Computer Vision (4F12 Cribs) - 2021

Q1(a)(i)

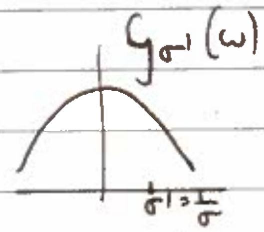
$$S(x,y) = \sum_{-n}^{n} \sum_{-n}^{n} g_\sigma(u)\, g_\sigma(v)\, I(x-u, y-v)$$

$$g_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$ sampled $N = 2n+1$ times

eg. $\sigma = 1$, $n = 3$

size of kernel $= N = 2n+1$

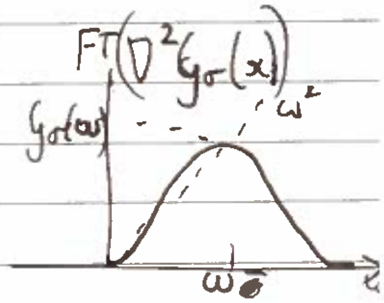
(ii) $g_\sigma(x,y) \overset{FT}{\Longleftrightarrow} G_{\sigma'}(\omega_1, \omega_2)$



$g_\sigma(x) \Longleftrightarrow G_{\sigma'}(\omega)$ where $\sigma' = \frac{1}{\sigma}$

$\therefore$ Low-pass filter with cut off frequency $\propto \frac{1}{\sigma}$

(iv) $\frac{d^2 g_\sigma(x)}{dx^2} \overset{FT}{\Longleftrightarrow} -\omega^2 G_{\sigma'}(\omega)$



band-pass filter

$$\sigma^2 \nabla^2 (G_\sigma * I) \simeq [G_{\sigma_{i+1}} * I - G_{\sigma_i} * I]$$

$$\simeq I * [G_{\sigma_{i+1}} - G_i]$$

Use difference of blurred images

Q1.a-iv

By examining first order Taylor expansions: $\frac{\partial I}{\partial x}|_{(x,y)} \approx I(x-1, y) - 2I(x, y) + I(x+1, y)$.
Similarly: $\frac{\partial I}{\partial y}|_{(x,y)} \approx I(x, y-1) - 2I(x, y) + I(x, y+1)$.
Hence, $\nabla^2 I|_{(x,y)} = \frac{\partial^2 I}{\partial x^2}|_{(x,y)} + \frac{\partial^2 I}{\partial y^2}|_{(x,y)}$.
The filter is: $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$.

Q1.a-iii

While a Fourier transform based implementation of a 2-D convolution can be very efficient for convolving with large filters, it has large over-head costs (requires two forward and one inverse Fourier transform). Direct 2-D convolution with small filters can be implemented very efficiently, especially if separable 2-D filters are used. Also note that smoothing with medium size kernels can be implemented as sequential smoothing with smaller size kernels, providing an efficient way of building image pyramids.

Q1(a)(v)

- Sample $S_\sigma(x, \sigma)$ at discrete $\sigma_i = \sigma_0 2^{i/s}$ (log-scale)

- use incremental blur to blur (in octave)

$$g_{\sigma_{i+1}} = g_{\sigma_K} * g_{\sigma_i}$$

$$\sigma_{i+1}^2 = \sigma_K^2 + \sigma_i^2 \qquad \sigma_{i+1} = 2^{\frac{1}{s}} \sigma_i$$

$$\therefore \sigma_K = \sigma_i \sqrt{2^{\frac{2}{s}} - 1}$$

- When $\sigma_i = 2\sigma_0$, subsample to $\frac{1}{4}$ size by skipping every other pixel and every other row (produce a new octave)

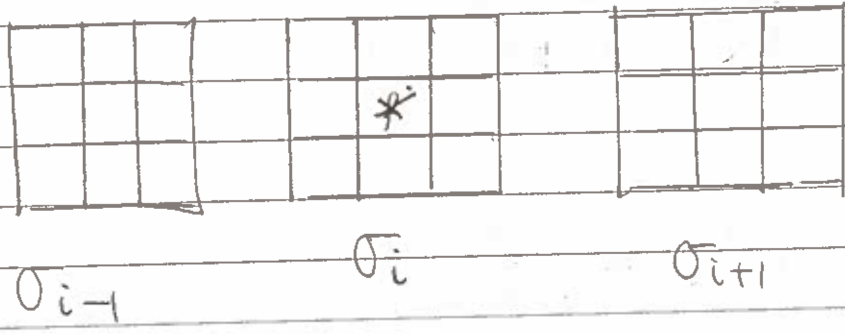
- re-use same incremental kernels g_{σ_K}

