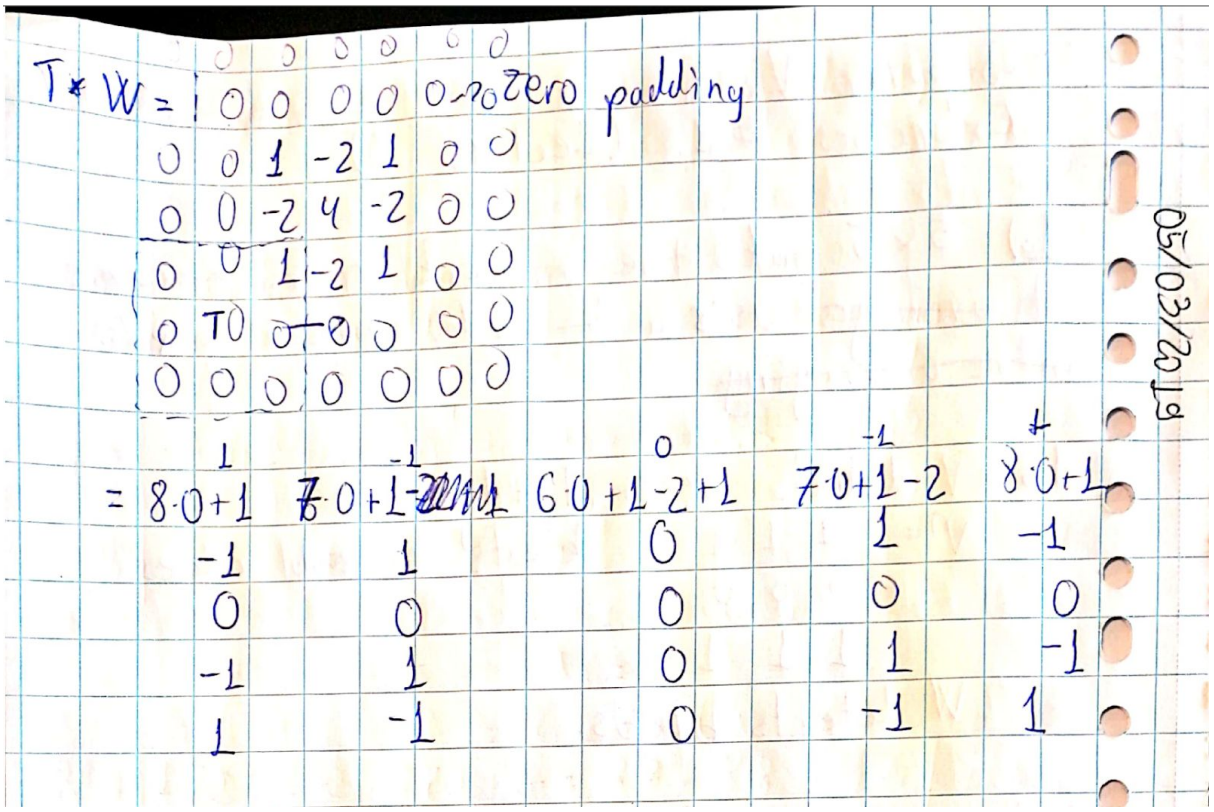
$$W = \begin{array}{|c|c|c|}\hline 1 & -2 & 1 \\ \hline -2 & 4 & -2 \\ \hline 1 & -2 & 1 \\ \hline \end{array}$$

V: vertical edges (kernel symmetrical in one direction)

W: blobs (kernel symmetrical in two directions)

c)

You wish to transform an image by applying a 3x3 uniform filter followed by the 3x3 Laplacian filter W. Show that this can be implemented by a single a 5x5 filter and calculate the elements of this filter.



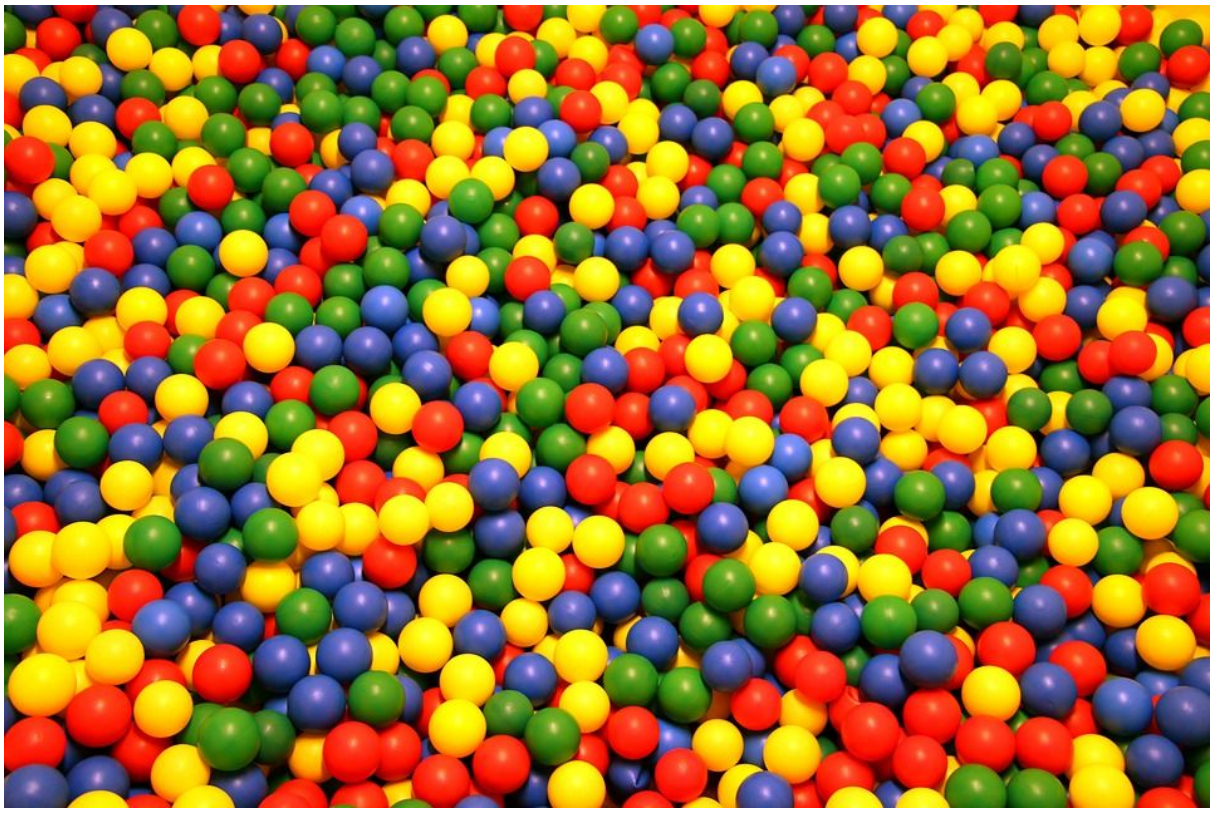
$$W = \begin{bmatrix} 1 & -1 & 0 & -1 & 1 \\ -1 & 1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & -1 \\ 1 & -1 & 0 & -1 & 1 \end{bmatrix}$$

Exercise 2. Color Constancy

a)

Give an example of an image for which the white patch method will fail. Please explain.

An image containing a few colorful (object) colors. The assumption is that an achromatic patch is present in the image.



(it is not that easy to find an image without white!)
Another case where the assumption goes wrong is for images where the only white patch is under-exposed to light (it is captured as some gray shade) and there is a colorful patch which is brighter than the white one.

Similar for the gray world assumption. It assumes that the average of all colors in an image is gray (128), however some images will have their average color far from gray, using the gray world assumption will de-color most of the image.

Also, if there is an image of just one color (discussing corner case), it will be turned to grey by the grey world assumption.

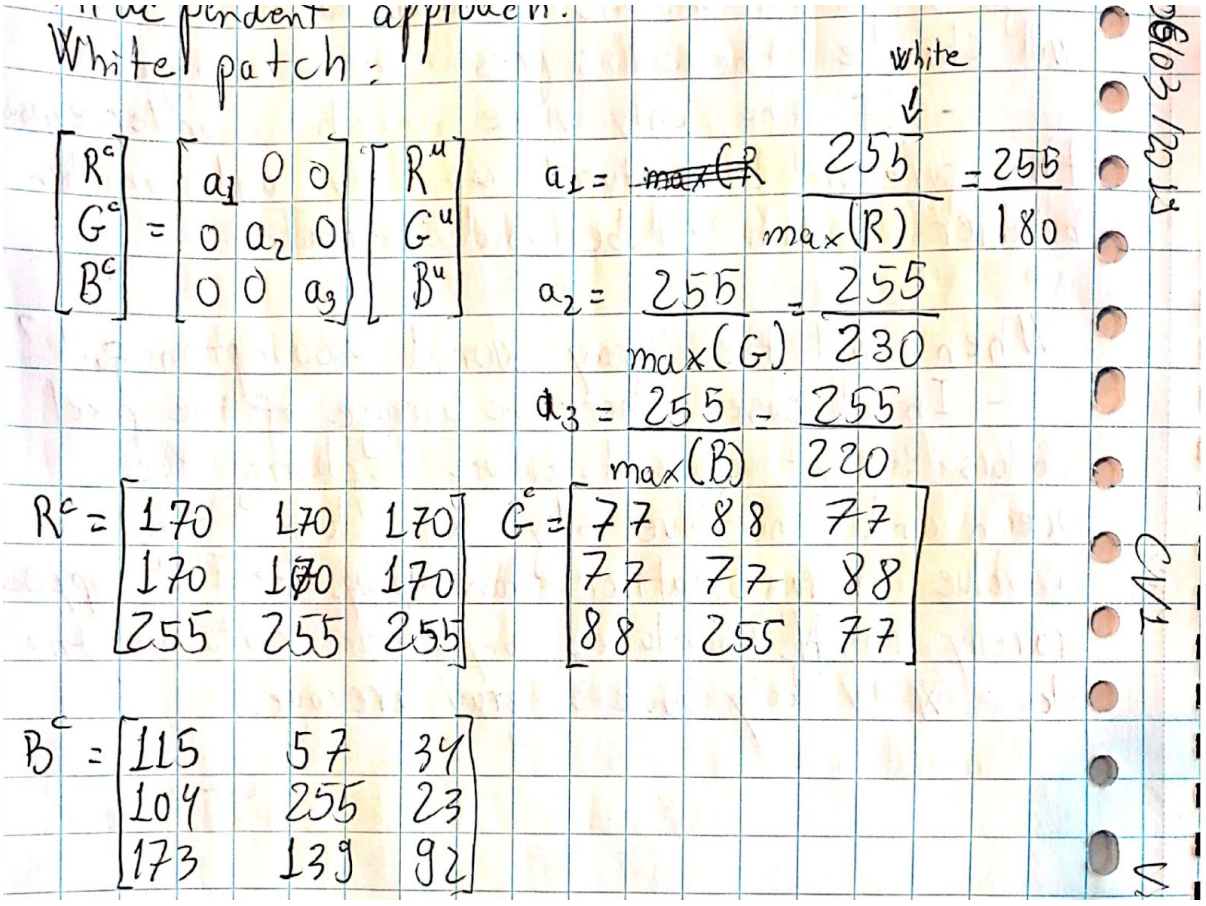
b)
Calculate the results of both algorithms for the image shown in Table 1.

$$R = \begin{array}{|c|c|c|}\hline 120 & 120 & 120 \\ \hline 120 & 120 & 120 \\ \hline 180 & 180 & 180 \\ \hline \end{array}$$

