4F12 Computer Vision (Crib 2015)

Q1.(a) Raw pixels — sensitive to contrast + brightness changes $I' = \alpha I + \beta$
— depend on lighting (position + distribution) + camera optics (interna

Normalize — edge detection and image gradients (edges, HOGs, SIFT) ie. deri
— normalize intensities $\left(\frac{I_i - \mu}{\sigma}\right)$ as in NCC matching

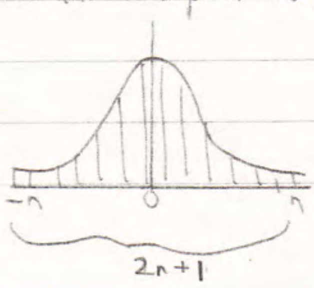
1(b). (i) Low-pass filtering to reduce noise before differentiation
Use a 2D Gaussian

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

implemented as 2 1D convolutions

$$\boxed{\begin{aligned} S(x,y) &= I(x,y) * G_\sigma(x,y) = I(x,y) * g_\sigma(x) * g_\sigma(y) \\ &= \sum_{-n}^{+n}\sum_{-n}^{+n} I(x-u, y-v)\, g_\sigma(u)\, g_\sigma(v) \end{aligned}}$$

where $g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$ and is sampled to $2n+1$ * discrete values



1b(ii).

Compute scale-space $S(x, y, \sigma_i^2)$ for $\underline{\sigma_i = 2^{\frac{i}{5}} \sigma_0}$
(ie. log spacing for σ_i).

— 2D convolution is done by 2 × 1D convolutions
— incremental blur in each octave

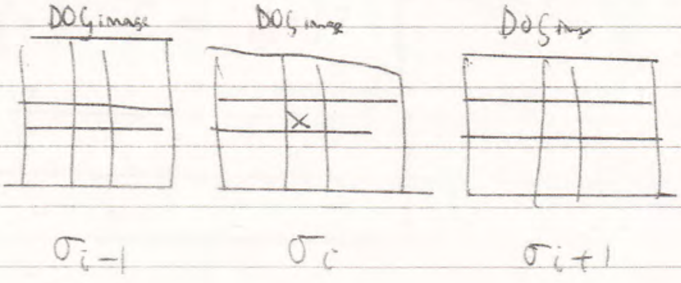
$$\sigma_{i+1} = 2^{\frac{1}{5}} \sigma_i \quad \text{and} \quad g_{\sigma_{i+1}} = g_{\sigma_i} * g_{\sigma_k}$$

— when σ_i doubles after 5 blurs, <u>subsample</u> to $\frac{1}{4}$ size
and repeat incremental blur with small kernels.
* (<u>same</u> kernels)

1(c) (i) Blob-like shapes: — convolve with $\nabla^2 G_\sigma(x,y)$ and look for max/min in $\nabla^2 S_\sigma(x,y)$

— $\nabla^2 S_\sigma(x,y) \simeq S(x,y,\sigma_{i+1}^2) - S(x,y,\sigma_i^2)$
ie. use DOG images computed from image pyramid neighbours

— Evaluate 26 neighbours to get position and σ_i



— max/min in $\nabla^2 S(x,y,\sigma_i)$ is blob centre (x,y) and blob size (σ_i)

Q1

1(c)(ii) Feature orientation — sample 16x16 pixel gradient ~~at~~ approp. scale and produce a smoothed histogram (10° bins). Smooth

— peak is dominant orientation

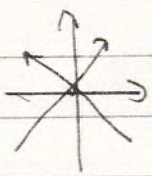
Sample 16x16 pixels around blob centre in σ_i image aligned with dominant orientation, $S(x, y, \sigma_i)$, by interpolation.

1c(iii) – Sample gradients in 16x16 patch after

– blur by $g(1.5\sigma)$.

– Produce HOG in 4x4 cells

Each bin records gradients in 8 dirⁿ (with interpolation)



orientation dirⁿ

– concatenate ~~normalize~~ 4x4 HOGs to 128D vector

and normalize size; truncate any entry above 0.2 to 0.2 to reduce effect of very strong gradients/highlights.

