4F12 Computer Vision (2017)

Q1(a)

(i) - Low-pass filter to remove high-frequency additive noise which is amplified in gradient operations

- σ determines amount of blur | large σ, small low-pass filter cut-off fre

(2)

(ii)

- $g_\sigma(x) * g_\sigma(y) = g_\sigma(x)\, g_\sigma(y)$

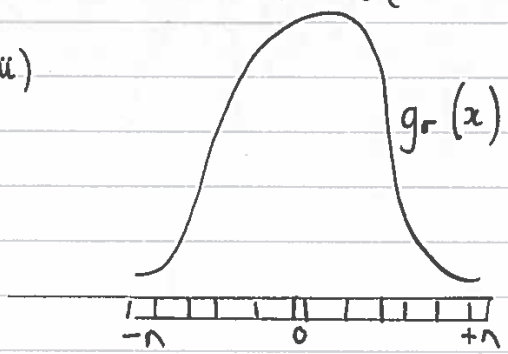
$= g_\sigma(x, y)$

where $g_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$

- Instead of N^2 multiplaters, $(N = 2n+1)$ we perform $2(N)$.

(4)

(iii)



Sample $g_\sigma(x)$ until tails are $< \frac{1}{1000}$ peak value $\quad g_{\sigma=1}(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$

n	-4	-3	-2	-1	0	1	2	3	4
$\frac{g_\sigma(i)}{g_\sigma(0)}$	$e^{-\frac{4^2}{2}}$	$e^{-\frac{3^2}{2}}$	$e^{-\frac{2^2}{2}}$	$e^{-\frac{1^2}{2}}$	1	$e^{-\frac{1}{2}}$	$e^{-\frac{2^2}{2}}$	$e^{-\frac{3^2}{2}}$	$e^{-\frac{4^2}{2}}$
					1	0.3[illegible]	0.135	0.011	0.0003

$e^{-8} < \frac{1}{1000}$

$\therefore n = 3$

$n = 3$ Size of filter $2n+1 = 7$

	$i=-3$	$i=-2$	$i=-1$	$i=0$	$i=1$	$i=2$	$i=3$	
X	0.004	0.054	0.242	0.399	0.242	0.054	0.004	X

(6)

Q1(b)

(i) Create a descriptor from vector of pixels in 16×16 patch, 25D vector

$$\underset{256\times 1}{X} = \begin{bmatrix} I(0,0) \\ \vdots \\ I(15,15) \end{bmatrix} \qquad X_{NC} = \begin{bmatrix} \frac{I(i,j)-\mu}{\sigma} \end{bmatrix}$$

Normalize by subtraction of mean, μ and σ (normalise length).

$$S.S.D = \sum_i \sum_j \left(I_1(i,j) - I_2(i,j)\right)^2 \longrightarrow \text{Compute } X_{NC_1} \cdot X_{NC_2} \text{ for comparison.}$$

(ii) (2)

SIFT descriptor

- sample 16×16 pixels $S(x, y, \sigma_i)$ at dominant orientation
- compute ∇S of pixels (gradient magnitude + orientation)
- weight gradients by gaussian weighting (σ = 8 pixels at scale of 16×16)
- produce 4×4 cells (16)
- produce HOG at 45° bins (8 dir^) , adding grad mag to bins
- concatonate to 128 D vector
- normalize length
- remove outliers by truncating any value >0.2 to 0.2.
- renormalise to a unit vector. (4)

(iii) N.C — very simple to compute, 25D, poor if scale + orientation incorrect

SIFT — invariant to scale/orientation + lighting (2)

— robust to small geometric/perspective distortion. Poor at boundaries

Q2

(a) (i) pin-hole camera
planar perspective onto plane
no non-linear distortion. (eg no radial lens distortion)

$$X = \frac{X_1}{X_4}$$
$$Y = \frac{X_2}{X_4}$$
$$Z = \frac{X_3}{X_4}$$

homogeneous representation

(ii). S represents non-linear inverse-depth scale (geometry)
S is homogeneous scaling (algebra)
(denominator).