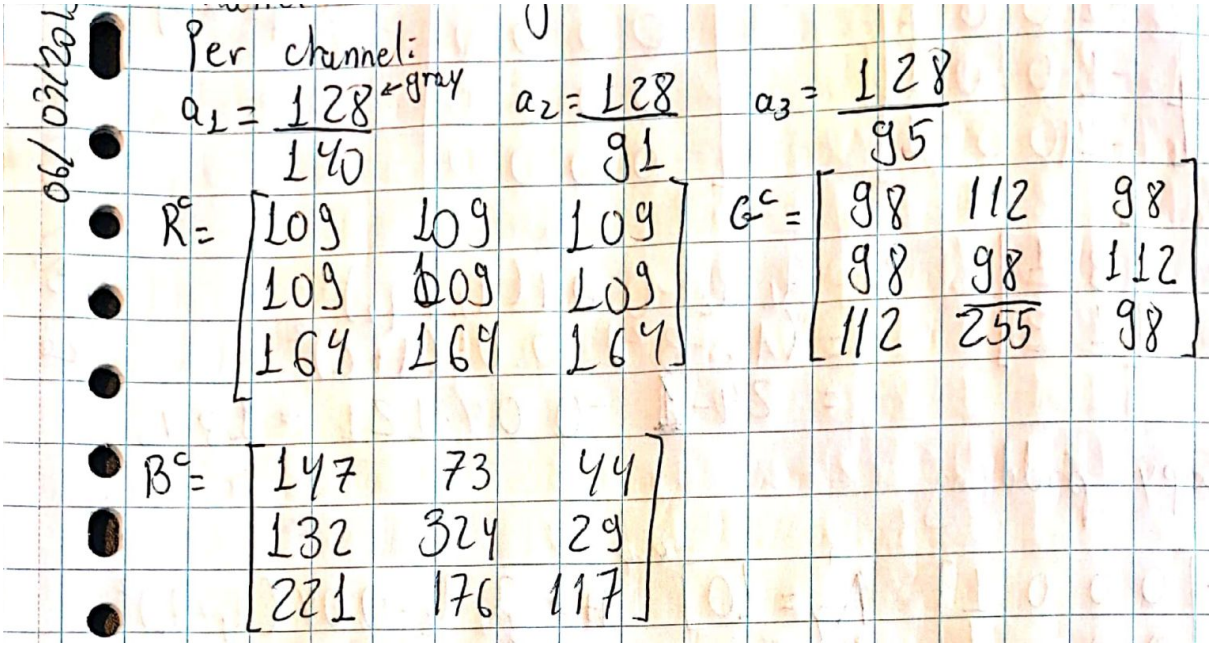
$$G = \begin{array}{|c|c|c|}\hline 70 & 80 & 70 \\ \hline 70 & 70 & 80 \\ \hline 80 & 230 & 70 \\ \hline \end{array}$$

$$B = \begin{array}{|c|c|c|}\hline 100 & 50 & 30 \\ \hline 90 & 220 & 20 \\ \hline 150 & 120 & 80 \\ \hline \end{array}$$

In the white-patch method R values get scaled to white by separately taking the pixel with max intensity for that color, like 255/max(R), then similar for G/B.
White patch: a1=255/180, a2=255/230, a3=255/220.



In the grey-world method R values get scaled to grey like 128/mean(R), similar for G/B.
Grey-world: a1=128/140, a2=128/91, a3=128/95.4.



Adding to this, there are two methods to calculate the patches. The first one is taking each channel independent (as is done above), the second is calculating it for the whole color space RGB (for the white patch it would be [180, 230, 220]).

Exercise 3. Edges and Corners

$A =$

1	1	1	0
1	1	1	0
1	1	1	0
1	1	1	0

$B =$

0	0	0	0
1	1	1	0
1	1	1	0
1	1	1	0

Table 2: Intensity values of image patches A and B.

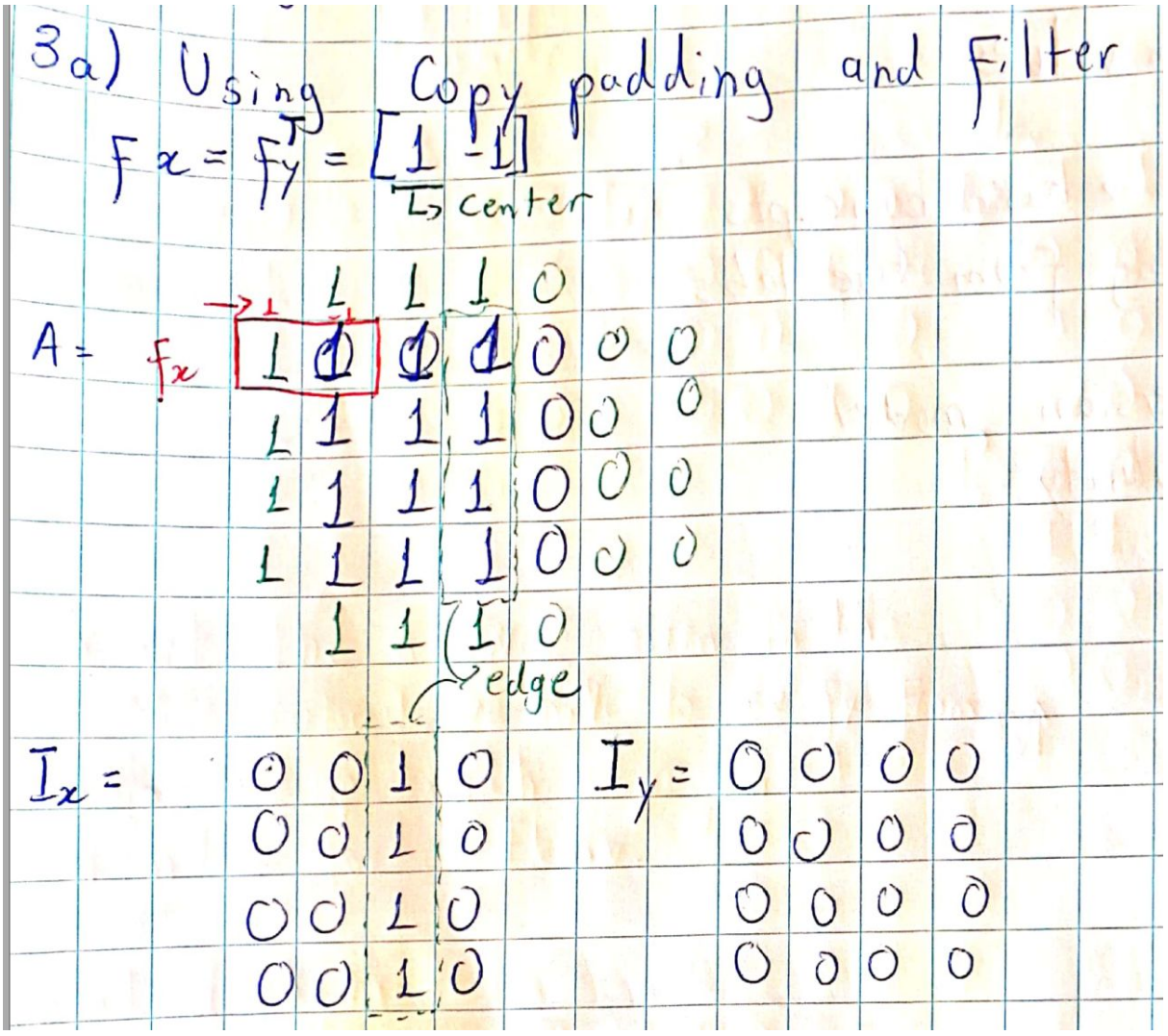
a)
Compute the gradient magnitude and the Harris corner response of image patch A (using a simple derivative filter e.g. [1-1]).

(A.a) Gradient: $\nabla f = \sqrt{f_x^2 + f_y^2} =$

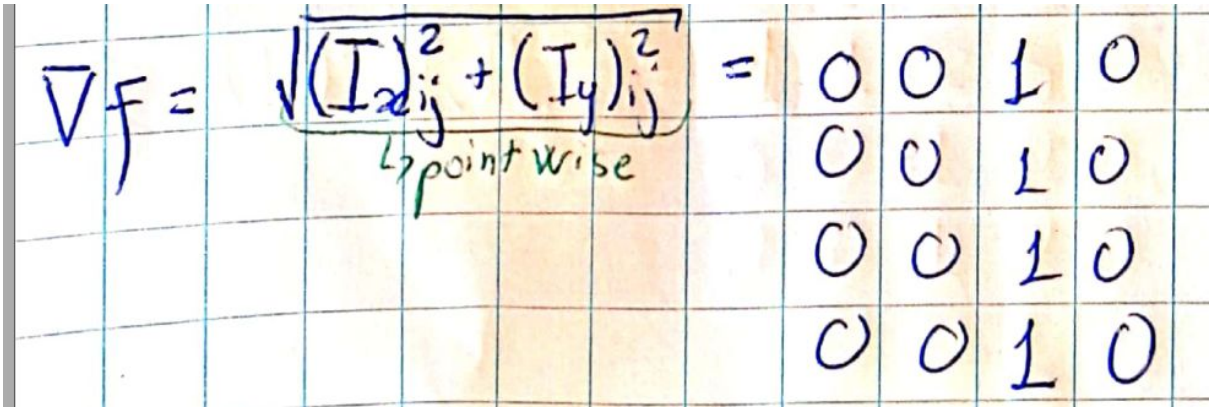
0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0

Harris corner response $R = \mathrm{Det}M - k(\mathrm{Trace}(M))^2 = -0.64$ where $M = \begin{bmatrix} 4 & 0 \\ 0 & 0 \end{bmatrix}$.

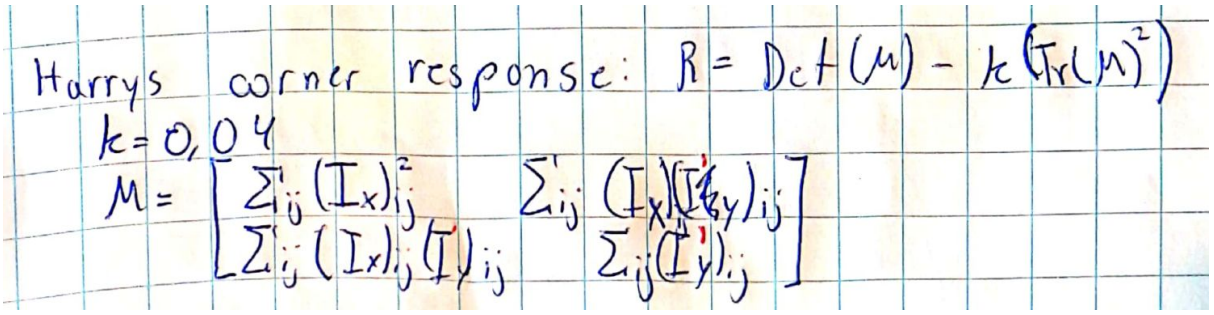
k is an empirical constant, usually from 0.04-0.06.

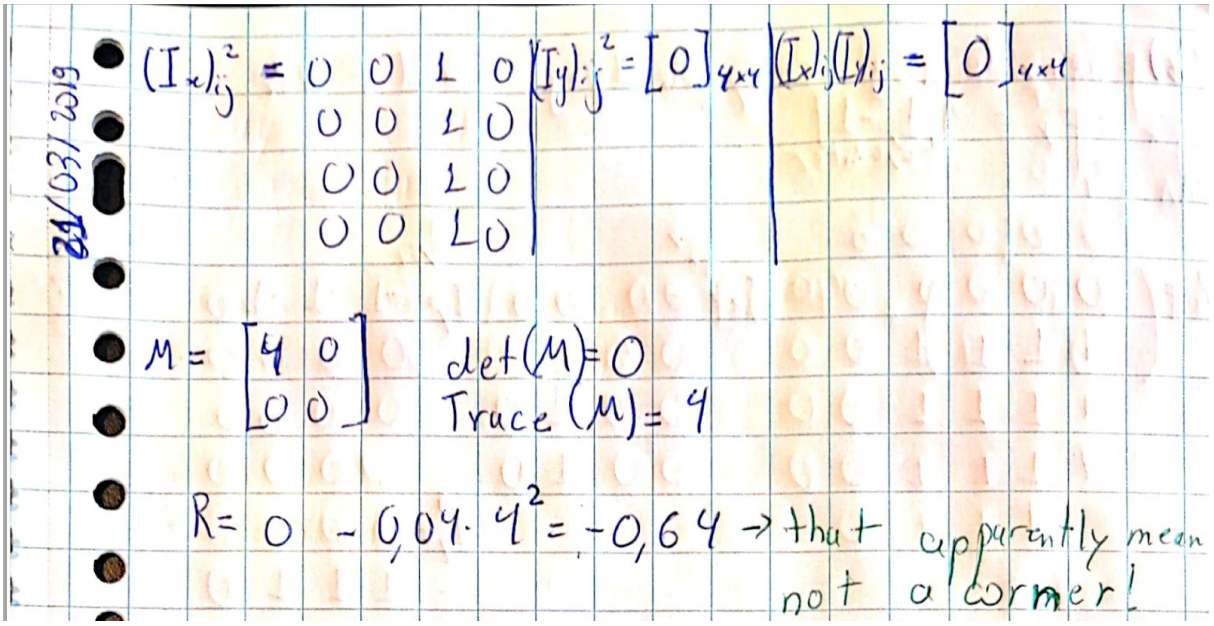


Gradient Magnitude:



Harry's corner response:





I think that UvA should donate the part of our tuition fee dedicated to pay the teaching hours of Theo to wikipedia: https://en.wikipedia.org/wiki/Image_gradient

b)

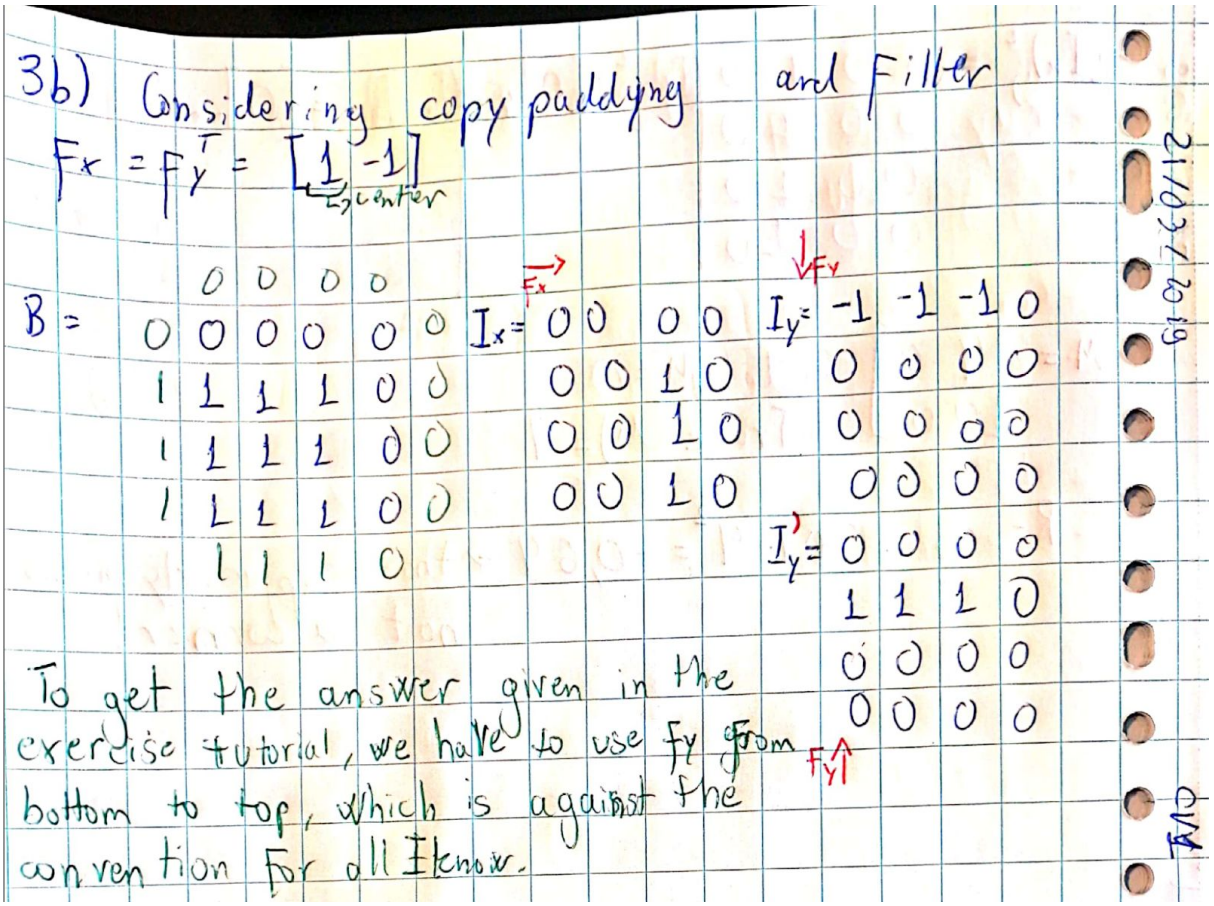
What are the interest points of image patch B?

Edges and corners.

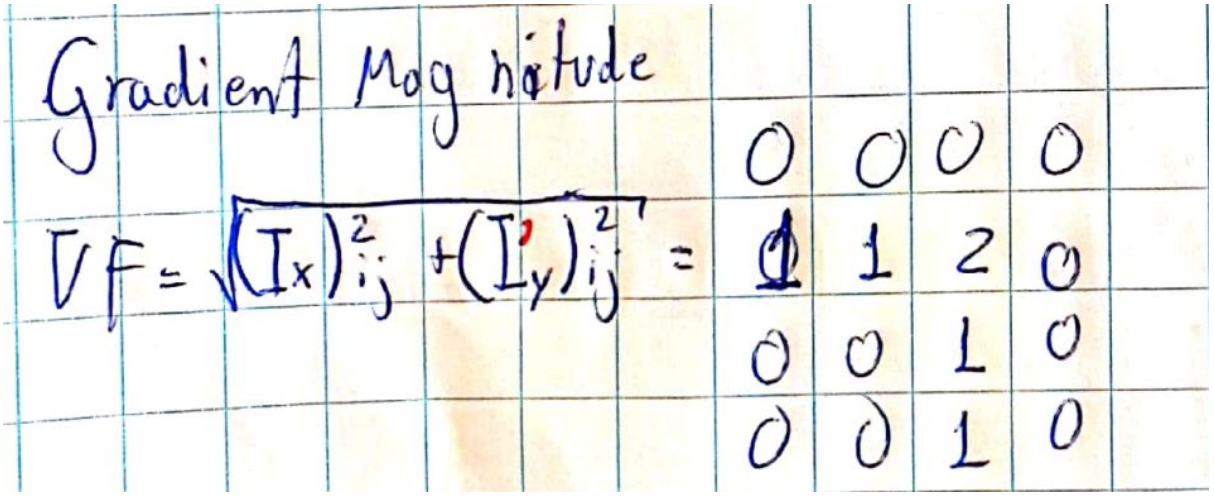
c)

Compute the gradient magnitude and the Harris corner response of image patch B (using a simple edge filter e.g. [1,-1]).

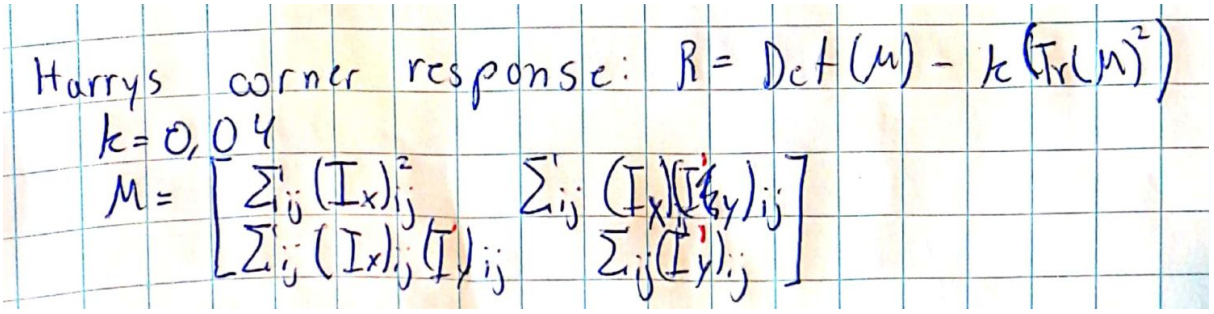
M = [Σij (Ix)^2_ij, Σij (Ix)(Iy)_ij; Σij (Ix)(Iy)_ij, Σij (Iy)^2_ij]
M=[3 1; 1 3] and hence R = 6.56 which is >0 and therefore a corner.

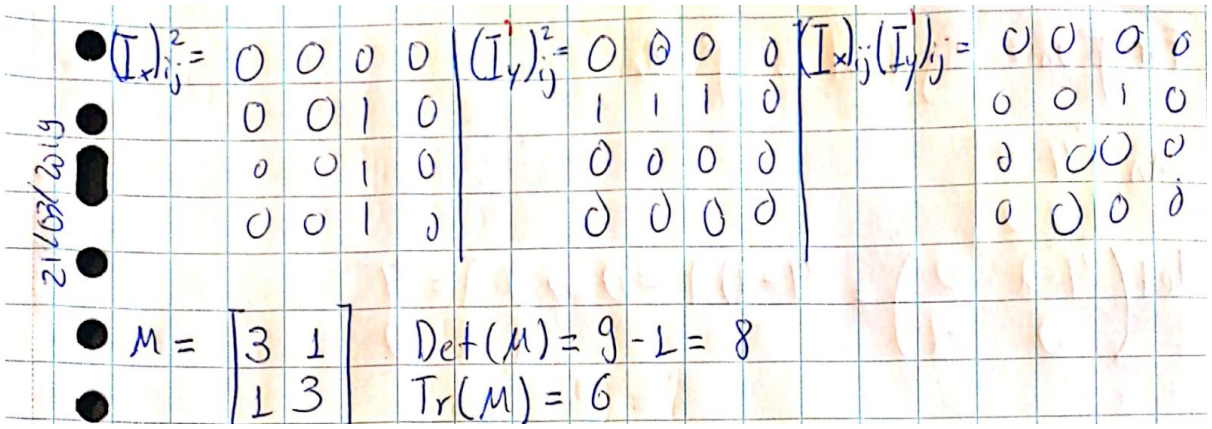


Gradient Magnitude:



Harry's corner response:





d)
Compute the eigenvalues of M for patch B where M is the 2x2 matrix computed from the image derivatives i.e. second moment matrix (autocorrelation matrix).

Eigenvalues are λ1=4 and λ2=2.

3 c) Eigen values of M_{13}

$$M_{13} = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$$

How to Find eigen values? → They are very shy so just calling them won't work!

$Mx = \lambda x$ → x eigen vector, λ eigen values

$Mx - \lambda x = 0$

$\det(M - \lambda I) = 0$

$$\det\left(\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = 0$$

$$\det\begin{pmatrix} 3-\lambda & 1 \\ 1 & 3-\lambda \end{pmatrix} = 0 \Rightarrow (3-\lambda)^2 - 1 = 0$$

$$\lambda^2 - 6\lambda + 8 = 0$$

Bhaskara: $\lambda = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{6 \pm \sqrt{36-32}}{2} = \begin{cases} \lambda_1 = 4 \\ \lambda_2 = 2 \end{cases}$

For Fun: eigen values of M_A

$$M_A = \begin{bmatrix} 4 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\det\begin{pmatrix} 4-\lambda & 0 \\ 0 & -\lambda \end{pmatrix} = (4-\lambda)\lambda = 0 \Rightarrow \begin{cases} \lambda = 4 \\ \lambda = 0 \end{cases}$$

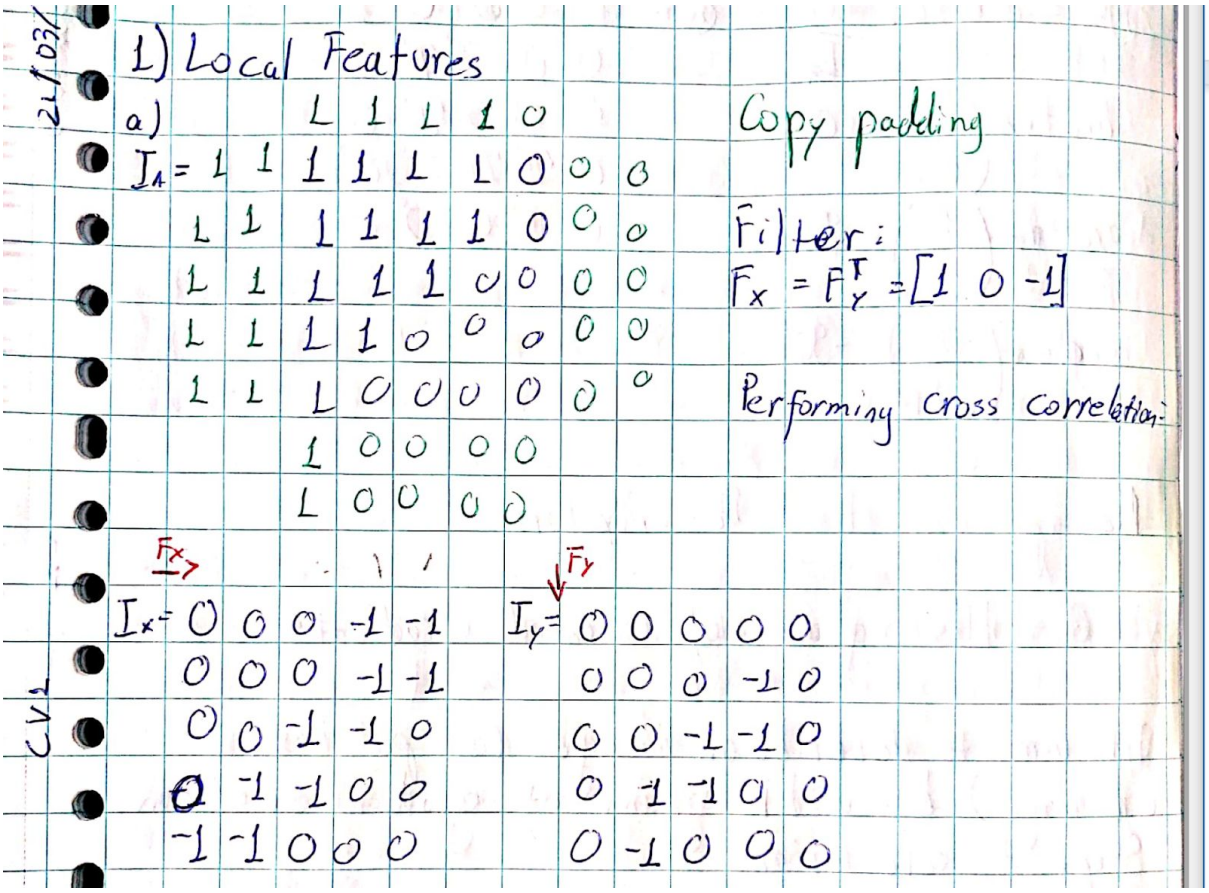
Exercises (Lecture 4: Bag of Words)

Exercise 1. Local Features

$$I_A = \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

a)
Compute the gradient Gx and Gy, using image filters. Which filters do you use?

