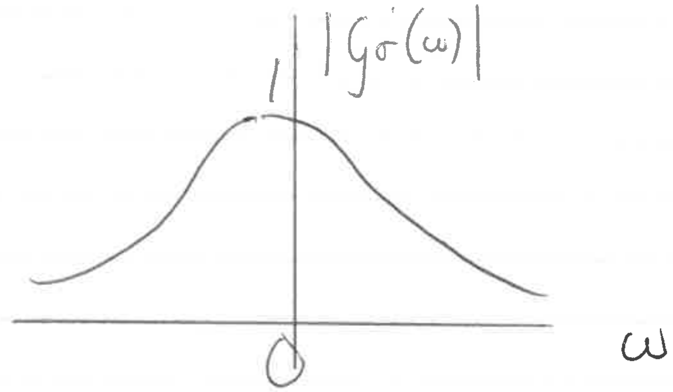
4F12 Computer Vision (2025)

Q1(a)(i) Smoothing
- reduce high spatial frequency before differentiati
- choice of scale (low-pass cut-off frequency)

Gaussian filter - low-pass filter as seen from Fourier Transform

$$g_\sigma(x) \longleftrightarrow e^{-\frac{\omega^2\sigma^2}{2}}$$



$$|G(j\omega)| = 1 \text{ at } \omega = 0$$
$$= 0 \text{ at } \omega = \infty$$

2

- smooth in frequency and time domain
- σ controls scale and cut-off freq $\propto \frac{1}{\sigma}$

Q1 a(ii) Compute $S(x, y, \sigma_i) = I(x,y) * G_{\sigma_i}(x,y)$

$= I(x,y) * g_{\sigma_i}(x) * g_{\sigma_i}(y)$

— Implement as 2 × 1D convolutions

— Discrete set, s per octave

$\sigma_i = 2^{\frac{i}{s}} \sigma_0$

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

— Incremental blurs:

$$g(\sigma_{i+1}) = g(\sigma_i) * g(\sigma_{K_i})$$

$$\sigma_{K_i} = \sigma_i \sqrt{2^{\frac{2}{s}} - 1}$$

— sub-sample (reduce to $\frac{1}{4}$ size) after scale doubles to generate next octave.

— apply same $g(\sigma_{K_i})$ to sub-sampled images

(iii) $S=3$ Need $g(\sigma_0)$, $g(\sigma_{K_0})$, $g(\sigma_{K_1})$ and $g(\sigma_{K_2})$

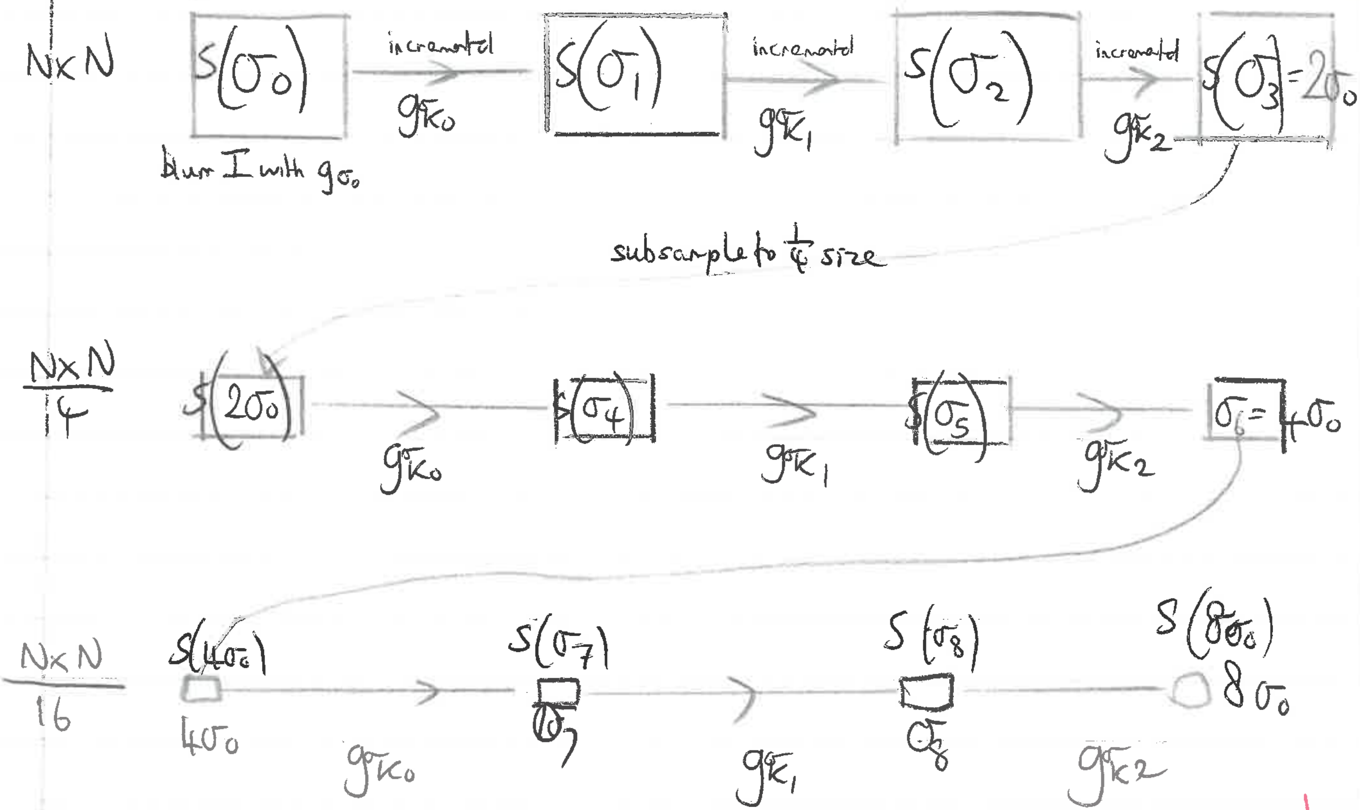
where $g(\sigma_{K_0})$, $\sigma_{K_0} = \sigma_0 \sqrt{2^{\frac{2}{3}} - 1}$