(iii) Let $X_4 \to 0$ for pts. at ∞

(iv) $\underset{3\times4}{P} = [K][R \mid T]$

orientation of = rotation matrix (orthonormal matrix, 3 dof)

K: 3×3 orthonormal; $[R \mid T]$: 3×1; camera position. (Optical centre) (3 dof).

$$K = \begin{bmatrix} f k_u & 0 & u_0 \\ 0 & f k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where k_u, k_v (scale/pixel size) are internal parameters
u_0, v_0] principal point
f – focal length

2(b)(i) $\begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ x_{i4} \end{bmatrix} = \begin{bmatrix} a_1 + \lambda_i b_1 \\ a_2 + \lambda_i b_2 \\ a_3 + \lambda_i b_3 \\ 1 \end{bmatrix}$ (2)

Now let $\lambda_i \to \infty$ and rescale by dividing by λ_i $\begin{bmatrix} \frac{a_1}{\lambda_i} + b_1 \\ \frac{a_2}{\lambda_i} + b_2 \\ \frac{a_3}{\lambda_i} + b_3 \\ \frac{1}{\lambda_i} \end{bmatrix}$ (2)

$$\begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ x_{i4} \end{bmatrix} \longrightarrow \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ 0 \end{bmatrix}$$

$\therefore$ Under perspective projection: $\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} & P & \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ 0 \end{bmatrix}$

$$= [K|R] \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

(ii)

$$u_\infty = \frac{p_{11} b_1 + p_{12} b_2 + p_{13} b_3}{p_{31} b_1 + p_{32} b_2 + p_{33} b_3}$$

$$v_\infty = \frac{p_{21} b_1 + p_{22} b_2 + p_{23} b_3}{p_{31} b_1 + p_{32} b_2 + p_{33} b_3}$$

(4)

All lines with same $\underline{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$ will have same V.P which is independent of $\underline{a}$

(iii). Find VPs for $X = \infty$ and $Y = \infty$ and $\underline{h} = \underline{V_{PX}} \times \underline{V_{PY}}$. (4)

$$\underline{h} = \begin{vmatrix} i & j & k \\ p_{11} & p_{21} & p_{31} \\ p_{12} & p_{22} & p_{32} \end{vmatrix}$$

equation is $[x \; y \; 1]^T \underline{h} = 0$

Q3(a) (i) (A) Rotation about optical axis with no translation

$$\underset{3\times3}{T} = \underset{\text{camera calibration}}{K} \underset{3\times3}{R} K^{-1} \qquad \underline{t} = 0$$

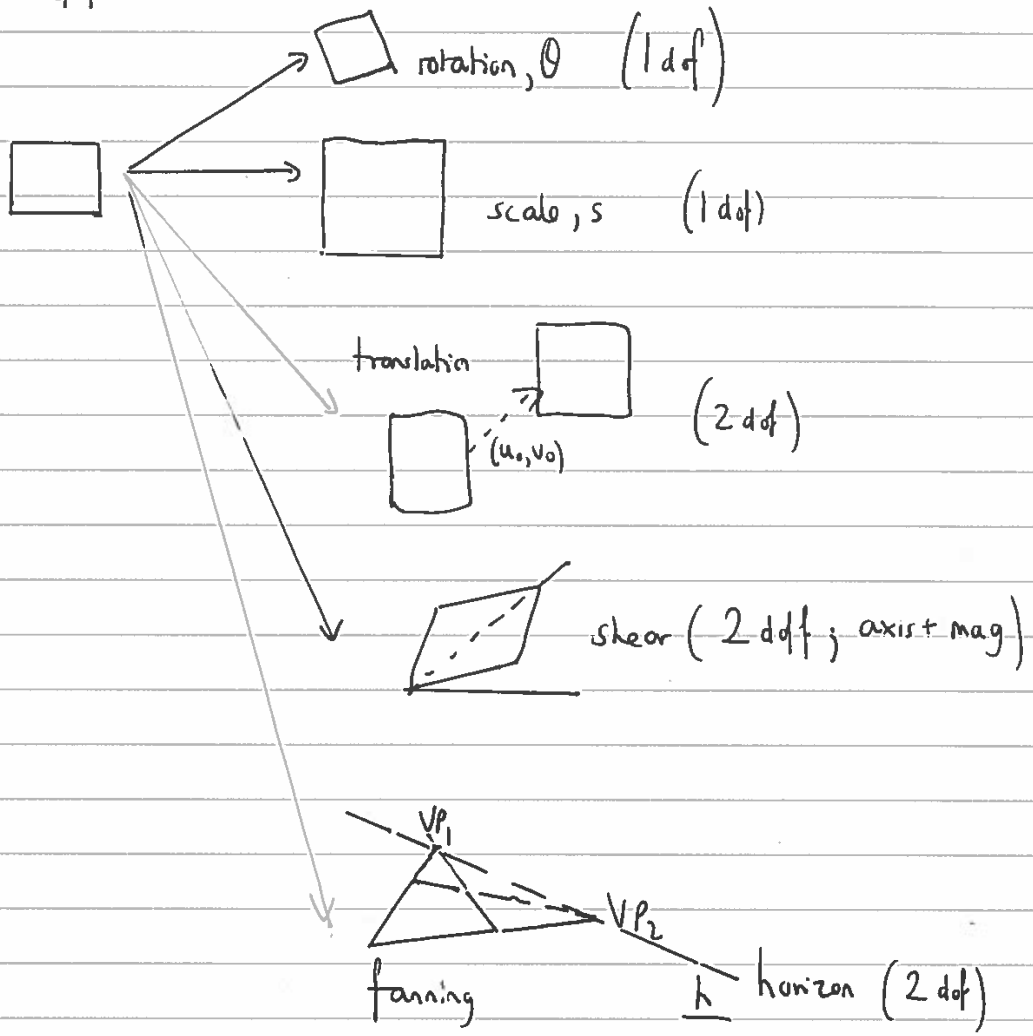
(B) Viewing a plane eg $Z=0$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \underbrace{\begin{bmatrix} r_{11} & r_{12} & t_1 \end{bmatrix}}_{P_1} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \qquad \text{where } R_1 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \end{bmatrix}$$