Fx = [1, 0, -1] and Fy = Fx^T
Gx = [
 0 0 0 1 1
 0 0 0 1 1
 0 0 1 1 0
 0 1 1 0 0
 1 1 0 0 0
]
Gy = [
 0 0 0 0 0
 0 0 0 1 0
 0 0 1 1 0
 0 1 1 0 0
 0 1 0 0 0
]



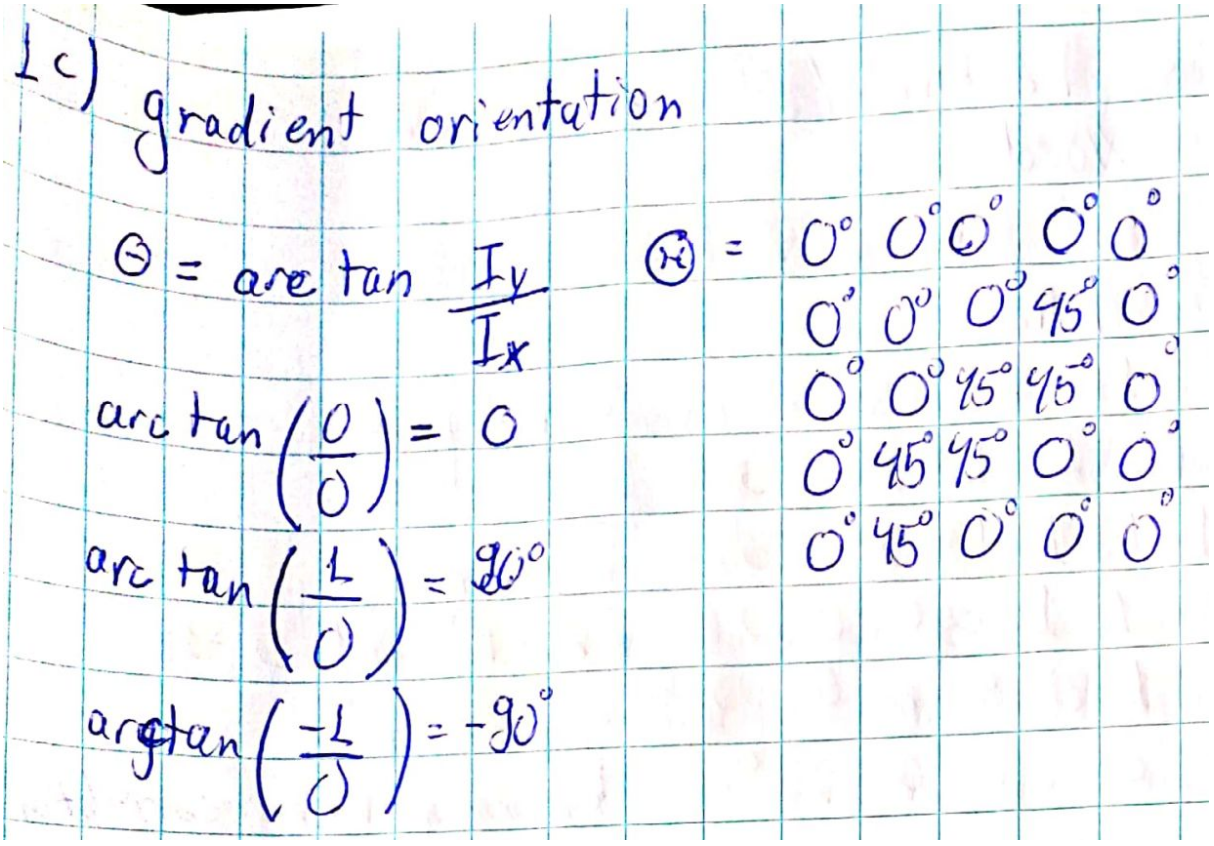
b)
Compute the gradient magnitude

∇f = sqrt((Gx)^2 .+ (Gy)^2) = [
0 0 0 1 1
0 0 0 √2 1
0 0 √2 √2 0
0 √2 √2 0 0
1 √2 0 0 0
]

c)
Compute the gradient orientation (in degrees)

theta = arctan(Iy/Ix)(180/pi), using assumptions: 0/0 = 0, 1/0 = inf, -1/0 = -inf
= 180/pi * arctan [
0 0 0 0 0
0 0 0 1 0
0 0 1 1 0
0 1 1 0 0
0 1 0 0 0
]
= 180/pi * [

0 0 0 0 0
0 0 0 π/4 0
0 0 π/4 π/4 0
0 π/4 π/4 0 0
0 π/4 0 0 0
]
= [
0 0 0 0 0
0 0 0 45 0
0 0 45 45 0
0 45 45 0 0
0 45 0 0 0
]



d)

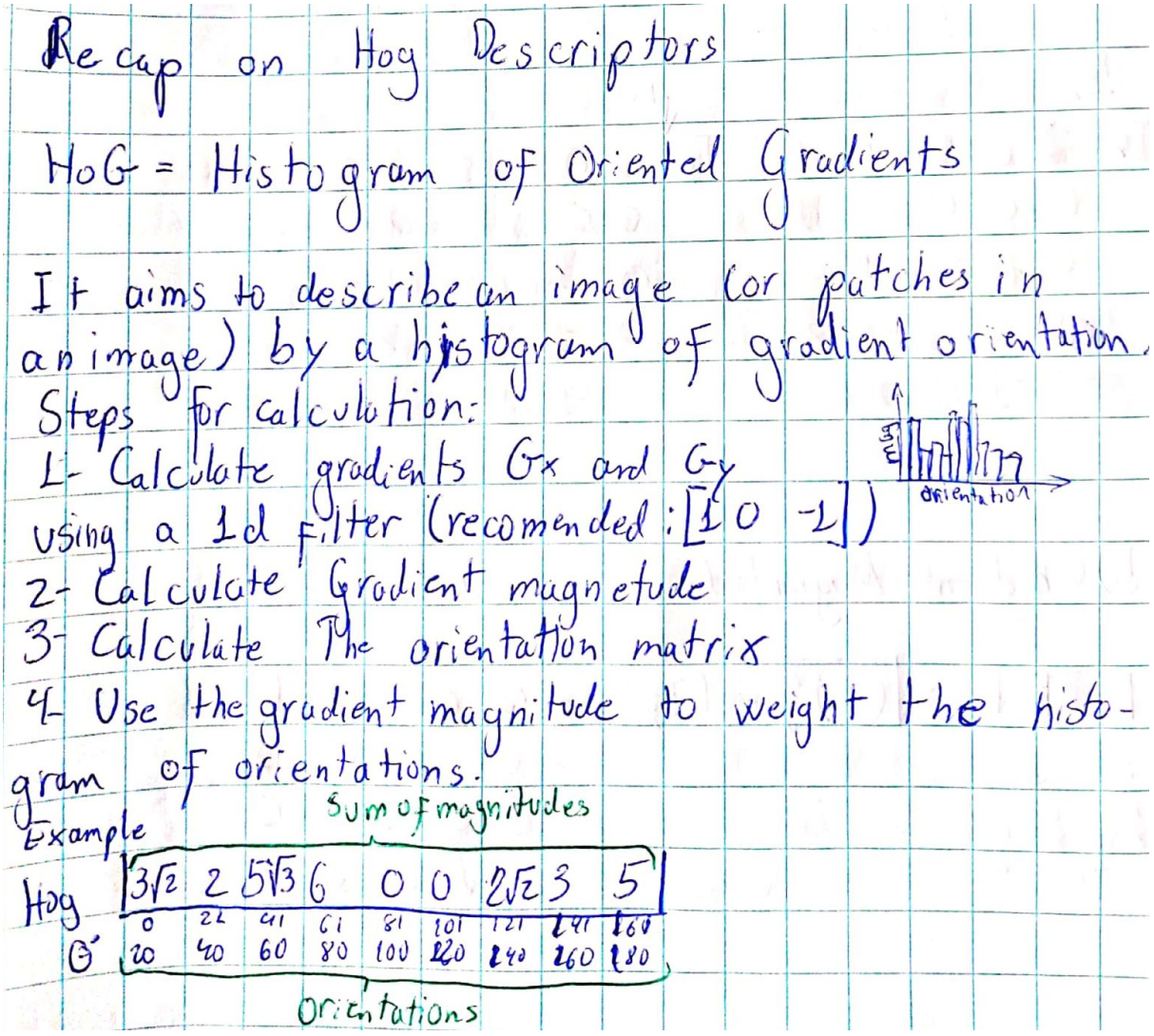
Compute the Histogram of gradients (HoG) descriptor, using a 9 bin histogram

Combine the direction of the gradient with its magnitude. So you can differentiate between a "0" because of no gradient and a "0" because of a vertical gradient. For real HoG descriptors some more tricks are performed, including *unsigned gradients* (using 0 - 180 only), and sharing gradients over bins (i.e. 30 degree will count in bin of 0 and bin of 40), these fall beyond the scope of this exercise. See also tutorial.

1d) HoG using 9 bins

HoG	4	0	$6\sqrt{2}$	0	0	0	0	0	0	(not normalized)
degrees	20	40	60	80	100	120	140	160	180	

The bins are ranges of values, [0,20], (20, 40], (40,60]....(160,180]
The HoG values are the sums of the magnitudes (hence the √2 here).



e)

Is the HoG descriptor invariant to overall lighting, i.e., is the HoG descriptor of I = 2∗IA equal to the HoG descriptor of IA?

No, It is invariant to additive color (I = I_A + c), but not its scaling. For more details see the nice hands-on tutorial on https://www.learnopencv.com/histogram-of-oriented-gradients/
If we scale the image, I=2*I, then the gradient will change. Magnitude and direction depend on gradient, so those will change as well → HoG not invariant to overall lighting.

When normalized the HoG descriptor is invariant to light intensity, color addition and rotation. When it is not normalized it is still variant to color addition and rotation.

Exercise 2. Bag-of-Words

2.a
Describe the basic steps of the Bag-of-Visual Words model?

Offline stage:

- compute BoW vocabulary (see 2.e)

For each image:

1. Sample patches;
2. Compute the descriptor for each patch;
3. Assign to closest word in the vocabulary.

1. From an image set, collect **descriptors** of each image.
2. Use those descriptors to build a **vocabulary**. For such purposes a **clustering** technique is used so that similar descriptors are clustered together and then a visual word is represented by the centroid of each cluster. The number of clusters defines the **size** of the vocabulary. k-means is often used for the clustering.
3. The visual words are used to describe the images in a **BoW** fashion. From the target image, descriptors are taken, those descriptors are assigned to clusters and the image is finally represented as the number of visual words it contains for each visual word (visual word = centroid of a cluster). Essentially building a **histogram-feature** of visual words for each target image.
4. Use those histogram-features and corresponding image labels to train your model using SVM or other similar classifiers.