Invariance – use derivative + normalization ∴ invariant to contrast + brightness

– use Histograms (+ scale) → invariant to exact alignment

– Scale selection – invariant to to size of image

– works over small viewpoint change (< 30°)

Cannot cope with large viewpoint change + features on occluding boundaries.

Q2(a) - Pin-hole camera, perspective onto plane; no non-linear distortion

- $u = \frac{x_1}{x_3}$ $v = \frac{x_2}{x_3}$

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

- If $x_3 = 0$: pt at ∞ in image plane
  $X_4 = 0$ pt at ∞ in 3D world

2(b) (i) $u_i = \frac{p_{11} X_i + p_{12} Y_i + p_{13} Z_i + p_{14}}{p_{31} X_i + p_{32} Y_i + p_{33} Z_i + p_{34}}$

$v_i = \frac{p_{21} X_i + p_{22} Y_i + p_{23} Z_i + p_{24}}{p_{31} X_i + p_{32} Y_i + p_{33} Z_i + p_{34}}$

(ii) Rearrange into 2 equations which are linear in (X_i, Y_i, Z_i) i.e. planes

$0 = (p_{11} - u_i p_{31}) X_i + (p_{12} - p_{32} u_i) Y_i + (p_{13} - p_{33} u_i) Z_i + (p_{14} -$

$0 = (p_{21} - v_i p_{31}) X_i + (p_{22} - p_{32} v_i) Y_i + (p_{23} - p_{33} v_i) Z_i + (p_{24} -$

These 2 plane are not // and hence define ray.

2b(iii) Same equations are linear in p_{ij} if $X_i Y_i Z_i$ and u_i, v_i are k[illegible]

Rows of A: 2 per image measurement.

$\therefore$ Stack linear equations $n > 6$ we can solve for p_{ij} (12×1) vector

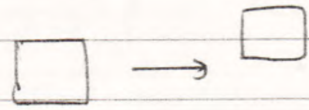
— $\underset{2n\times 12}{A}\ \underset{12\times 1}{\underline{p}} \simeq 0$ find $|\underline{p}|$ as linear least-squares sol^n (SVD)

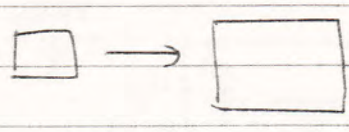
— Non-linear optimisation $\min\limits_{p_{ij}} \sum\limits_i (u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2$

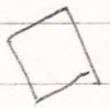
2(c).(i) Let $Z = 0$ for 2D projection of plane $\rightarrow$ 2D projective transformation

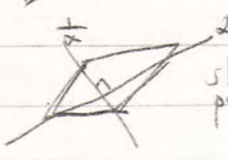
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \overset{3\times 3}{\begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix}} \begin{bmatrix} X_1 \\ X_2 \\ X_4 \end{bmatrix}$$

(ii) 8 d.o.f.

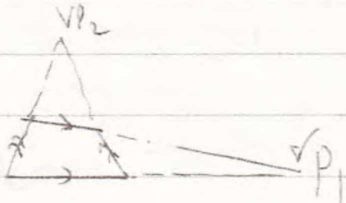
—  u_0, v_0 translation (2 dof)

—  uniform scale (1 dof)

—  rotation (1D)

— 

shear : axis + magnitude (2 dof)
pure.

— $\parallel$ lines have vanishing points 

horizon has line (2 dof)

6

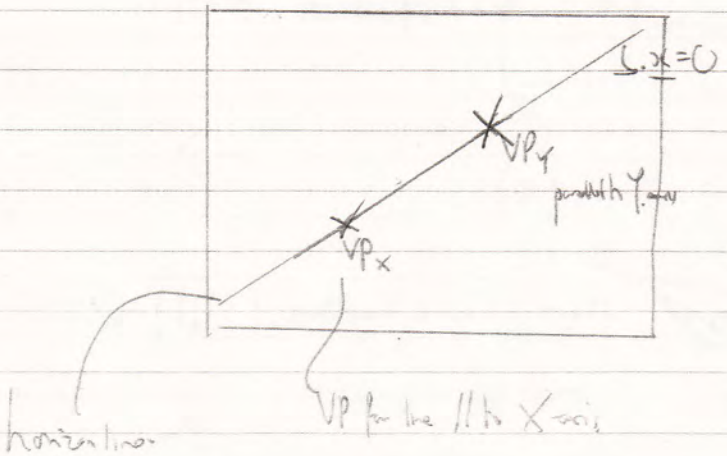
2(c)(iii)