Q1(b)

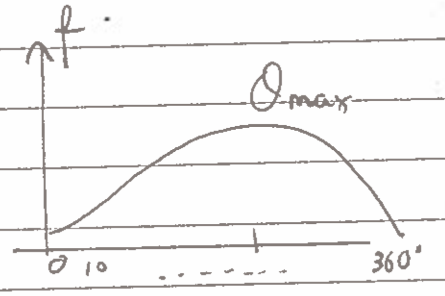
(i) Look for max/min in $\nabla^2 S_\sigma(x,y)$ response. Compute as

$$\sigma^2 \nabla^2 S_\sigma(x,y) \simeq S_{\sigma_{i+1}}(x,y) - S_{\sigma_i}(x,y)$$

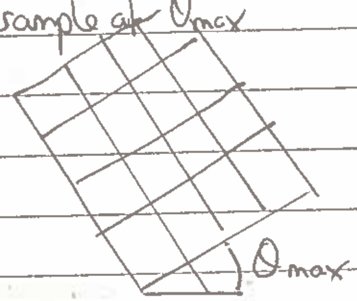
Look for local max/min by inspecting 26 neighbours



(ii) Sample 16x16 at $S_{\sigma_i}(x,y)$. Compute gradients $\nabla S_{\sigma_i}(x,y)$
Histogram, 10° bins and look for a peak
(need smoothing).
Peak is dominant orientation



Re-sample at θ_{max}



(iii) Encode 2D shape by looking at gradients ("edges")
Invariant to position, scale and orientation
Normalisation to unit vector, 128D, gives invariance to lighting
Histograms give some robustness to distortion

Poor at object/occluding boundaries and large changes in viewpoint.

(iv) Nearest neighbour of 128D descriptors: Accept match if $\frac{(x_2 \cdot x)}{(x_1 \cdot x)} < 0.$
k-d trees

Q3.

(a)

(i) General motion in a 3D scene.

$$F = [K]^{-T}[T_x][R][K]^{-1}$$

where $[T_x] = \begin{bmatrix} 0 & -T_z & T_Y \\ \cancel{T_z} & 0 & -T_x \\ -T_Y & T_x & 0 \end{bmatrix}$ skew-symmetric matrix

(b) Looking at a planar object

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} 3\times3 \\ H_1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \underbrace{\overset{\text{2 column of } R}{\begin{bmatrix} R & T \end{bmatrix}}}_{3\times3} = [H_1]$$

$$\begin{bmatrix} su' \\ sv' \\ s \end{bmatrix} = \begin{bmatrix} 3\times3 \\ H_2 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \underbrace{\overset{\text{2 column}}{\begin{bmatrix} R' & T' \end{bmatrix}}}_{3\times3} = [H_2]$$

$$\therefore \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{ij} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad \text{where } H = H_2 H_1^{-1}$$

Rotation about optical centre (eg mosaic)

$$H = [K][R][K]$$

Q2.a-ii

Note $R = I$ and $K' = K$. Using the property in the question the following holds up to scale:
$K^{-\top} [\mathbf{t}]_\times RK^{-1} = K^{-\top} [\mathbf{t}]_\times K^{-1} = K^{-\top}K^\top [K\mathbf{t}]_\times = [\mathbf{e}]_\times$.

Since translation is parallel to x-axis $[\mathbf{e}]_\times = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$.

Hence, $\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0.$

It follows $v = v'$.

If a pair of stereo cameras is in the aforementioned setup, pixel depth can be recovered purely from disparities (differences in location) of matching pixels situated on the same row in the left and right images. Also matching along horizontal epipolar lines can be performed more accurately and efficiently as there is no need to sample different orientations of patches.

(c)(i) $N=4$ for homography
$N=8$ for fundamental matrix

(ii) RANSAC
- Random sample (min) N pairs of correspondences $\left(\begin{array}{l} N=4 \text{ for } H \\ N=8 \text{ for } F \end{array}\right)$
- compute H or F
- check for inliers.
- accept if inliers > max.