2.b
What is the difference between dense sampling and interest point sampling?

- **Interest points:** find interest points on salient parts of the image (e.g. using Harris). Advantage: focuses on salient parts of the image/object.
- **Dense (multi-scale) sampling:** ensures you have an even number of sampled patches per image, on any place in the image, as finding high quality/good interest points is difficult.

Key Points:
The descriptors of an image are taken when they exceed a descriptiveness threshold.
Those descriptors have different scales.
Different images may have a different number of key point descriptors. More detailed images should present more descriptors than images with less details.
It is expected that most of the descriptors rover on the objects of the image. However it is possible that very detailed backgrounds also get a large number of descriptors.

Dense Sampling:

The descriptors are sampled evenly from the images.
Every image has the same number of descriptors (given that they all have the same size).
Descriptors are all of the same scale, unless dense sampling with multiple scales is done. It usually yields a higher number of descriptors per image.

2.c
Assume we have an image retrieval system, with 5000 images, 100 query images, and we use a BoW representation with 10K words, using SIFT descriptors. We observe that a BoW with dense-sampled patches outperform BoW with interest points patches, when using precision@10 as evaluation measure. What could be the reason?

Only a very few images, so possibly the relevant images do not have enough interest points to get stable descriptors.

2.d
Now we increase the dataset to 5M images, and interest points perform better. What could be the reason?

Precision is more important than recall (in this evaluation measure), so with the much larger dataset, the interest points could be able to find better matches (i.e. object only).

2.e
Explain how k-Means clustering can be used to obtain a visual vocabulary.

1. **Sample** a large set of patches from the train set (1M)
2. For each patch get the **descriptor**
3. Run **K-Means** over this set of descriptors
4. The resulting **means** are your visual words
5. The value of k is a **hyperparameter**
6. In practice it works better to **compare** a few random initialisations with only one or two iterations of k-means, than have a single random initialisation and run to convergence

Exercise 3. Retrieval

3.a For retrieval, each image is described with 1000 interest points, each interest point is 128 dimensional. When finding matches between 2 images, how many computations are required?

- **compare** each interest point in image 1 with each interest point in image 2
- each comparison takes 128 computations
- total of 1000 * 1000 * 128 = 128M **computations**

3.b Comparing 1M interest points, takes 1 second. How long does it take to compare an image with a dataset containing 1M images?

- Note here then granularity is ”comparing interest points”
- So, comparing two images takes 1 second (see question above).
- Answer: 1M images * 1M interest points / 1M interest points per second = 1M seconds (not taking into account the sorting).

3.c After retrieving results with BoW, we use geometrical verification to rerank the top 100 images. However, our evaluation shows no difference in precision and recall measured at k=100. Why?

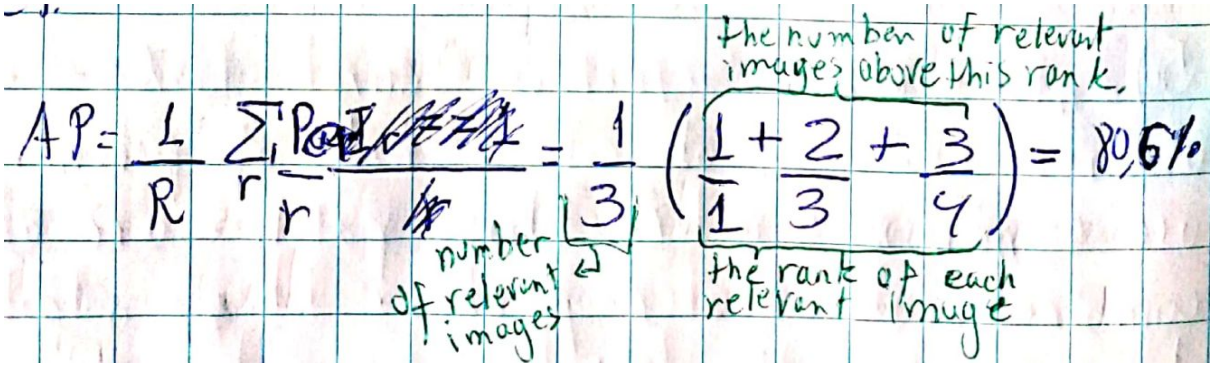
We evaluate the set of 100 images for retrieval and recall, re-ordering these would not change precision and recall at 100. It could improve e.g. precision and recall at 10.

3.d Explain why accuracy is not a good metric

Accuracy is defined as the average number of ”correct assignments”; Both relevant and non-relevant images are taken into account. Given that for retrieval most images are not relevant, always returning ”no image at all” give a very high accuracy, but is not a sensible retrieval system.

3.e Compute Average Precision for the following relevance ranking: [R, N, R, R, N], where R denotes relevant, and N not-relevant

AP = 1/R * ∑_r P(r) ∗ R(r),
= 1/3 (1/1 + 2/3 + 3/4) ≈0.80,
where R is the total number of relevant documents (3), P(r) is the precision at rank r (1, 1/2, 2/3, 3/4, 3/5), and R(r) is the relevance of the document at rank r (1, 0, 1, 1, 0).



Exercise 4. Classification

a)
Explain how to train a ”cat” vs ”non-cat” classifier using linear classification (i.e. SVMs)

Answer:
- **collect** a large dataset of annotated images
- **split** into train, validation, and test set
- compute **BoW vocabulary** over train and validation

- compute **BoW representation** of all images (train, val and test)
- use train set to **train** your favourite (linear) classifier, use val set to select hyperparameters (number of words,regularisation, etc).
- **evaluate** the performance of your classifier on the test set.

b)

What is the loss and the gradient of the loss for binary SVM / logistic regression, when using F(x) = w^T x

For binary SVMs, using yi={−1,+1}, we have (without regularisation) Ω(w):

$$\mathcal{L} = \sum_{i=0}^{n} ||1 - y_i \boldsymbol{w}^\top \boldsymbol{x}_i||_+$$

$$\nabla_w \mathcal{L} = -\sum_i [\![y_i \boldsymbol{w}^\top \boldsymbol{x}_i < 1]\!] \, y_i \boldsymbol{x}_i$$

c)

What is the loss and the gradient of the loss for multi-class SVM / logistic regression, when using F_c(x) = w_c^T x

Answer: Consider the softmax loss (or multi-class logistic loss, or the cross-entropy loss):

$$p(c|\boldsymbol{x}_i) = \frac{1}{Z_{x_i}} \exp(\boldsymbol{w}_c^\top \boldsymbol{x}_i), \qquad Z_{x_i} = \sum_{c'} \exp(\boldsymbol{w}_{c'}^\top \boldsymbol{x}_i) \tag{7}$$

$$\mathcal{L} = -\sum_i \log p(c = y_i|\boldsymbol{x}_i) = -\sum_i \log \frac{1}{Z_{x_i}} \exp(\boldsymbol{w}_{y_i}^\top \boldsymbol{x}_i), \tag{8}$$

$$\nabla_{w_c} \mathcal{L} = \sum_i \Big(p(c|\boldsymbol{x}_i) - [\![y_i = c]\!] \Big) \boldsymbol{x}_i, \tag{9}$$

where $[\![z]\!]$ equals 1 if z is true and 0 otherwise. **Note:** the derivation is with respect to w_c.
Longer derivation:

$$\max \sum_i \log p(c = y_i|\boldsymbol{x}_i) = \min - \sum_i \log p(c = y_i|\boldsymbol{x}_i)$$

$$\mathcal{L} = -\sum_i \log p(y_i|\boldsymbol{x}_i) = \sum_i \log \left(\frac{1}{Z_{x_i}} \exp(\boldsymbol{w}_{y_i}^\top \boldsymbol{x}_i) \right),$$

$$\nabla_{w_c} \mathcal{L} = -\sum_i \frac{1}{p(y_i|\boldsymbol{x}_i)} \left[\frac{1}{Z_{x_i}} \exp(\boldsymbol{w}_{y_i}^\top \boldsymbol{x}_i) \, \boldsymbol{x}_i [\![y_i = c]\!] - \frac{1}{Z_{x_i}} \exp(\boldsymbol{w}_{y_i}^\top \boldsymbol{x}_i) \frac{1}{Z_{x_i}} \sum_{c'} \left(\exp(\boldsymbol{w}_c^\top \boldsymbol{x}_i) \boldsymbol{x}_i [\![c' = c]\!] \right) \right]$$

$$= -\sum_i \frac{1}{p(y_i|\boldsymbol{x}_i)} \left[p(y_i|\boldsymbol{x}_i) \, \boldsymbol{x}_i \, [\![y_i = c]\!] - p(y_i|\boldsymbol{x}_i) \frac{1}{Z_{x_i}} \sum_{c'} \left(\exp(\boldsymbol{w}_c^\top \boldsymbol{x}_i) \, \boldsymbol{x}_i \, [c' = c] \right) \right]$$

$$= -\sum_i \frac{1}{p(y_i|\boldsymbol{x}_i)} \Big[p(y_i|\boldsymbol{x}_i) \, \boldsymbol{x}_i \, [\![y_i = c]\!] - p(y_i|\boldsymbol{x}_i) p(c|\boldsymbol{x}_i) \, \boldsymbol{x}_i \Big]$$

$$= -\sum_i \Big[[\![y_i = c]\!] - p(c|\boldsymbol{x}_i) \Big] \, \boldsymbol{x}_i$$

Exercise 5. Object Detection

Good reading:
https://www.learnopencv.com/selective-search-for-object-detection-cpp-python/
http://www.huppelen.nl/publications/selectiveSearchDraft.pdf

a)
Compute the number of box evaluations for a single image in a multi-class object detection problem

total number of evaluations: #locations × #aspect ratios × #scales × #classes

The goal of the Object Detection is to find a bounding box such that is the tightest around the detected object without leaving out any part of it. Well, some objects have about the same ratio (x/y) for all perspectives (it does not matter from where you take a picture of a ball, its aspect ratio should always be 1). However, some objects can have nearly infinite number of aspect ratios (the cows in the slides are a good example).
If the method we use for Object Detection is windowing, then, to capture all those possible aspect ratios from the objects, we need to specify windows of different aspect ratios as well.

That is just to say that windowing for object detection is a bad idea!

b)
Explain the idea of Selective Search?

Find a (small) set of class agnostic object bounding boxes.

Selective Search segments the image into many small regions, each region possibly contains one object and is stored as a bounding box for later analysis. Those segments are then aggregated by their similarity (a few similarity metrics are used) and regions corresponding to these larger segments are proposed/stored as bounding boxes. This is repeated multiple times to obtain bounding boxes of various sizes (hierarchical).
This method ensures very high recall, it is very likely that an object will be captured in one of those images, the bounding boxes will be tight around the object and is cheap to run (cheaper than other methods). The number of boxes is smaller than simply windowing and the quality of the boxes is also superior.

c)
How does Selective Search increases the variations?

At least by using hierarchical clustering and using different color spaces

Selective Search is robust to many image variations because it uses many different color channels and similarity measures, such as texture and size of the segments.

Tutorial Lecture 5: ConvNets & GANs

1 Conv Layer Arithmetics

Good read: http://cs231n.github.io/convolutional-networks/
Good video: https://www.youtube.com/watch?v=jajksuQW4mc

1.a How many parameters are needed for a fully connected layer, when W=H (height) = 100, and H (hidden)=1000?11

W × H (height) × H (hidden) = 10M
“hidden” here corresponds to FC layer.

1.b How many parameters are needed for a locally constrained layer, where each neuron looks at a 10x10 window, when using W=H=100, and stride of 5?

Hidden = ((W - W_window) / stride + 1) * ((H - H_window) / stride + 1) = (100-10)/5+1 * (100-10)/5+1 = 19 × 19 = 361, so 10x10x361 = 36K

1.c How many parameters are needed when a convolutional layer is used, using a filter of size 10x10, and using 100 different filters?

Convolutional layers do weight sharing. Which means that the same filter is applied on the entire image. This causes a huge decrease in number of parameters necessary to train.
filter height * filter width * filters = 10 × 10 × 100 = 10K
Answer above: you’re forgettings biases; (filter_height * filter_width + bias) * n_filters = (10*10 + 1) * 100 = 10.1K

1.d What is the size of the output volume with stride 1?

Assuming we don’t do padding, the output is (32 − 5)/Stride + 1 = 28, so 28 × 28 × 5

1.e How many weights are learned?

5 × 5 × 3 = 75 per filter 375 in total.