Let $X_4 \to 0$ to rep point $\to \infty$

$X \to \infty$ $\quad VP_x = \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \end{bmatrix}$ since $\begin{bmatrix} & & \\ & P & \\ & & \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \end{bmatrix}$

$Y \to \infty$ $\quad VP_Y = \begin{bmatrix} p_{12} \\ p_{22} \\ p_{32} \end{bmatrix}$ since $\begin{bmatrix} & & \\ & P & \\ & & \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} p_{12} \\ p_{22} \\ p_{32} \end{bmatrix}$

All other directions can be be written as $\begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix}$ and vanishing points will lie on line between VP_x and VP_Y, $\underline{l}$.



horizon $\underline{l} = \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \end{bmatrix} \times \begin{bmatrix} p_{12} \\ p_{22} \\ p_{32} \end{bmatrix}$

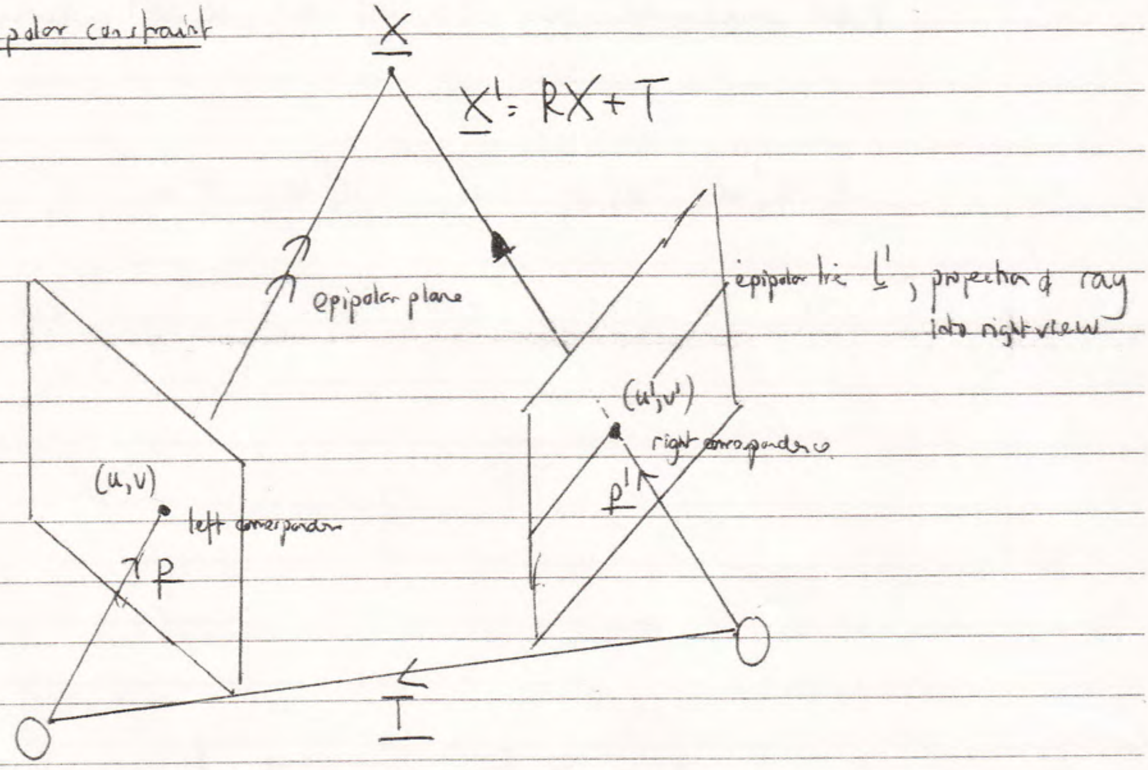
where $\underline{x} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$