(d)(i) Conic section:
$$au^2 + buv + cv^2 + du + ev + f = 0$$

We can re-write in homogenous co-ordinates

$$[u, v, 1] \underbrace{\begin{bmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{bmatrix}}_{C} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

(ii) $\therefore [u\ v\ 1]\ C \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$

After change viewpoint $\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = H \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$

By substitution $\therefore [u'\ v'\ 1]\ H^{-1}\ C\ H \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = 0$

This is still a conic section. Circle → ellipse.

$H' = H^{-1} C H$

Q2.

(a) Pin-hole camera, no non-linear distortion.

(i)

$$\underline{X}_c = R\underline{X} + \underline{T} \qquad \text{Rigid body}$$

$$\begin{bmatrix} \underline{x}_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0\,0\,0 & 1 \end{bmatrix} \begin{bmatrix} \underline{X} \\ 1 \end{bmatrix}$$

perspective

$$\begin{bmatrix} \alpha x \\ \alpha y \\ \alpha \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \underline{X}_c \\ 1 \end{bmatrix}$$

$$x = \frac{fX_c}{Z_c}$$

$$y = \frac{fY_c}{Z_c}$$

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha x \\ \alpha y \\ \alpha \end{bmatrix} \qquad \text{CCD pixel scaling}$$

$$\therefore \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} K \end{bmatrix} \begin{bmatrix} R & T \end{bmatrix} \underset{4\times 1}{\begin{bmatrix} \underline{X} \\ \vdots \\ 1 \end{bmatrix}}$$

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \underbrace{\begin{bmatrix} 3\times 4 \end{bmatrix}}_{\substack{p_{ij} \\ \text{projection matrix}}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

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

(ii) 3D calibration points :— span large field of view

— easy to localise accurately and match

— must NOT be coplanar or linear

N >

11 unknown params, . Need $5\frac{1}{2}$ pts or $N \geq 6$

Q3(b)(iii).

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

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

Re-arrange to give 2 linear equations in p_{ij}

$$\begin{array}{cc} & 12 \\ 2 & \left[\quad\quad\quad\quad\quad\quad \right] \end{array} \begin{bmatrix} p_{11} \\ \vdots \\ \vdots \\ p_{34} \end{bmatrix} = 0$$

N image pts $\underline{A}\,\underline{p} = 0$ Solve by least-squares

$$\lambda_1 \leq \frac{p^T A^T A p}{p^T p} \leq \lambda_{12}$$

Find smallest eigenvector corresponding to λ_1 of $A^T A$.

or Look at SVD.

Q3 (a)(iv)

Need to minimise measurement error - projection error from model $(\hat{u}_i, \hat{v}_i)$

$$\therefore \min_{f} \sum_{i=1}^{N} (u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2$$

Non-linear optimisation.

Recover 3×4 matrix

Decompose

$3\times3 = KR$ by QR decomposition

Estimate $T = K^{-1}\begin{bmatrix} p_{14} \\ p_{24} \\ p_{34} \end{bmatrix}$

$$K = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\therefore f = \frac{\alpha_u}{k_u}$$ (need pixel size)

need to know

Version IB/1

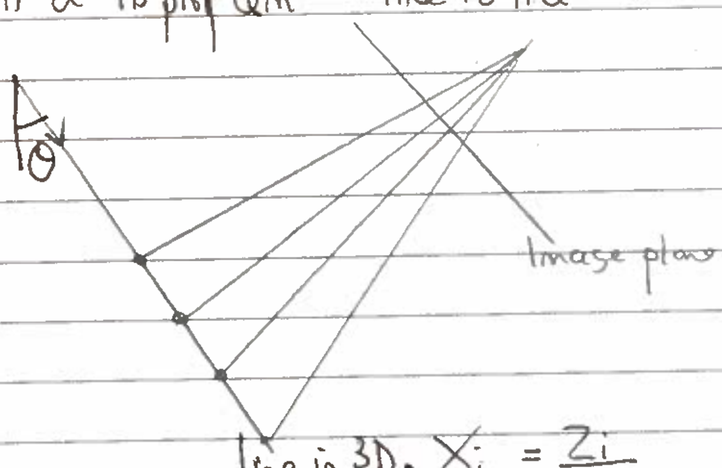
Q3.a-v

Modern mobile phones come with pre-calibrated cameras - the CCD calibration matrix $\mathbf{K}$ is known. World plane to image plane homography $\mathbf{H}$ can be recovered with only 4 points of planar marker.
A point on a world plane is projected to the image as follows:

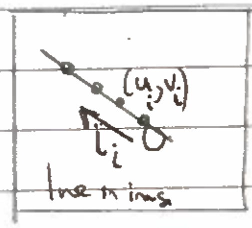
$$\mathbf{w}' = \mathbf{K}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{T} \end{bmatrix}\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathbf{K}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{T} \end{bmatrix}\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}.$$

Hence $\mathbf{H} = \lambda\mathbf{K}\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{T} \end{bmatrix}$.
$\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{T}$ can be extracted from $\mathbf{K}^{-1}\mathbf{H}$. By normalising $\mathbf{r}_1$ and $\mathbf{r}_2$ to be unit lenght the correct scale is obtained for $\mathbf{T}$.
While $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$.
Since $\mathbf{R}' = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix}$ may not be a proper rotation matrix, SVD can be used to obtaining the closest rotation matrix $\mathbf{R}$ to its measurement.
Finally: $\mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \end{bmatrix}$.

Q3 b) This is a 1D problem — line to line



$X_i = \frac{Z_i}{\cos\theta}$



In general 1D $\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} & 3\times 2 & \end{bmatrix}\begin{bmatrix} X \\ 1 \end{bmatrix}$ WLOG (let Z=0, Y=0)

(i) we can re-parametrize along line in image $\begin{bmatrix} sL \\ s \end{bmatrix} = \begin{bmatrix} & 2\times 2 & \end{bmatrix}\begin{bmatrix} kZ \\ k \end{bmatrix}$

$= \begin{bmatrix} & \end{bmatrix}\begin{bmatrix} Z \\ 1 \end{bmatrix}$

(ii) Need 3 points to calibrates i.e. 3 pastings of know thickness
(~~or 3 blocks of known thickness~~, Z)

Predict thickness by inverting matrix

$\begin{bmatrix} Z_i \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} p & q \\ r & s \end{bmatrix}^{-1}}_{\text{Invertible}} \begin{bmatrix} L_i \\ 1 \end{bmatrix}$

4. (a) The convolutional layers CONV1 and CONV2 extract translation invariant features from an image.

The pooling layers MAX-POOL1 and MAX-POOL2 perform image subsampling in order to encourage learning of feature hierarchies and to reduce number of parameters.

The fully connected layer FC1 forms final features of our proposed network. Note that FC1 layer features are not translation invariant.

The use of non-linear activation functions such as Rectified Linear Unit (ReLU) enables the network to learn complex (non-linear) decision boundaries.

The architecture is finalised by adding the fully connected FC2 layer with a Softmax non-linearity. Using this layer corresponds to applying a softmax classifier to the output of layer FC1.

Softmax activation function constraints network output to correspond to a probability distribution over ten class labels. [20%]

(b) Detailed calculation of the output shape (OS) of each layer and the corresponding number of parameters (P).

(CONV1, K = 5×5, S = 1, C = 32, A = ReLU) - OS = $28\times28\times32$, P = $5\times5\times1\times32+32 = 832$.
(MAX-POOL1, K = 2×2, S = 2) - OS = $14\times14\times32$, P = 0.
(CONV2, K = 5×5, S = 1, C = 48, A = ReLU) - OS = $14\times14\times48$, P = $5\times5\times32\times48+48 = 38448$.
(MAX-POOL2, K = 2×2, S = 2) - OS = $7\times7\times48$, P = 0.
(FC1, C = 256, A = ReLU) - OS = 256, P = $7\times7\times48\times256+256 = 602368$.
(FC2, C = 10, A = Softmax) - OS = 10, P = $256\times10+10 = 2570$.

Total number of parameters: 644218. [15%]

(c) Classification accuracy. Too many parameters may make the network overfit to the training data RESULTING IN ~~preventing leading to~~ a poor performance on test data. Too few parameters may make the network not expressive enough for solving the problem of ~~choice~~. INTEREST. Computational efficiency. Too many parameters may prevent the network from fitting into GPU memory or make it too slow.