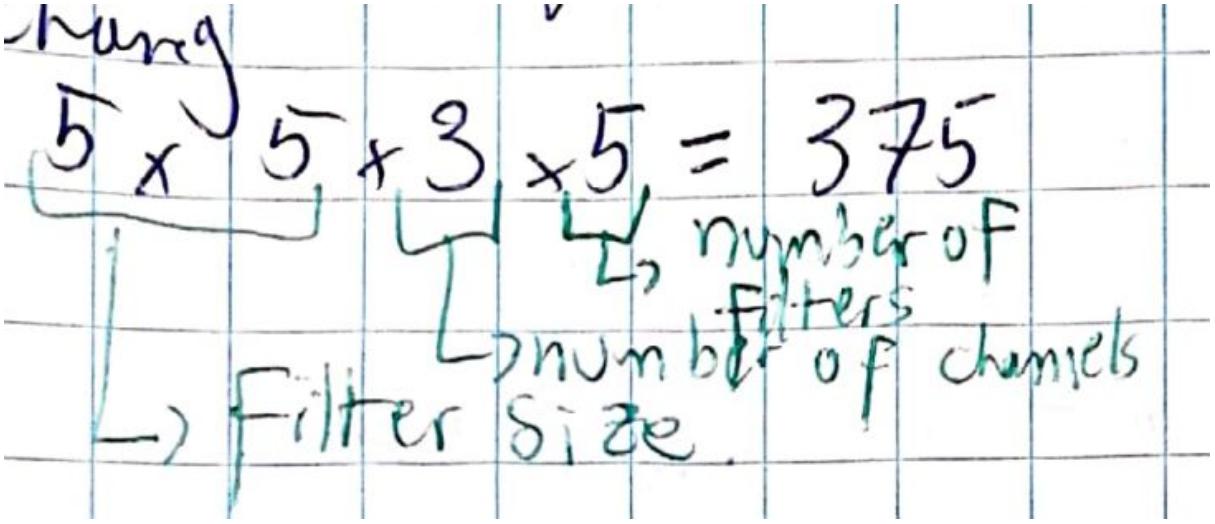
1.f What is the depth of a filter in the next layer?

5, because there are 5 filters.

1.g What is the size of the output volume with stride 3?

We now have:
$w_{out} = h_{out} = \frac{32-5}{3} + 1 = 10$
so 10 × 10 × 5

1.h How many weights are learned in that case?
Because it is a convolution layer, the number of weights does not change.



2 Sharing of Weights

2.a Describe a scenario where weight sharing -as is done in the Convolutional Layer - is not beneficial for recognition or training.

When images are registered, for example for face identification, you want to have an ”eye”-neuron, and a ”mouth”-neuron.

When inputs are not ordered.

Convolution is necessary in order to be translation invariant. If the dataset is static (the features are always represented in the same spot, like in a dataframe) then convolution is not really the way to go. If there is uncertainty about the position of the features, then convolutions can be a good way to build robustness against that. In passport images, the face features are always in the same regions, but not always include the same pixels, that is not ideal for a fully connected layer, but it may work at the cost of having many more parameters than a convolutional layer would.

3 Gradients

3.a What is the difference between the analytical gradient and the numerical gradient?

TensorFlow/PyTorch etc. use numerical gradients for gradient descent. Analytical gradient is when I + δ is used and compared to I.
[This answer seems to be wrong! My interpretation:
Analytical method relies on calculating the exact gradient using calculus and is computationally quicker, but error prone thus requiring careful implementation and also gradient check. Many ML libraries like Tensorflow and PyTorch use Analytical implementation.
Numerical method is an approximate way of calculating the gradient using Taylor series expansion or Newton-Raphson techniques and so on. There are plenty of such techniques and are usually computationally expensive to get a good approximation, but are easy to implement.]
Check: http://cs231n.github.io/optimization-1/

3.b Consider the following conv notation: a_rc = x^⊤_r θ, what is the gradient wrt the parameters?

That is x_r, however, for the full gradient, it needs to be summed over all regions, including the loss of each region from the layer above.

3.c Consider the ReLu non-linearity z = max(0, a), what is the gradient?

The gradient is

$\nabla z =$ {
1, if a > 0.
0, if a < 0.
undefined if a = 0
}.

3.d A ReLU is considered to be 'dead' if it never updates. Describe (a) when this happens, and (b) one method to circumvent dead ReLUs

a When the input data of the neuron is always negative, it will never receive a gradient.
b Use Leaky-ReLu's or initialise ReLu's with some positive bias

3.e Describe the max-pooling layer, and the gradient of the layer

It is a form of quantisation, it selects the maximum value of the region and passes that through to the next layer. The gradient is

$\nabla iz =$
{1, if argmax = i,
0, otherwise.

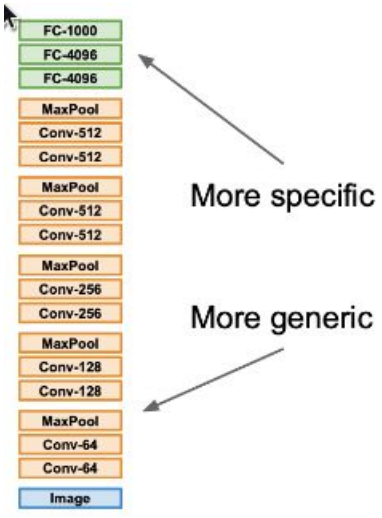
In other words, it only passes through the ”pixel” with the maximum of the receptive field of the max-filter.

Additional explanation:
https://datascience.stackexchange.com/questions/11699/backprop-through-max-pooling-layers

4 Transfer Learning and Invariances

		very similar target data	very different target data
4.a	very little target train data	A	C
	quite a lot target train data	B	D

Please give advice about learning scheme for a ConvNet scheme for the 4 scenarios (A-D) above.



	very similar dataset	very different dataset
very little data	Use Linear Classifier on top layer	You're in trouble… Try linear classifier from different stages
quite a lot of data	Finetune a few layers	Finetune a larger number of layers

Check: http://cs231n.github.io/transfer-learning/

Basically, Fine tuning with very little target data would lead to overfitting, so never do that. But with a lot of target data we can afford to do fine tuning (retraining with bakprop) without risking overfitting.

With very similar target data, the last couple of layers (very specific) from the ImageNet would still be relevant for the target data and thus use them.
But with very different data, cut off the last couple of layers (FCs and others) and use the bottom layers close to the input which are quite generic (blob, gradient detection).

4.b How could you test the invariance of a convnet towards rotation?

Do an experiment! Train your network on ImageNet, use a small image/patch and rotate it. Check for the performance as a function of degree of rotation.

5 GANs

GANs by the creators: **https://arxiv.org/pdf/1406.2661.pdf**

5.a What is a GAN? Explain its components. What is the input/output of each component?

A Generative Adversarial Network (GAN) is a generative model that learns to estimate the underlying probability distribution of the data using an adversarial strategy of two competing models:

- a **generator** G outputs synthetic samples given a noise variable input z
- a **discriminator** D that is optimized to tell the fake samples from the real ones

5.b Write down steps for training a GAN? (hint: 2 steps)

1. Fix G and train D: Generate fake inputs from generator and train discriminator on combination of fake data and real data (possibly multiple times).
2. Train G and fix D: Train generator with backpropagating output error of the discriminator (possibly multiple times).

5.c Write down the cost function that the generator G is optimizing. (check Table below)

the generator is trained to increase the chances of D producing a high probability for a fake example:

Cost function for G: $\mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$.

So the goal is to find a generator G that minimizes cost function above.

Symbol	Meaning
Pz	distribution of noise z
Pg	distribution of generated samples
Pr	distribution of real samples

5.d Write down the cost function that the discriminator D is optimizing.

the discriminator is expected to output a probability close to zero for a fake sample and close to 1 for a real sample, so we try to maximise the probability of assigning the correct label.

Cost function for D: $\mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$

maxD Ex~P_r [logD(x)] + E_z~Pz [log(1 − D(G(z)))]

5.e Write down the total loss function L(D, G).

min_G max_D Ex~Pr [logD(x)] + Ez~Pz [log(1−D(G(z)))]
min_G max_D Ex~Pr [logD(x)] + Ex~Pg [log(1−D(x))]

5.f When does the global optimal happen? (hint: consider the goal of a GAN)

When the generative distribution p_g is equal to the distribution of the dataset p_r: $p_g = p_r$.

5.g What is the optimal value for the function a.log(y) + b.log(1 − y)?
(hint: compute df(y) = 0 dy)

y∗ = a/(a+b)

5.h Rewrite loss function L(D, G) by using expectation formula. (hint: Ex~p(x)[f (x)] = xp(x)f(x)dx)

∫ L(G, D) = x(Pr (x)log(D(x))+Pg (x)log(1−D(x)))dx

$$V(G,D) = \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_{z} p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz$$
$$= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx$$

5.i What is the optimal value for discriminator D? (Hint: rename x = D(x), A = Pr (x), B = Pg (x))

by changing the variables: f (x) = A.log(x) + B.log(1 − x) therefore x∗ =A/(A+B) ⇒ D∗ = Pr / (Pr +Pg)
Once the generator is trained to its optimal value, Pg gets very close to Pr. When Pg = Pr , D^∗=1/2.
(the optimal value is an unbiased coin flip, which would mean the generator is producing sufficiently real looking images)

5.j What is the global optimal value for L(D, G)?

L(G, D) = ∫_x (Pr (x)log(D(x)) + Pg (x)log(1 − D(x)))dx
= log 1/2 ∫_x Pr(x)dx + log 1/2 ∫_x Pg(x)dx
= −2 log2

6 Problems of training GANs

6.1 Stability

It has better answers without trying to answer the questions!
https://medium.com/@jonathan_hui/gan-why-it-is-so-hard-to-train-generative-advisory-networks-819a86b3750b

Consider a two-player game. Suppose first player takes control of x to minimize f1(x) = xy, while the other player updates y to minimize f2(y) = −xy.

6.a Write down the cost function for this problem. (Hint: it's a min-max problem)

min_f1 max_f2 L(f1, f2) = xy

6.b Intuitively, what is the optimal values of x and y (equilibrium point)?

x = y = 0

6.c Write down the learning update formula for parameters x and y.

$\partial f1/\partial x = y$ and $\partial f2/\partial y = -x$
$x \leftarrow x - \alpha y$
$y \leftarrow y + \alpha x$

6.d This is a non-cooperative game that each model updates its cost independently with no respect to another player in the game. What is the problem of optimizing using gradient descent? Can it reach the equilibrium point? (hint: update x and y for multiple iterations and check their behaviours)

As x and y have different signs, every following gradient update causes huge oscillation and the instability gets worse in time. As a result, it will not converge to an equilibrium.

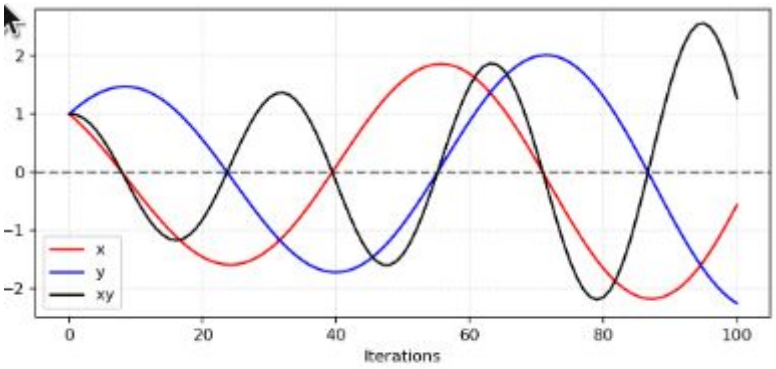


Figure 2: oscillation of the function

6.2 Vanishing Gradient & Mode Collapse

7.a
Suppose that we have a very bad Discriminator. What will happen during training?

If the discriminator behaves badly, the generator does not have accurate feedback and the loss function cannot represent the reality.

7.b
Suppose that we have a perfect Discriminator. What will happen during training?

the loss function L falls to zero and we end up with no gradient to update the loss during learning iterations thus the learning becomes super slow or even jammed.

If the Discriminator is perfect, then it will always correctly distinguish fake images from real ones. This means that the gradient update of the discriminator is zero (it is already perfect, no need to change the weights).

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Because both G and D are trying to optimize the same function, if the gradient of D is zero, then the gradient of G is also zero. Due to that the generator will not improve and never create realistic images.