$\underline{l}.\underline{x} = 0$

$$0 = u(p_{21}p_{32} - p_{31}p_{22}) + v(p_{31}p_{12} - p_{11}p_{32}) + (p_{11}p_{22} - p_{21}p_{12})$$

7

Q3 (a)(i) Epipolar constraint



→ restrict matching points to a line in 2nd view corresponding to the perspective projection of ray defined by first view.

a(ii). 2 rays and baseline are <u>coplanar</u>

$$\underline{\omega} = K\underline{p} \quad \text{and} \quad \underline{\omega}' = K'\underline{p}'$$

Coplanarity can be written by:

$$\underline{p}' \cdot \left(\underline{T}_\times R \underline{p} \right) = 0$$

$$\text{or} \quad \underline{p}'^T \left[T_\times R \right] \underline{p} = 0 \qquad \text{or} \quad \underline{p}'^T \underset{\text{essential matrix}}{E} \underline{p} = 0$$

$$\text{In terms of pixels: } \underline{\omega}' \left[K'^{-T} \, \underline{T}_\times R \, K^{-1} \right] \underline{\omega} = 0$$

$$\text{where } \underline{\underline{F = K'^{-T} \, \underline{T}_\times R \, K^{-1}}}$$

8

3b(i)

Each correspondence pair (u_i, v_i) and (u_i', v_i') gives 1 equation in 8 unknowns

$$\begin{bmatrix} u_i'u_i & u_i'v_i & u_i' & v_i'u_i & v_i'v_i & v_i' & u_i & v_i & 1 \end{bmatrix} \underbrace{\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix}}_{\underline{f}}^{9\times 1} = 0$$

Stack up for $n \geq 8$ and solve for $\underline{f}$ by linear least squares

— $A\underline{f} \simeq 0$

— But F has special constraint $\det F = 0$

— optimise by NL optimisation and ensure $\operatorname{rank} F = 2$ by SVD
(i.e. set 3rd singular value to 0)

b.(ii) From F, $E = K'^T F K = T_\times R = U \Lambda V^T$ by SVD

$$T_\times = U \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T \quad \text{and} \quad R = U \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T$$

$$\underline{P} = K[I \mid 0] \quad \text{and} \quad \underline{P}' = K'[R \mid T]$$

9

3 (c). Recover by triangulation – solve least squares or non-linear optimisation.
By additional views, track features over multiple view to generate larger base-lines. Optimise (Bundle adjustment) for camera position, orientations and 3D positions.

Q4 (CNNcmb). 10

Q4 a) Provide a probabilistic interpretation for the network's output and use this to justify the form of the objective function. [20%]

Answer

Interpret the output of the network as the probability that a pedestrian is present in the image: $x(Z;V,W) = p(t=1|Z,V,W)$.

The probability of the dataset labels given the parameters is therefore:

$$p(D|V,W) = \prod_{n=1}^{N} x(Z^{(n)};V,W)^{t^{(n)}}(1-x(Z^{(n)};V,W))^{t^{(n)}})$$
$$= \exp\left(\sum_{n=1}^{N}\left(t^{(n)}\log x(Z^{(n)};V,W) + (1-t^{(n)})\log(1-x(Z^{(n)};V,W))\right)\right) \quad (1)$$

Placing independent zero-mean Gaussian priors over the weights $V_{i,j}$ and $W_{i,j}$ with variance $1/\alpha$ and $1/\beta$ respectively, yields:

$$p(V,W|\alpha) = \prod_{i,j} p(w_{i,j})p(v_{i,j}) = \prod_{i,j}\frac{1}{Z(\alpha,\beta)}\exp\left(-\frac{1}{2}\left(\alpha W_{i,j}^2 + \beta V_{i,j}^2\right)\right) \quad (2)$$

In this way we can interpret the objective function as relating to the probability of the weight vector given the training data, $p(W,V|D,\alpha) = \frac{1}{Z(\alpha,\beta)}\exp(-G(V,W))$.