$\sigma_{K_1} = \sigma_0 2^{\frac{1}{3}} \sqrt{2^{\frac{2}{3}} - 1}$

$\sigma_{K_2} = \sigma_0 2^{\frac{2}{3}} \sqrt{2^{\frac{2}{3}} - 1}$

2

Q1(a)(iii) Image pyramid — 4 kernels needed



1

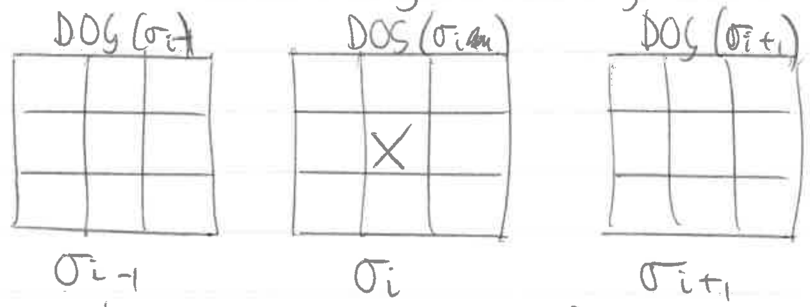
Q1(b)(i) ① scaled-normal laplacian of gaussian $= \sigma_i^2 \nabla^2 G_{\sigma_i}$

or ② difference of gaussian $DOG(\sigma_i) = G_{k\sigma_i} - G_{\sigma_i}$ 2

Show it is a band-pass filter by looking at the Fourier Transform $\propto \omega^2 G_{\sigma_i}(j\omega)$

(DOG) - $\sigma_i^2 \nabla^2 G_{\sigma_i} \simeq G_{k\sigma_i} - G_{\sigma_i}$ where $k \simeq 1.2 - 1.5$

(ii) Find max/min of band-pass-filtered image over position (x_j, y_k) and scale σ_i looking at 26 neighbours in $\sigma^2 S''(x, y, \sigma_i)$ or $DOG(\sigma_i)$



2

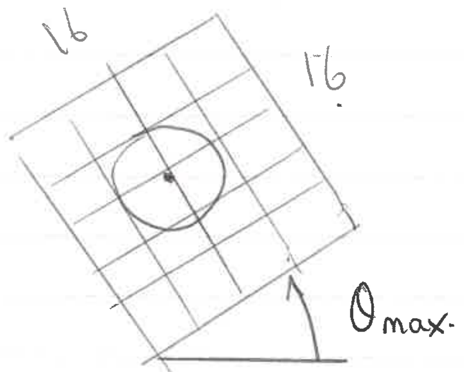
σ_i of max determine blob size $r_i = \sqrt{2}\sigma_i$