7.c

Mode collapse is when the generator generates a limited diversity of samples, or even the same sample, regardless of the input noise z. Suppose you have two modes A and B in your data. (e.g. one can say MNIST has 10 modes of 0-9). Describe a situation where mode collapse happens.

Consider the following steps:

1. The generator learns to fool the discriminator by producing very realistic single mode A regardless of noise vector z.
2. The discriminator thinks that As are real (so it generates loss). So it updates itself to detect this single mode.
3. The gradient from the discriminator back-propagates to the generator to correct itself. But Since the generator desensitizes the impact of z already, the gradients push the generator to move to the next victim mode B. Generators now just generate optimal Bs.
4. The discriminator now thinks that Bs are real (so it generates loss). So it updates itself to detect this single mode. It repeats...(see Fig. 3). In this case both networks are overfitted to exploit the short-term opponent weakness.

Figure 3: mode collapse

7.d

One cause of mode collapse is that there is nothing in the objective function of GANs that explicitly forces the generator to generate different samples given the input. Can you suggest a solution for it?

The logical straightforward solution is comparing samples across a batch instead of comparing each sample individually in isolation. Minibatch discrimination gives the discriminator the power to compare generated samples across a batch to help determine whether a batch is diverse or not.

A more complicated solution is to cluster the real images in advance, cluster per mode (that assumes that the differences between modes are greater than the distances between

images of the same mode). The generated images are assigned to clusters. When training the discriminator, real and fake images from each cluster are given. If the cluster has no fake image assigned, no real image from this cluster is used either. This process ensures that the discriminator will not forget modes, because all generated ones are included in the training and it ensures balance of real and fake images of different modes.

Other methods: Taken from http://aiden.nibali.org/blog/2017-01-18-mode-collapse-gans/

- Anticipate counterplay: Unrolling GANs: https://arxiv.org/pdf/1611.02163.pdf and https://medium.com/@jonathan_hui/gan-unrolled-gan-how-to-reduce-mode-collapse-af5f2f7b51cd
- Use experience replay: Once in a while substitute old Gen/Discrim in the model.

Tutorial Lecture 6

1 3D convolutions

Consider 5 gray-scale videos each with 16 frames and spatial size of 224x224. You want to generate a feature map with 64 output channels using 3x3x3 convolutional filters (proper padding).

1.a What is the dimension of the weight tensor? How many parameters there are in total?

Weight matrix: [64, 1, 3, 3, 3], bias: 64, total: 1792

1.b What is the dimension of the output feature maps if stride=1 in all 3 dimensions?

[5, 64, 16, 224, 224]

1.c What is the dimension of the output feature maps if stride=2 in temporal dimension and stride=1 for spatial dimensions?

[5, 64, 8, 224, 224]

1.d Suppose you have a zero-bias 3x3x3 kernel with parameters:

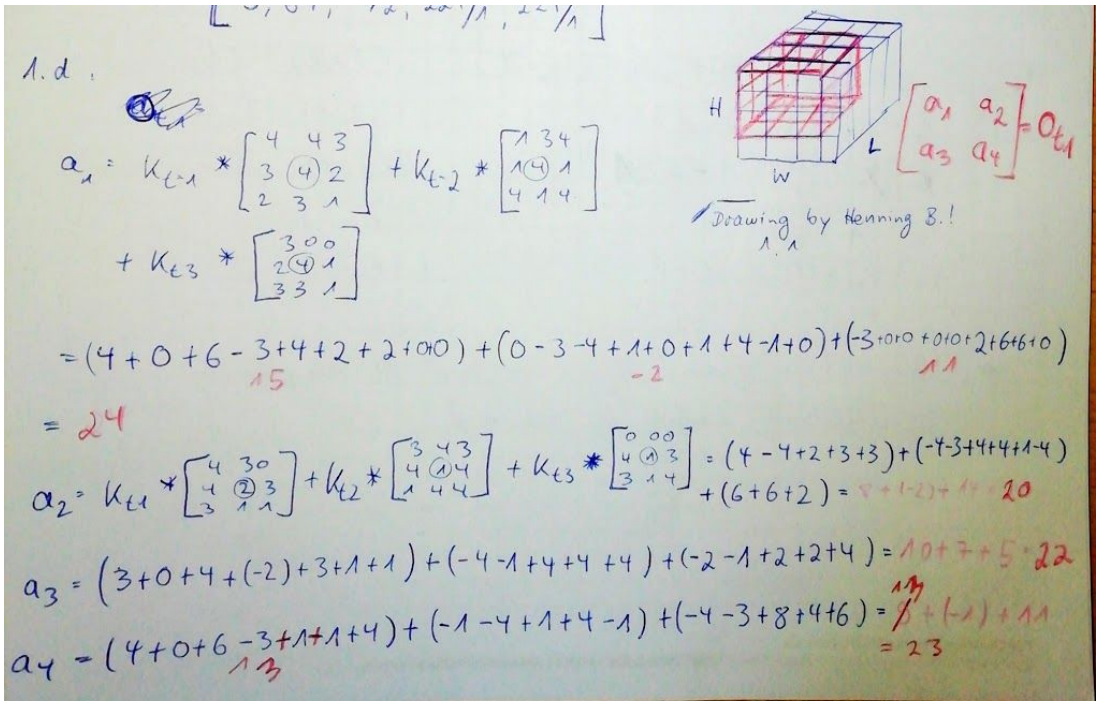
Suppose you have a zero-bias 3x3x3 kernel with parameters

$$K_{t=1} = \begin{bmatrix} 1 & 0 & 2 \\ -1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} K_{t=2} = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} K_{t=3} = \begin{bmatrix} -1 & 0 & -1 \\ 0 & 0 & 2 \\ 2 & 2 & 0 \end{bmatrix}$$

Compute the output feature maps for the input data of size 4x4x4

$$I_{t=1} = \begin{bmatrix} 4 & 4 & 3 & 0 \\ 3 & 4 & 2 & 3 \\ 2 & 3 & 1 & 1 \\ 1 & 4 & 3 & 1 \end{bmatrix} I_{t=2} = \begin{bmatrix} 1 & 3 & 4 & 3 \\ 1 & 4 & 1 & 4 \\ 4 & 1 & 4 & 4 \\ 4 & 0 & 1 & 2 \end{bmatrix} I_{t=3} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 2 & 4 & 1 & 3 \\ 3 & 3 & 1 & 4 \\ 1 & 2 & 3 & 0 \end{bmatrix} I_{t=4} = \begin{bmatrix} 2 & 1 & 0 & 4 \\ 3 & 1 & 1 & 0 \\ 3 & 1 & 1 & 2 \\ 4 & 1 & 3 & 4 \end{bmatrix}$$

```
Feature map: [[[24, 20], [22, 23]], [[28, 19], [14, 32]]]
```



Example on how to compute values by hand. Only for output layer 1. Hope that the drawing helps to understand how it's done.

1.e Compute the max-pool of 2x2x2 with stride=2 for the above output.

32

1.f Suppose that a standard 3D convolution layer takes an input feature matrix F with shape (lF , wF , hF , cF) and outputs a feature matrix G of size (lG, wG, hG, cG) where cF and cG are number of channels before and after the convolution. Suppose the kernel size is k × k × k.

1.g What is computation cost of a 3D conv in this case? compute it for k=3,5,7.

```
1.g What is computation cost of a 3D conv in this case? compute it for k=3,5,7.
Computation cost for k = 3 is 3x3x3x3x64x51x51x51 = 687662784
Computation cost for k = 5 is 5x5x5x3x64x50x50x50 = 3000000000
Computation cost for k = 7 is 7x7x7x3x64x49x49x49 = 7747892544
```

1.h What is the disadvantage of 3D convolutions compared to 2D convolutions?

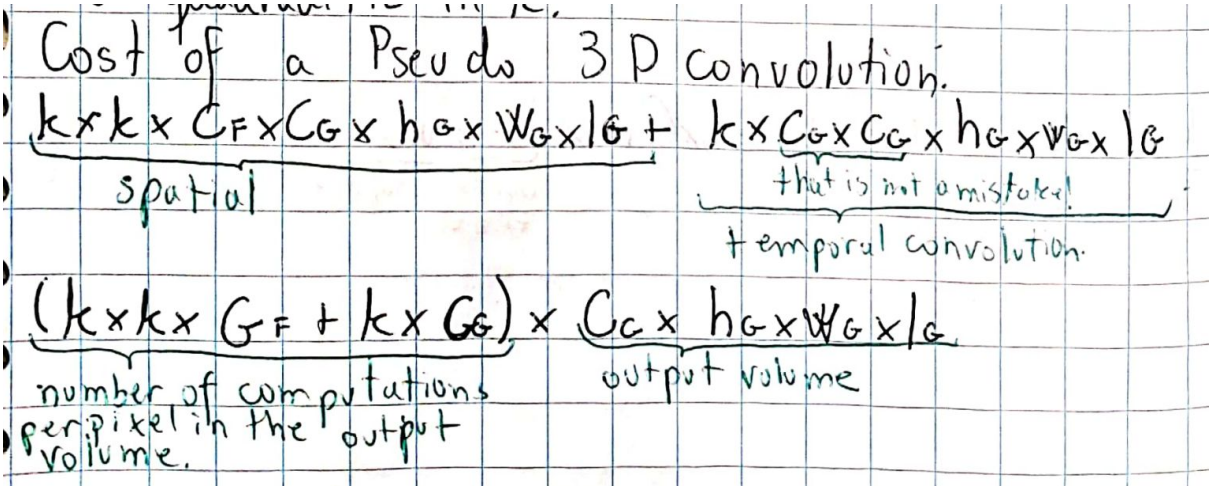
number of parameters grows exponentially compared to 2D convs (9 vs 27, 25 vs 125t, 49 vs 343), more computations.

1.i Give two solutions for above problem. what is their computational cost?

1) splitting a 3x3x3 convolution into a 3x3x1 convolution and a 1x1x3 convolution. cost:

```
-Pseudo 3D: A 2D convoltion is done in the spacial dimensions and a 1D convolution is done in the temporal dimension. (p
aper: https://arxiv.org/pdf/1711.10305.pdf)
    Computational cost: (2D_filter_size + 1D_filter_dize ) x output_channels x input_channels x output_size
                        (k x k          + k              ) x cG              x cF             x wG x hG x lG
```

Paper: https://arxiv.org/pdf/1711.10305.pdf
The Pseudo 3D convolution applies a 2D convolution in the spatial dimensions (the input frames) and then a 1D convolution in the time dimension. The computational costs grows quadratic with the kernel size instead of cubic for 3D convolutions.
The 1D convolution is applied on the resulting volume from the 2D convolution.

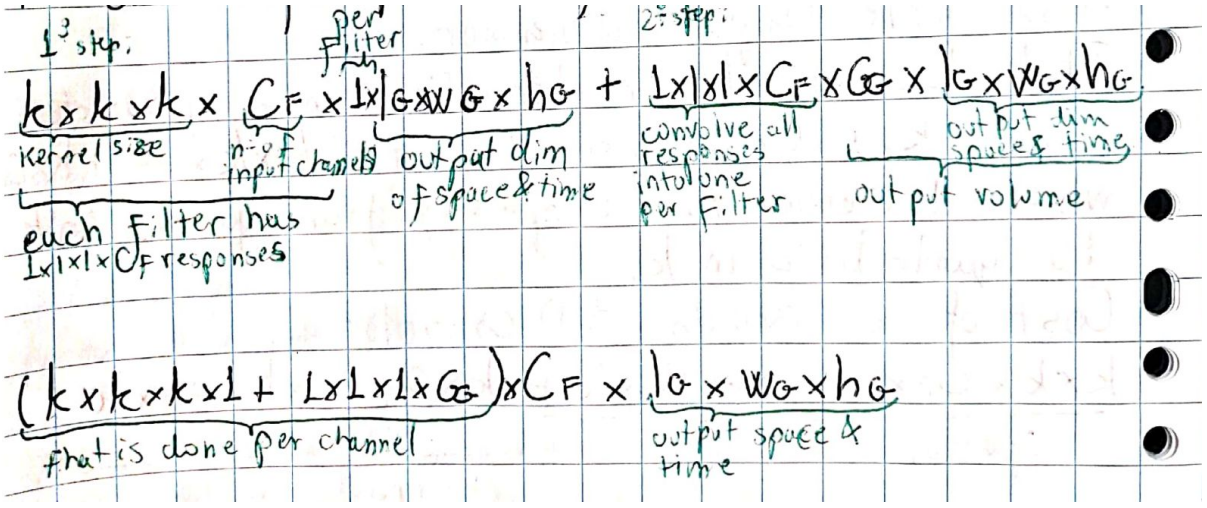


2) factorize a standard 3D convolution into: a. separate convolutions on separate channels; b. then pointwise convolutions on all the channels (to carry out a linear combination of layers) cost:
Paper: https://arxiv.org/pdf/1808.01556.pdf
Video: https://www.youtube.com/watch?v=T7o3xvJLuHk

```
-Depth Wise Convolution: It applies separete convolutions for each channel of the input and then convolve the channels t
ogether. (step by step viedo: https://www.youtube.com/watch?v=T7o3xvJLuHk, paper: https://arxiv.org/pdf/1808.01556.pdf)
    Computational cost: (filter_size + input_channels x unit_convolution) x output_channels x output_size
                        (k x k x k   + cF             x 1 x 1 x 1       ) x cG              x wG x hG x lG
```

In a normal convolution, all the filters are applied globally to all channels. In the depth wise this is done in two steps.