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

Answer

The gradient descent algorithm operates as follows:

i. initialise the weights (e.g. using Gaussian noise with a small variance)

ii. compute the derivative of the objective function with respect to the convolutional weights $\frac{\mathrm{d}G(V,W)}{\mathrm{d}W_{i,j}}$

iii. step down the gradient $W_{i,j} \leftarrow W_{i,j} - \eta\frac{\mathrm{d}G(V,W)}{\mathrm{d}W_{i,j}}$ (η is a user defined learning rate)

iv. loop to step (2) until $\Delta G(V,W) < \text{tol}$

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

$$\frac{\mathrm{d}}{\mathrm{d}W_{a,b}}G(V,W) = \sum_{n=1}^{N}\sum_{i,j}\frac{\mathrm{d}G(V,W)}{\mathrm{d}x^{(n)}}\frac{\mathrm{d}x^{(n)}}{\mathrm{d}y_{i,j}^{(n)}}\frac{\mathrm{d}y_{i,j}^{(n)}}{\mathrm{d}a_{i,j}^{(n)}}\frac{\mathrm{d}a_{i,j}^{(n)}}{\mathrm{d}W_{a,b}} + \beta W_{a,b}. \quad (3)$$

where each of the terms are,

$$\frac{\mathrm{d}G(V,W)}{\mathrm{d}x^{(n)}} = \frac{x^{(n)} - t^{(n)}}{x^{(n)}(1-x^{(n)})}, \quad \frac{\mathrm{d}x^{(n)}}{\mathrm{d}y_{i,j}^{(n)}} = V_{i,j}x^{(n)}(1-x^{(n)}) \quad (4)$$

$$\frac{\mathrm{d}y_{i,j}^{(n)}}{\mathrm{d}a_{i,j}^{(n)}} = f'(a_{i,j}^{(n)}) \quad \frac{\mathrm{d}a_{i,j}^{(n)}}{\mathrm{d}W_{a,b}} = Z_{i-a,j-b}^{(n)} \quad (5)$$

Combining the terms together yields

$$\frac{\mathrm{d}}{\mathrm{d}W_{a,b}} G(V, W) = -\sum_{n=1}^{N} \left(\mathrm{t}^{(n)} - x^{(n)}\right) \sum_{i,j} V_{i,j} f'(a_{i,j}^{(n)}) Z_{i-a,j-b}^{(n)} + \alpha W_{ik}. \tag{6}$$

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) Describe enhancements to the architecture of the network that might improve its ability to perform pedestrian detection. [40%]

Answer

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

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.

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.

The answer should describe these enhancements in detail which is bookwork.

1. **Gaussian smoothing and Interest point descriptors for matching**. Attempted by 43/43 Part IIB candidates, average mark 13.8/20.

   A question covering convolution with low pass and band pass filters in scale space to localise features for image matching. Well answered by most candidates. Only part to cause problems being (c)i - finding the orientation of a feature of interest before sampling window of gradients for computing SIFT descriptor.

2. **Perspective projection, transformations and camera calibration**. Attempted by 43/43 candidates, average mark 13.1/20.

   A question covering perspective projection, planar homographies and vanishing points. Well answered by most candidates. Most candidates struggled with finding the equation of the horizon of the ground plane in terms of the projection matrix elements. compose the transformation to get the orientation of the plane.

3. **Multiple view geometry and 3D reconstruction**. Attempted by 42/43 candidates, average mark 13.9/20.

   A straightforward question covering multiview geometry and 3D reconstruction. The decomposition of the fundamental matrix to recover translation and rotation caused some erros. The last part on structure from motion - reconstruction and bundle adjustement was poorly answered.

4. **Object recognition with convolutional neural networks**. Attempted by 4/43 candidates, average mark 13.8/20.

   An unpopular question because this was new material. The few candidates that attempted made good progress.

Roberto Cipolla, Principal Assessor and Richard Turner, Assessor