VGG-16 reduces the number of parameters by using small 3×3 convolutional filters and by frequent application (every 2 or 3 convolutional layers) of max-pooling based subsampling of the outputs of preceding layers. [20%]

(d) (i) Relative cross-entropy can be used as objective function:

$$G(W) = -\sum_{n=0}^{N-1}\sum_{c=0}^{9} t_c^{(n)} \log y_c^{(n)}$$

Here N is a total number of training images, $t_c^{(n)}$ is a one-hot encoded ground truth class label for n-th training image and W is a set of weights $\{w_{0,0} \ldots w_{255,9}\}$ of the fully connected layer FC2. [10%]

(ii) Objective function $G(W)$ can be rewritten as:

$$G(W) = -\sum_{n=0}^{N-1}\sum_{c=0}^{9} t_c^{(n)} \log\left(\frac{\exp\left(\sum_{i=0}^{255} x_i^{(n)} w_{c,i} + b_c\right)}{\sum_{k=0}^{9} \exp\left(\sum_{i=0}^{255} x_i^{(n)} w_{k,i} + b_k\right)}\right) =$$

$$= -\sum_{n=0}^{N-1}\sum_{c=0}^{9} t_c^{(n)} \left[\left(\sum_{i=0}^{255} x_i^{(n)} w_{c,i} + b_c\right) - \log\left(\sum_{k=0}^{9} \exp\left(\sum_{i=0}^{255} x_i^{(n)} w_{k,i} + b_k\right)\right)\right]$$

Hence, we have:

$$\frac{dG(W)}{dw_{c,i}} = -\sum_{n \in N_1}\left[x_i^{(n)} - y_c^{(n)} x_i^{(n)}\right] - \sum_{n \in N_0}\left[-y_c^{(n)} x_i^{(n)}\right] = -\sum_{n=0}^{N-1}(t_c^{(n)} - y_c^{(n)})x_i^{(n)}$$

Here N_1 corresponds to a set of data points for which $t_c^{(n)} = 1$ and N_0 corresponds to a set of data points for which $t_c^{(n)} = 0$.

Note that students were not explicitly shown how to calculate derivatives for the relative cross-entropy objective function during lectures. [25%]

(e) A batch normalization layer should be added. It increases networks ability to fit training data (convergence speed) by simplifying optimization procedure. In particular, it normalises the outputs of the convolutional layers CONV1 and CONV2 so that output vectors of these layers have zero mean and unit variance for each batch. Note that the answer cannot be a dropout layer since it would result in an even longer training time, if applied. [10%]

Engineering Part IIB 2021

Module 4F12 (Computer Vision) Assessor's Report

1. **Gaussian smoothing, bandpass filtering and SIFT**. Attempted by 74/81 Part IIB candidates, average mark 13.9/20.

   The first part of the question covering convolution with low pass filters was generally well answered. Second part convering image pyramid construction and scale estimation was answered particularly well. Some marks were lost in the third part covering SIFT descriptor invariance to lightning and viewpoint changes. Many students missed the verfication step performed in SIFT feature matching.

2. **Epipolar geometry and stereo vision**. Attempted by 63/81 candidates, average mark 13.6/20.

   Parts covering epipolar geometry (a), 2D projective transformation (b) and transformation estimation from point correspondences (c) were mostly well answered with occasional marks lost for lack of precision or detail: e.g. determining the right number of degrees of freedom (DoF) but using a wrong number of constraints provided point to compute total number of point correspondences needed. Candidates displayed a particularly good understanding of RANSAC algorithm. Part (d-i) of the question on conic sections was found easy by most candidates while many struggled to derive the equation for conic section in the second viewpoint in part (d-ii).

3. **Perspective projection and camera calibration**. Attempted by 77/81 candidates, average mark 13.9/20.

   Part (a) was well answered by most of the students. They demonstrated a particularly good knowledge of perspective projection and the key steps required for calibration with a known 3D object. Marks were lost in part (a-v) as only a handfull of students noticed that in order to recover projection matrix from a single image of a known planar object, the knowledge of intrisic parameters of the camera (e.g. mobile phone) is required. Most of the students noticed that part (b-i) covered the modelling of a line to line projection. Marks were lost in providing the details of how the wood chip thickness can be recovered using this projection model in part (b-ii).

4. **Image classification with convolutional neural networks**. Attempted by 27/77 candidates, average mark 14.1/20.

   Many candidates that attempted this question made excellent progress. Most of the marks were lost to mistakes in computing the derivative of the loss function with respect to model parameters.