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = K \underbrace{\begin{bmatrix} r_{21} & r_{21} & t_2 \end{bmatrix}}_{P_2} \underline{X} \qquad \text{where } R_2 = \begin{bmatrix} r_{21} & r_{22} & r_{23} \end{bmatrix}$$

$$\therefore \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \underbrace{K \begin{bmatrix} r_{21} & r_{22} & t_2 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_1 \end{bmatrix}^{-1} K^{-1}}_{T} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Q3(b) 8 dof.



(4)

(ä)

Under weak perspective

$$T_{WP} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & t_{33} \end{bmatrix} = \begin{bmatrix} t_{11}' & t_{12}' & t_{13} \\ t_{21}' & t_{22}' & t_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

becomes an Affine transformation

and no non-linear perspective effects

(ie. no fanning, // lines stay //).

(2)

Q3(b) Rigid-body motion — epipolar geometry

$$\underline{X}' = R\underline{X} + \underline{T}$$

$$\underline{E} = T_x R \qquad \text{where } T_x = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

$$\underline{F} = K^{-1} T_x R K^{-1}$$

(c) Find $n \geq 8$ correspondences and solve $A\underline{f} = 0$

$$\begin{bmatrix} a_{n\times 1} \\ \\ f_{ij} \text{ of } F \end{bmatrix}$$

Project F to F' with rank 2 i.e. $\det F = 0$

Solve for $E = K^T F K$ using known K

Now $E = T_x R$

Decompose into anti-symmetric and orthonormal matrix by SVD of E

$$\therefore E = U \Lambda V^T \qquad \text{where } \Lambda = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$T_x = U \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T \qquad R = U \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T$$

4 solutions

Resolve ambiguity by check depths are positive.

Q4

a) Describe the components that make up a standard convolutional neural network that is used for image classification. Briefly explain the rationale for the form of each of the components. [40%]

Answer

Bookwork that should include the following:

- Convolutional filtering, motivated by the translation invariance of images and the need to reduce the number of parameters to be learned
- (Pointwise) Non-linearities, motivated by the fact that non-linear computation needs to be made, and certain point-wise non-linearities followed by linear weighting are universal approximators.
- Pooling / sub-sampling, motivated by the fact that high-level features are coarser and that subsampling again reduces the number of parameters. It also helps to build in translation invariance.

b) Compute the derivative required to implement gradient descent learning of the network's convolutional weights W. Simplify your expression and interpret the terms. [40%]

