(i) (ii)

sum = 200

	Prob
S_1 →	0.8
S_2 →	0.1
S_3 →	0.05
S_4 →	0.025
S_5 →	0.025

S_1	0.8
S_2	0.1
S_3	0.05
S_4S_5	0.05

→

$S_1 \rightarrow 0.8$

$S_2 \rightarrow 0.1$

$S_3(45) \rightarrow 0.1$

assign 0

$S_1 \rightarrow 0.8 \Longrightarrow$

$S_2[S_3(S_4S_5)] \rightarrow 0.2 \Longrightarrow 1$

∴ $S_1 \rightarrow$ [0] 1bit

$S_2 \rightarrow$ [10] 2bits

$S_3(45) \rightarrow 11$

$S_3 \Rightarrow$ [110] 3bits

S_4S_5 111

S_4: [1110] 4bits

S_5: [1111] 4bits

2

for Huffman coding

#bits = 1 * 160 + 20 * 2 +
10 * 3 + 5 * 4 + 5 * 4

= 270 bits

for fixed length coding : 5 symbols
we need 3 bits ⇒ 3 * 200 = 600 bits

---

(1) RLC (0,1) ⇒ a 1 preceeded with No zeros

(3,1) ⇒ a 1 preceeded with 3 zeros

(0,0) ⇒ EOB (a zero ⟶ a Run-length of zeros)

---

3 b i

f(4 4) ⇒

0.04	0.065	0.04
0.065	0.1	0.065
0.04	0.065	0.04

⟹

116	79	70
87	200	79
87	200	103

pixel value

Σ multiply & add)

⟹

---

ii) yes because $\boxed{\sum w_{ij} = 1}$

iii) Box filter (3X3) so each weight ⇒ $\frac{1}{9}$

---

c) i) $300 * 300 * 3 * 8$

ii) LUT size = 128 ⇒ 2^7 (7 bits)

so, $(300 * 300 * 7)$ + $(128 * 3 * 8)$ =

$(300 * 300 * 7)$: indexed img. size

$(128 * 3 * 8)$: LUT size

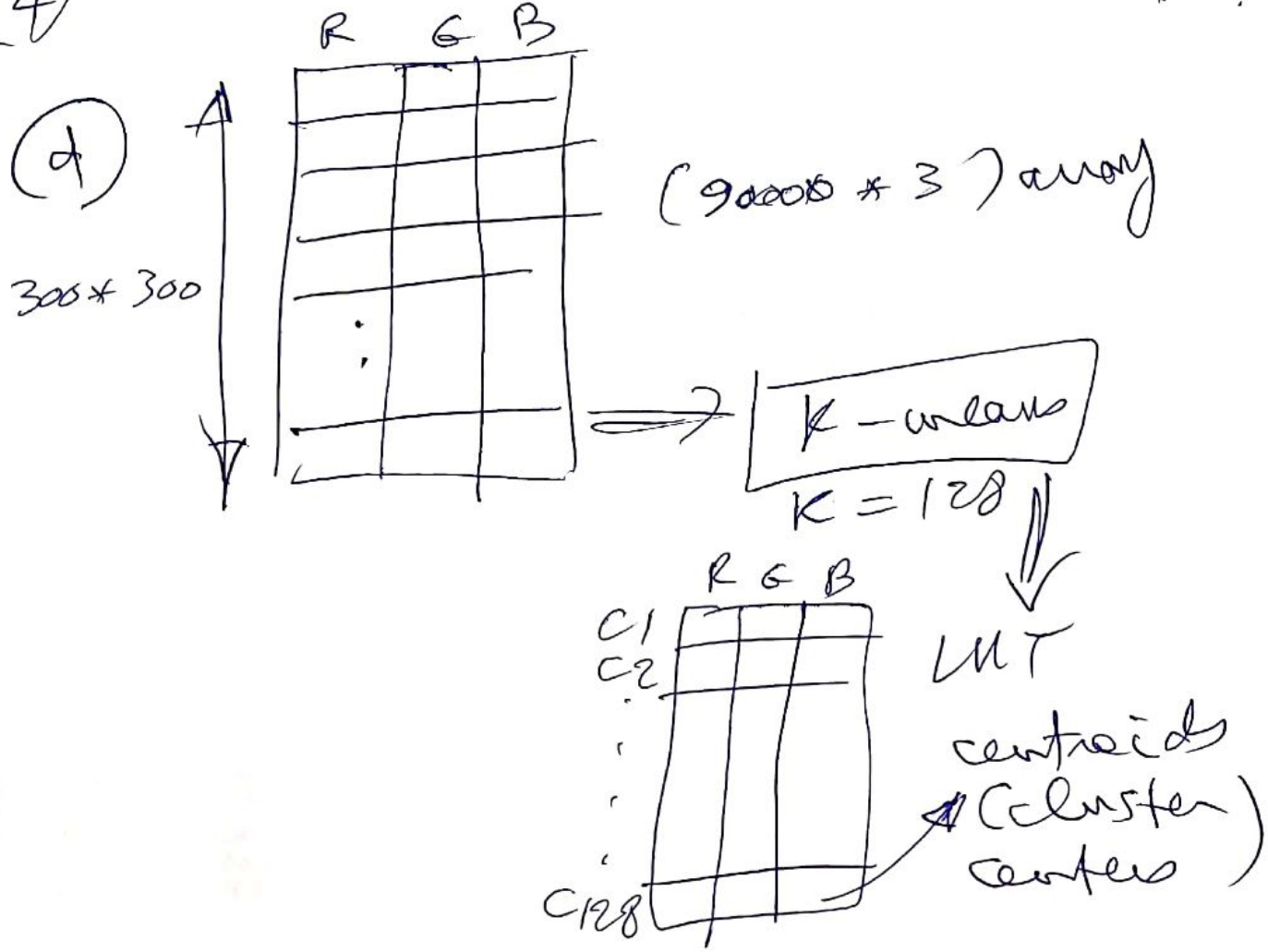
iii) $(300 * 300 * 8)$ + $\left[\frac{1}{4}(300 * 300 * 8)\right] * 2$