Q1(c)(i) - Sample 16×16 pixels from $S(x_k, y_k, \sigma_i)$
(centred on blob centre on σ_i image)
(normalize for scale)

- Compute $\nabla S(x, y, \sigma_i)$

- Histogram (10° bins) and smooth and choose θ_{max} as dominant orientation

- Sample 16×16 from $S(x, y, \sigma_i)$ aligned with θ_{max}
(normalize for orientation)



Sample $S(x, y, \sigma_i)$ image at θ_{max}

SIFT = 128D $\underline{v}$

(ii) scale - σ_i
orientation - θ_{max}
] geometric invariance

lighting - use gradients
normalize descriptor to be unit vector $|\underline{v}|$
truncate any value greater than 0.2 and renormalize
] photometric invariance

bins and histograms allow small mis-alignment

edges encode 2D shape information of fore-ground.
local ≡ occlusion

Q1 (iii) limitations — not on occluding boundary

— large viewpoint change. (>30°) 2

— computationally expensive ,

2(a)(iii).

Vanishing point: Perspective projection of a point at ∞ in direction ~~Point~~ parallel lines.

$$\underset{\sim}{\tilde{V}}_{Pz} = \begin{pmatrix} p_{13} \\ p_{23} \\ p_{33} \end{pmatrix} \quad \begin{pmatrix} u_{\infty} \\ v_{\infty} \end{pmatrix} = \begin{pmatrix} \frac{p_{13}}{p_{33}} \\ \frac{p_{23}}{p_{33}} \end{pmatrix}$$

1

(iv.) Horizon:

Perspective projection of // planes at ∞

1

$$\underset{\sim}{\tilde{VP}}_X = \begin{pmatrix} p_{11} \\ p_{21} \\ p_{31} \end{pmatrix} \qquad \underset{\sim}{\tilde{VP}}_Y = \begin{pmatrix} p_{12} \\ p_{22} \\ p_{23} \end{pmatrix}$$

$$\text{horizon, } \underset{\sim}{\tilde{l}} - \underset{\sim}{\tilde{VP}}_X \times \underset{\sim}{\tilde{VP}}_Y$$

$$= \begin{vmatrix} \underline{i} & \underline{j} & \underline{k} \\ p_{11} & p_{21} & p_{31} \\ p_{12} & p_{22} & p_{23} \end{vmatrix}$$

2

since $\underset{\sim}{\tilde{VP}}_X \cdot \underset{\sim}{\tilde{l}} = 0$ and $\underset{\sim}{\tilde{VP}}_Y \cdot \underset{\sim}{\tilde{l}} = 0$

Q2(b)(i) Camera calibration

Use known $\{(X_i, Y_i, Z_i)\}_{i=1}^{N}$ for $N \geq 6$

and measured $\{(u_i, v_i)\}_{i=1}^{N}$

to estimate 12 elements p_{jk} up to unknown scale 2

(ii). Each point constrains $\underline{p}_{jk}$ such that

$$\begin{bmatrix} X_1, & Y_1, & Z_1, & 1 & 0, & 0, & 0, & 0, & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0, & 0, & 0, & 0 & X_1 & Y_1 & Z_1 & 1, & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ \vdots \\ p_{33} \\ p_{34} \end{bmatrix} = \underline{0}$$

$N \geq 6$

Solve

$$\begin{bmatrix} A \end{bmatrix} \begin{bmatrix} \underline{p} \end{bmatrix} \simeq \underline{0}$$

Solve by finding $\lambda_1 < \dfrac{\underline{p}^T A^T A \underline{p}}{\underline{p}^T \underline{p}} < \lambda_{12}$

ie. smallest λ_1 and corresponding unit eigenvector $\underline{p}_1$

Q2(b)(ii)

Start with the linear "least-squares" sol^n and minimize reprojection error by searching for p_{ij} using gradient descent:

$$\min_{p} \sum_{N} (u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2$$

where
$$\begin{bmatrix} \hat{u}_i \\ \hat{v}_i \\ 1 \end{bmatrix} = \begin{bmatrix} p_{ij} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

$$\hat{u}_i = \frac{p_{11} X_i \dots}{p_{31} X_i + \dots}$$

$$\hat{v}_i = \frac{p_{21} X_i \dots}{p_{31}}$$

2(c) For plane
$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \underbrace{\begin{bmatrix} 3 \times 3 \end{bmatrix}}_{\begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{23} & p_{34} \end{bmatrix}} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix}$$