Answer

To compute the derivative we use backpropagation (aka the chain rule)

$$\frac{\mathrm{d}}{\mathrm{d}W_{a,b}}G(V,W) = \sum_{n=1}^{N}\sum_{i,j}\frac{\mathrm{d}G(V,W)}{\mathrm{d}x^{(n)}}\frac{\mathrm{d}x^{(n)}}{\mathrm{d}y_{i,j}^{(n)}}\frac{\mathrm{d}y_{i,j}^{(n)}}{\mathrm{d}a_{i,j}^{(n)}}\frac{\mathrm{d}a_{i,j}^{(n)}}{\mathrm{d}W_{a,b}} + \beta W_{a,b}. \tag{1}$$

where each of the terms are,

$$\frac{\mathrm{d}G(V,W)}{\mathrm{d}x^{(n)}} = \frac{x^{(n)} - \mathrm{t}^{(n)}}{x^{(n)}(1-x^{(n)})}, \quad \frac{\mathrm{d}x^{(n)}}{\mathrm{d}y_{i,j}^{(n)}} = V_{i,j}x^{(n)}(1-x^{(n)}) \tag{2}$$

$$\frac{\mathrm{d}y_{i,j}^{(n)}}{\mathrm{d}a_{i,j}^{(n)}} = f'(a_{i,j}^{(n)}) \quad \frac{\mathrm{d}a_{i,j}^{(n)}}{\mathrm{d}W_{a,b}} = Z_{i-a,j-b}^{(n)} \quad . \tag{3}$$

Combining the terms together yields

$$\frac{\mathrm{d}}{\mathrm{d}W_{a,b}}G(V,W) = -\sum_{n=1}^{N}\left(\mathrm{t}^{(n)} - x^{(n)}\right)\sum_{i,j}V_{i,j}f'(a_{i,j}^{(n)})Z_{i-a,j-b}^{(n)} + \alpha W_{ik}. \tag{4}$$

So, the derivative is simply the sum over all datapoints of the error between the predicted and true labels $\left(x^{(n)} - \mathrm{t}^{(n)}\right)$ multiplied by the sensitivity of the network's output on the convolutional weights plus a linear weight decay term. The sensitivity is a convolution between the product of the output weights and non-linearity derivative $V_{i,j}f'(a_{i,j}^{(n)})$ and the image flipped in the x and y directions $Z_{-i,-j}^{(n)}$.

c) The company would like to extend the network to estimate the average population density of the area in each image. Describe how to extend the architecture of the network to perform this additional task. Explain your design. [20%]

Answer

There are lots of possible ways of improving the architecture of the network.

The first step is to **add a second output to the network** $p^{(n)}$ that models the average population density ρ. It makes sense to use the same features as are used for classification to do this as you would expect population density to be higher in urban areas. These common features can be passed through a different set of output weights $U_{i,j}$ in order to form the scalar population density estimate output of the network,

$$p^{(n)} = \sum_{i,j} U_{i,j} y_{i,j}^{(n)}.$$

The network's regression weights U can then be trained by adding a regression term to the objective function and optimising all parameters together,

$$H(U, W) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} \left(\rho^{(n)} - p^{(n)}\right)^2 + \frac{\alpha}{2} \sum_{i,j} U_{i,j}^2.$$

Strictly the question does not ask for the form of the new objective, but it interacts with the design decisions.

There are also more general enhancements that could be mentioned, but which are not a substitute for the above.

i. One enhancement would be to use **additional sets of convolutional weights**. Currently the method only uses one set and this means that it is only able to extract a single feature (e.g. a specific oriented edge) to perform classification / regression.

ii. A second enhancement, would use a **pooling/subsampling stage** after the non-linear stage. This would pool over a local neighbourhood and pick e.g. the max or average value. This will introduce shift invariance and reduce the number of parameters that are required in the layers above.

iii. A third enhancement would be to use a neural network with **many layers** each of which is structured as above. Together these enhancements lead to deep convolutional neural networks.

4F12 Comments:

Q1

Well answered by most candidates. Only part to cause problems being (b)i- describing a descriptor for normalised intensities for use in cross-correlation.

Q2

Well answered by most candidates. Only difficulty was in (b)ii in deriving the equations for the perspective projection of a line with the given projection matrix and then computing the vanishing point. Many struggled with a simple derivation of the equation of the horizon.

Q3

A well-answered question. Marks were lost in not expressing the matrices in terms of the camera motion parameters.

Q4

An unpopular question because this was newer material but those that did attempt it tended to provide good answers. When describing the convolutional neural network in part (a) a common error was to describe just a single convolution / non-linear / subsampling component without mentioning how these are used to build a network. Many candidates failed to identify the last part of the question as a regression problem. The few candidates that attempted made good progress.