$(300 * 300 * 8)$: luminance

$\left[\frac{1}{4}(300 * 300 * 8)\right] * 2$: color parts

(4)

(a)



(K-means) ⟹ (a) initialize 128 centers at Random

(b) each data point is assigned to a centroid (Nearest Neighb. Rule)

(c) update each centroid

$$\hat{C}_j = \frac{1}{N_j} \sum_{x \in j} x_j$$

$x \in j$ → data assigned to centroid j

We Run K-means (n) times and find the solution giving Min. (SSE)

$$SSE = \sum_{j=1}^{J} \sum_{i=1}^{ij} (c_j - x_{ij})^2$$

(5) Q2

(a)(i) spatial → freq domain (dct2)

(ii)

(iii) because images mostly contain low-freq. components

(b) JPEG :

(i) because Y → luminance is very important eyes are sensitive to Gray ? (I, Q) ⇒ eye less sensitive so we do sub-sampling to reduce comp. complexity

(ii) (8-8) blocks so each block doesn't have much details ⇒ as smooth as possible → less high-freq components and because the big block would Mix smooth parts with edges.

6

iii) (dct2) used to convert from spatial → freq domain so we separate low from high freq. Compoents, so we can get rid of high-freq. components since eyes are less seasitive to

(iv) $\frac{F(u,v)}{Q(u,v)}$ ⟹ high freq compoent divided by larger values from l. freq

so round⟨·⟩ ⟹ lots of zeros

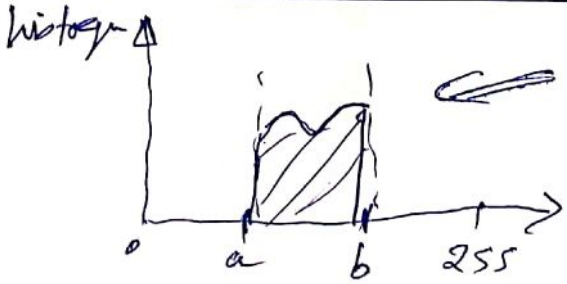
(v) to convert the (Non-zero/zero) values to binary stream ⟹ fixed-length coding not efficient so we use(VLC) ⟹ Huffman

some symbols occur a lot → small codes
" " " Rarely → large codes

100 5 0000
000 3 00
000..10

---

b) i)