H

Circle in world. $[X_i \ Y_i \ 1] \begin{bmatrix} C \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} = 0$

(c)(i) Under perspective projection:

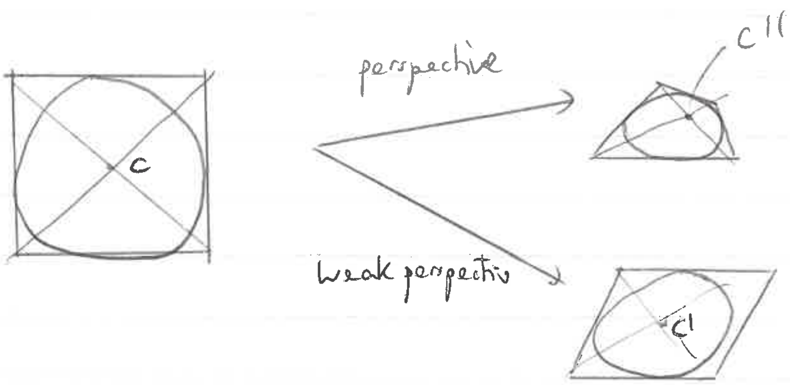
$$\underset{\sim}{\tilde{w}} = H \underset{\sim}{\tilde{X}}$$

$$\underset{\sim}{X}^T C \underset{\sim}{X} = 0$$

$$\underbrace{\underset{\sim}{\tilde{w}}^T H^{-T} C H^{-1} \underset{\sim}{\tilde{w}}}_{\text{equation of conic (ellipse)}} = 0$$

2

(ii) Under weak-perspective parallel lines project to parallel lines and centroid is preserved as projection of centre. Look at 5 pts



2.

$$H_{WP} \rightarrow \begin{bmatrix} x & x & x \\ x & x & x \\ 0 & 0 & x \end{bmatrix}$$

Q3.

(a)(i) Epipolar constraint: for any point in image the e.c gives a line in the other image on which the correspondence is constrained to lie $\longrightarrow$ it is the projection of the ray defined by the feature in the other view 2

stereo – used for searching for correspondence. Use known F.
SFM – unknown F. Find from ~~N~~ correspondences

(a)(ii) $\underline{X}' = R\underline{X} + \underline{T}$

$$\underline{T}_x \underline{X}' = \underline{T}_x R \underline{X} + \underbrace{\underline{T}_x \underline{T}}_{0}$$

Rays and baseline are co-planar:

$$\underline{X}' \cdot \underline{T}_x R \underline{X} = 0$$ 2

Define $\underline{T}_x = [T_x] = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$ such that $E = [T_x]R$,

$$\therefore \underline{X}'^T E \underline{X} = 0$$

Since rays $\underline{p}' \,//\, \underline{X}'$ and $\underline{p} \,//\, \underline{X}$ where $\underline{p} = \frac{f\underline{X}}{Z}$

we can re-write as $\underline{p}'^T E \underline{p} = 0$

or. using $\tilde{\underline{\omega}} = K\underline{p}$ $\quad \tilde{\underline{\omega}}'^T K^{-T} E K \tilde{\underline{\omega}} = 0$ 2

Q3a(ii) Let $F = K^{-T} E K$ be the fundamental matrix

$\therefore$ $\tilde{\omega}'^T F \omega = 0$ or $[u' \; v' \; 1] \begin{bmatrix} F \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$

epipolar constraint

(iii) Consider $\underline{\omega} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$ then $l'^T \underline{\tilde{\omega}}' = 0$ in right view

or $\underline{\tilde{\omega}}'^T \underline{l}' = 0$

but $\underline{\tilde{\omega}}'^T F \underline{\omega} = 0$ so where $\underline{\tilde{l}}' = F \underline{\omega}$

epipolar line in right view

$\underline{\hat{l}}'$ is line in right view on which correspondence $\underline{\tilde{\omega}}'$ must lie

Epipole in right image $F^T \underline{\tilde{\omega}}'_e = 0$ since $\underline{\tilde{e}}'^T F \underline{\tilde{\omega}} = 0$ for all points

$F^T \underline{\omega}'_e = 0$

Q.3(b)(i)