1- The channels are convolved separately. The response of one filter at this stage is a vector with the response of the filter applied to each channel.
2- This vector response is convolved into one single response for the filter.
The step 1 and 2 happen for every pixel of the output volume.



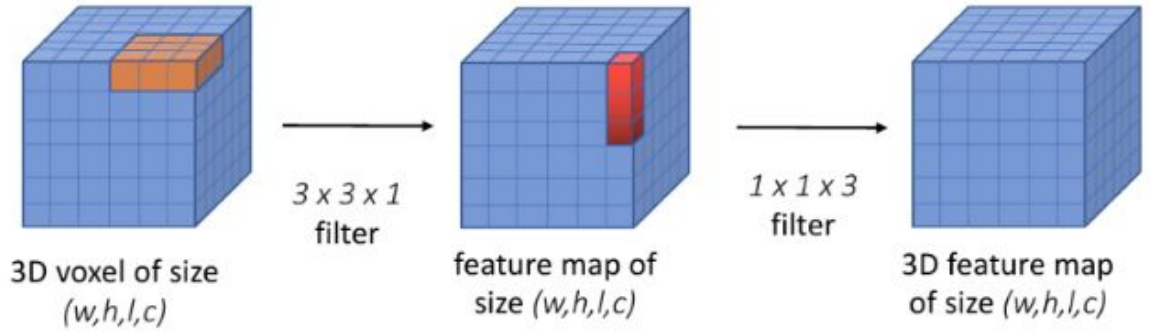


Figure 1: Pseudo convolution

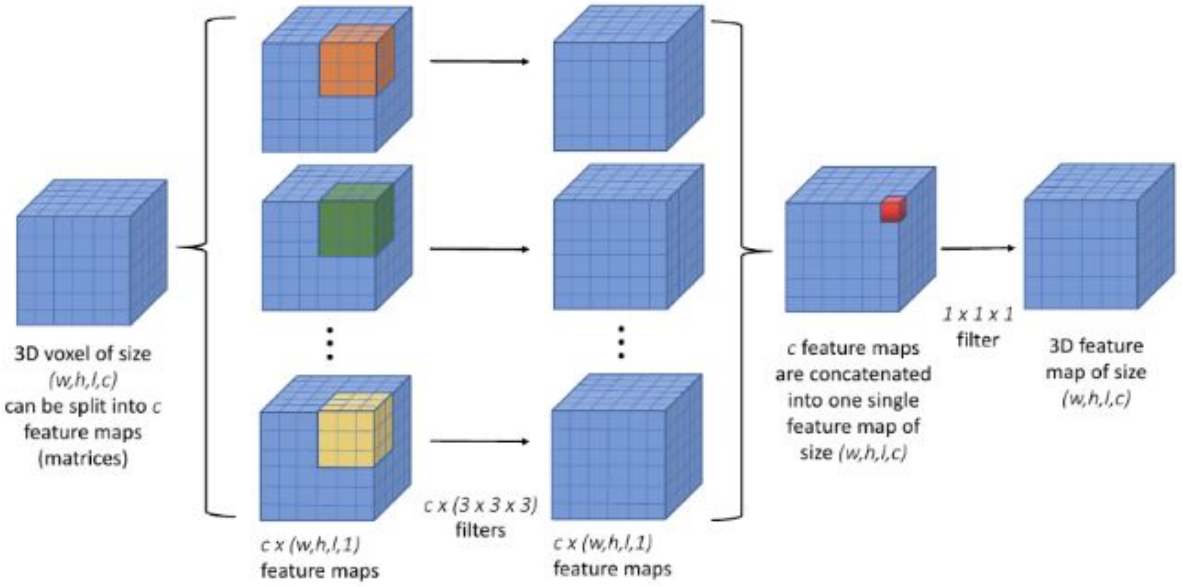


Figure 2: Depth-wise convolution

2 Vanila RNNs

Consider 5 gray-scale videos each has 16 frames and 224x224 resolution. You want to learn temporal dependencies using an RNN. The RNN has one layer, no-bias and 20 hidden neurons.

1.a Write down the function for the hidden layer of vanilla RNN.

$$h_t = tanh(W_i X_t + W_h h_{t-1} + b) = tanh(W[X_t, h_{t-1}] + b)$$

1.b What is the input dimension to the RNN?

224 * 224 = 50176, input tensor: [5, 16, 50176]

1.c What is the dimensionality of output?

20, weight tensor: [5, 16, 20]

1.d How many parameters it has in total?

Wi + Wh = 1003920,
Wi : [20, 50176]
Wh : [20, 20]

1.e How can a sigmoid function contribute to vanishing gradient problem?

1) with very big/small numbers, the output vector of sigmoid function z is almost binary: either 1 or 0. the gradient of sigmoid z is z*(1-z). Therefore, in both cases gradient become zero.
2) the gradient of z achieves a maximum at 0.25, when z = 0.5. That means that every time the gradient signal flows through a sigmoid gate, its magnitude always diminishes by half (or more)

1.f Suppose you have an image captioning model. What are the possible mechanisms for an RNN to determine the end of generated caption? (count 3 different ways)
The actual question here is: Suppose you have an image captioning model that outputs a sentence for a given image. What are the possible techniques that one can use to signal the RNN to stop the generation.

1- Let the RNN learn an EOF character. It stops the caption generation once it outputs the EOF.
2- The RNN has two outputs, every time step it outputs the word for the caption and a stopping score. When the stopping score exceeds a threshold, the RNN halts the generation.
3- (The stupid way) Two models are trained, a RNN and a NN (or other regression mmodel), the NN will output the lenght of the sentence and, the RNN will continue its generation until it reaches the max lengh defined by the NN.

3 LSTMs

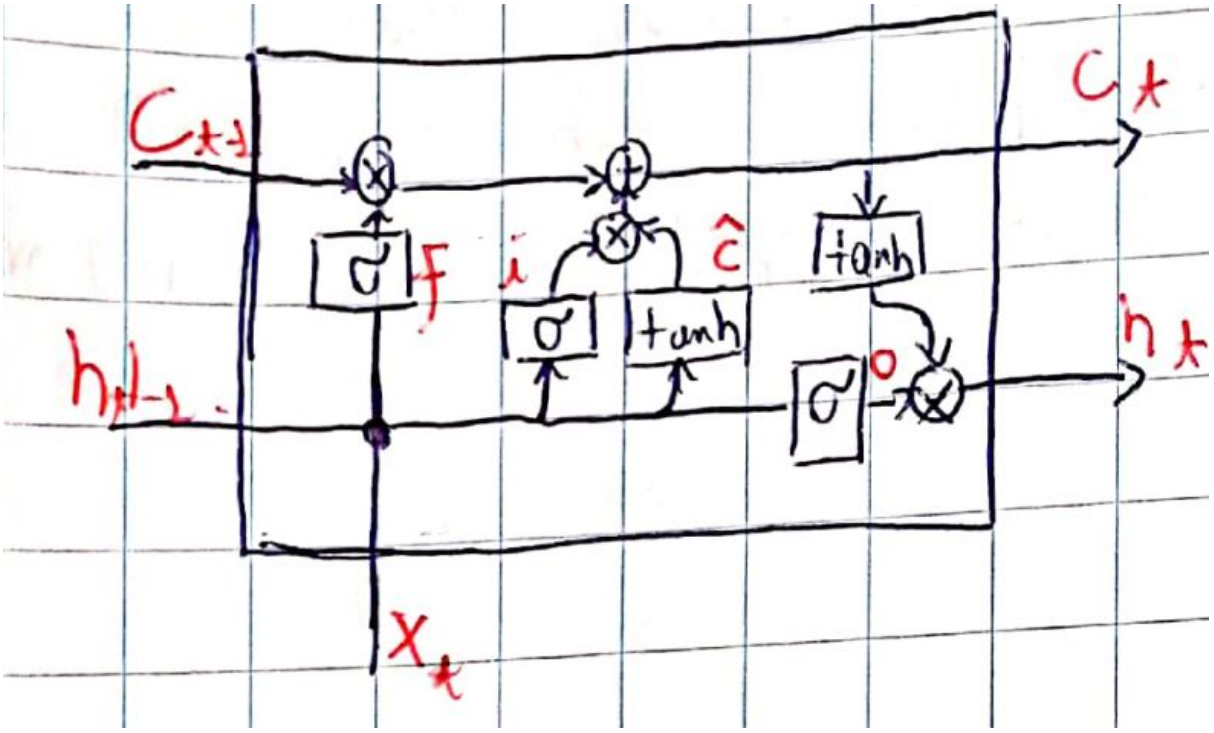
Consider 5 gray-scale videos each has 16 frames and 224x224 resolution. You want to learn temporal dependencies using an LSTM. The LSTM has one layer, no-bias and 20 hidden neurons.

1.a How many gates a LSTM has? Write down LSTM functions.

LSTM has 3 gates, but it has 4 units inside a cell.
$f = \sigma(W_f[X_t, h_{t-1}] + b_f)$ forget gate

$i = \sigma(W_i[X_t, h_{t-1}] + b_i)$ input gate
$c_i = tanh(W_{ci}[X_t, h_{t-1}] + b_{ci})$ not a gate!
$o = \sigma(W_o[X_t, h_{t-1}] + b_o)$ output gate (this part does the same thing as the RNN)

Outputs of the LSTM:
$c_t = f.c_{t-1} + i.c_i$ the cell state (acts as a long term memory)
$h_t = tanh(c_t).o$ the hidden state and used output of the LSTM. (acts as a short term memory)



1.b What is the input dimension to the LSTM?

224 * 224 = 50176, input matrix: [5, 16, 50176]

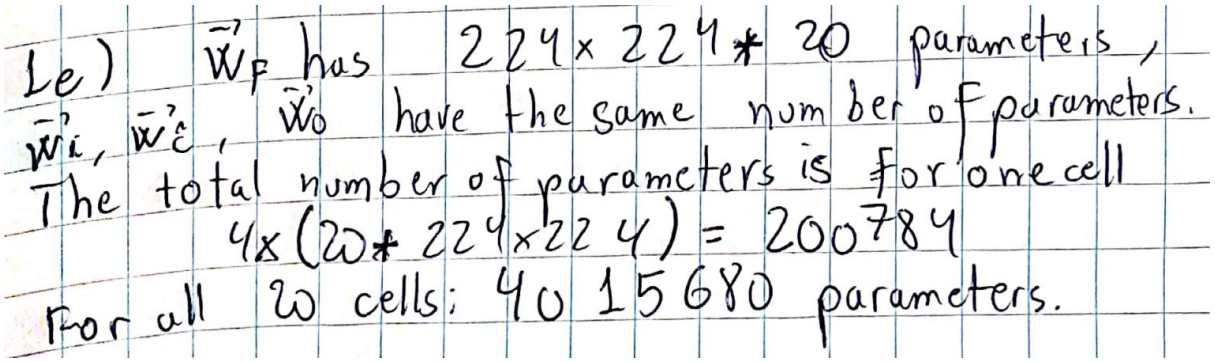
1.c What is the dimensionality of hidden state h?

20, matrix: [5, 16, 20]

1.d What is the dimensionality of cell state C?

20, matrix: [5, 16, 20]

1.e How many parameters LSTM has in total?
4 gates * (224 height * 224 width + 20 hidden cells) * 20 hidden cells = 4,015,680 parameters



input-hidden weights: Wii, Wif , Wig , Wio : [20, 50176]
hidden-hidden weights: Whi, Whf , Whg , Who : [20, 20]

1.f What is your solution to reduce the number of parameters of the LSTM?

first reduce the feature maps with a set of convolutional layers