SIFT — each keypoint gives a 128D descriptor $\underline{x}_m$

- find blob centre and dominant orientation
- compute 16x16 gradients and 16 HOGs
- 128D normalised and truncated to remove outliers
- $|\underline{x}_m| = 1$

Normalised — intensities

- 16x16 pixels
- normalize by subtracting μ and dividing by σ
- descriptor is 256D = patch of pixels

(ii) Matches: Find nearest neighbours — search $\underline{x}_i$ using a k-d tree

$$d_{im} = \sum_i^{128} (x_i - x_m)^2 = \|\underline{x}_i - \underline{x}_m\|^2$$

Look at closest points in euclidean distance

Find closest and second closest $\underline{x}_1$ and $\underline{x}_2$

and accept $\underline{x}_i$ if:

$$\frac{\|\underline{x}_1 - \underline{x}_m\|^2}{\|\underline{x}_2 - \underline{x}_m\|^2} < 0.7$$

RATIO test.

Q14

Q3(b)(ii)

RANSAC for N=8

- randomly sample N
- compute F
- check inliers and count
- accept sol^n if large # inliers
- perform least-square with all inliers

Set up $\underset{9\times9}{A}\ \underset{9\times1}{\underline{f}} = \underline{0}$

- where each correspondance gives a row of A - expand $[u'\,v'\,1][F]\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}=0$ (epipolar constraint)

$$\begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} f_{11} \\ \vdots \\ f_{33} \end{bmatrix} = \underline{0}$$

- Solve by $\lambda_1 \leq \frac{\underline{f}^T A^T A \underline{f}}{\underline{f}^T \underline{f}} \leq \lambda_9$ as smallest eigenvalue/eigenvector.
- Find nearest F matrix that has det F=0 or rank F=2 by SVD.

This is a non-linear constraint.

3 (c) (i) Find F, compute $E = K^T F K$

Decompose E by SVD into orthonormal matrix and skew-symmetry.

$$E = U \Lambda V^T$$

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

$$|\hat{T}| = 1 \quad \pm \hat{T} = \underline{U} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T$$

∴ 4 solⁿ — resolve by triangulation to give positive depths infront of camera,

$|T|$ is unknown and can-not be recovered from pt correspondences leading to a scale ambiguity. We can fix if we know baseline distance (eg from mobile phone IMU).

$P_1 = K[I|0]$ and $P_2 = K[R|T]$

(ii)

Triangulation: For known projection matrices we have give 4 equations in 3 unknowns and solve

$$\overset{4\times4}{\begin{bmatrix} & & \\ & C & \\ & & \end{bmatrix}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0 \quad \text{or} \quad \overset{4\times3}{\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} b \end{bmatrix}$$

2

Solve by least squares

16

Crib for 4F12 Computer Vision 2024/2025

Version: MJ/1

Marking guidelines: words in **blue** are key words that should be mentioned in long-form answers. Notes in **green** are guidelines to use when marking the answer.

Question Q4

4 a)

This is very similar to the second question on examples paper 4 except that there is more than one channel and zero padding of 1. These two elements should be mentioned and handled appropriately.

i)

We begin by embedding $\mathbf{I}_l$ into a $C \times (N+2) \times (N+2)$ matrix with zero padding in all spatial dimension. We then embed all valid $C \times 3 \times 3$ patches into a matrix $\mathbf{X}$ with $9C$ rows and N^2 columns. We then embed the filters $\mathbf{F}_{lk}$ into a matrix $\mathbf{W}$ which has K rows and $9C$ columns. The forward pass of the layer is then given by $\mathbf{O}_l = \mathbf{W}\mathbf{X}$.

ii)

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = \frac{\partial \mathcal{L}}{\partial \mathbf{O}_l} \frac{\partial \mathbf{O}_l}{\partial \mathbf{W}}$$
$$\frac{\partial \mathbf{O}_l}{\partial \mathbf{W}} = \mathbf{X}^T$$
$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = \frac{\partial \mathcal{L}}{\partial \mathbf{O}_l} \mathbf{X}^T$$

$\frac{\partial \mathcal{L}}{\partial \mathbf{F_{lk}}}$ is the kth row of $\frac{\partial \mathcal{L}}{\partial \mathbf{W}}$.

iii)

First we embed $\frac{\partial \mathcal{L}}{\partial \mathbf{O}_l}$ into a $C \times (N+2) \times (N+2)$ tensor $\mathbf{D}$ with zeros padding the border across all channels. We then embed all valid $C \times 3 \times 3$ patches from $\mathbf{D}$ into a matrix Y with $9C$ rows and N^2 columns. We then take each filter F_{lk} and rotate it spatially by $180°$ to get $\tilde{F}_{lk}$. We then embed $\tilde{F}_{lk}$ into a matrix Z with K rows and $9C$ columns. The partial derivative is then given by $\frac{\partial \mathcal{L}}{\partial \mathbf{I}_l} = ZY$ (with reshaping back into a $C \times N \times N$ tensor).

17

4

b)

i)

First, we should divide the data into **training**, **validation**, and **test** splits. During training, we can augment the data by taking **flipping the image** or taking **random crops**. The data should also be **normalised** to have zero mean and unit variance wrt over a **held-out dataset** (*e.g.* a face dataset) or from only the **training images**.

As we only care about the period of the PPG and not the amplitude, we can normalise each output sequence to range from -1 to 1. This allows a **saturating** output activation like tanh to be used, which will **stablise** training, and will aid in **generalisation**, since each target sequence will have the same range.

ii)

Some people may have more naturally flush skin than others, or have different skin tones that show blood oxygenation differently. Providing the first frame as a reference provides needed **context** to the network, aiding in **generalisation**. Furthermore, the **phase** of the sequence will be different for each sequence and so looking at the difference can provide a way to further **normalise** the data, *e.g.* by making each sequence have the same phase.

c)

i)

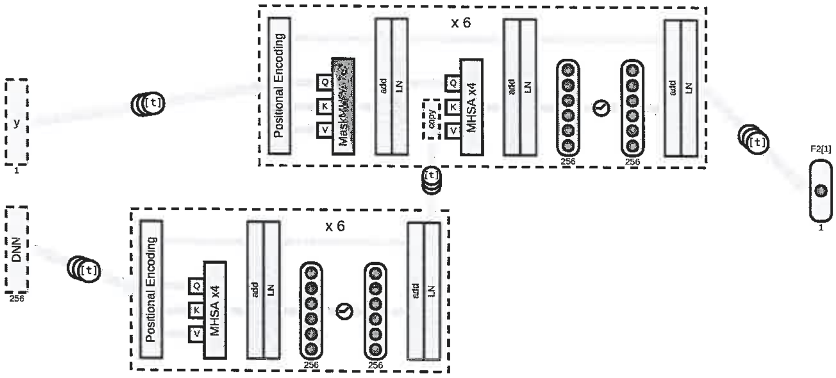
This problem can be posed as a **translation** problem in which the images in the video are being translated into the correlated output values. A transformer can incorporate the **temporal** nature of the data by learning **long-range** and **short-range** dependencies and patterns.

ii)

This diagram contains the maximum level of detail that would be expected of a student. The key elements that need to be depicted are the decoder (above) and the encoder (below). The DNN should feed into the encoder, and the target value should feed into the decoder. Positional encoding should be mentioned. There should be a masked MHSA block and a second MHSA block in the decoder with input from the encoder. The encoder should have a single MHSA block. There should be two linear layers with a non-linearity depicted in both encoder and decoder. The repeating nature should be mentioned in some way. The output should be a single scalar value. If the student has read ahead and made the output two scalars, that is also accept-

18/18

able.



iii)

The network should output a **mean** and **log variance** for each prediction instead of just a single scalar value. The variance gives the network a way to communicate its **uncertainty**, *i.e.* a high variance corresponds to high uncertainty. Beam search is a way to improve the quality of the final output by keeping track of the **k most likely sequences** at each step and then selecting the k most likely sequences overall. This allows the network to **explore different paths** and not get **stuck in a local minimum**. The likelihood of a sequence can be computed as:

$$\mathcal{L}(\mathbf{t}) = \prod_{t_i \in \mathbf{t}} P(t_i \mid \mathbf{t}_{i-1}) \propto \sum_{t_i \in \mathbf{t}} \log P(t_i \mid \mathbf{t}_{i-1})$$

$$P(t_i \mid \mathbf{t}_{i-1}) = \mathcal{N}(t_i \mid \mu_i, \sigma_i)$$

where t_i is a scalar output value and μ_i and σ_i are the outputs of the transformer at step i.

17/5/2025