(7) $s = (r - r_{min}) * \left(\frac{L-1}{r_{max} - r_{min}}\right)$

↑ new pixel value

$r_{min} \Rightarrow a$

$r_{max} \Rightarrow b$

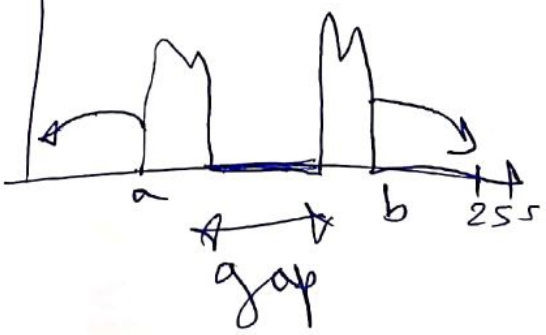
$L = 256$

⟹ contrast stretching

(*) histo-eq ⟹ $\left(s_k = \sum_{j=0}^{k} \frac{n_j}{n}\right)$

⟸ makes sure that histogram is as flat as possible ⟹ with No (Gaps)

in contrast stretching we may have a gap which will grow larger.



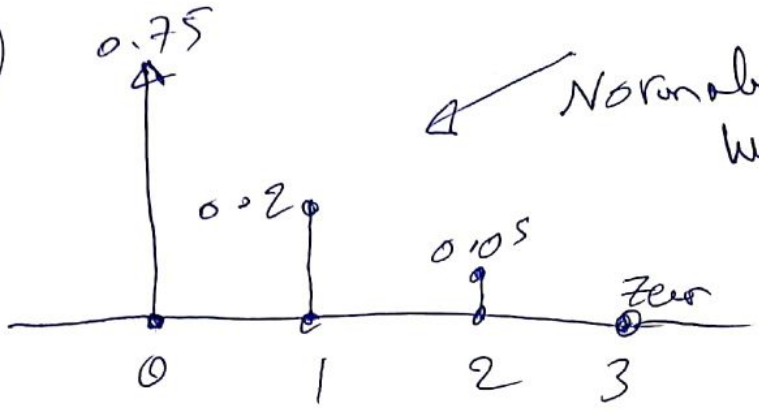
---

(9) for the Given image

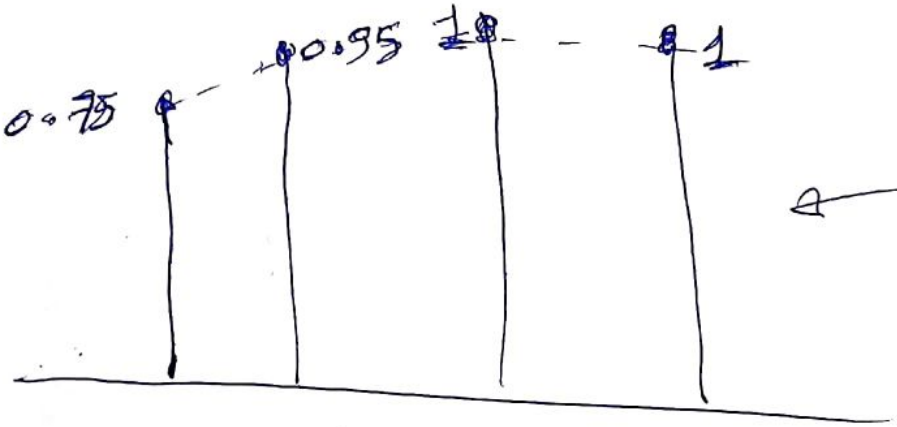
		pdf
0 ⟹	150 ⟶	0.75
1 ⟹	40 ⟶	0.2
2 ⟹	10 ⟶	0.05
3 ⟹	0 ⟶	0

(8)

(ii)



← Normalized histogram



← CDF

(iii)

here $L-1=3$

$$S_0 = 3 * \sum_{j=0}^{0} p(r_j) = 3 * 0.75$$

$$= 2.25 \sim 2$$

all pixels with value zero will be ⟶ 2

$$S_1 = 3 * (0.75 + 0.2) = 2.85 \sim 3$$

all pixels with value (1) ⟶ (3)

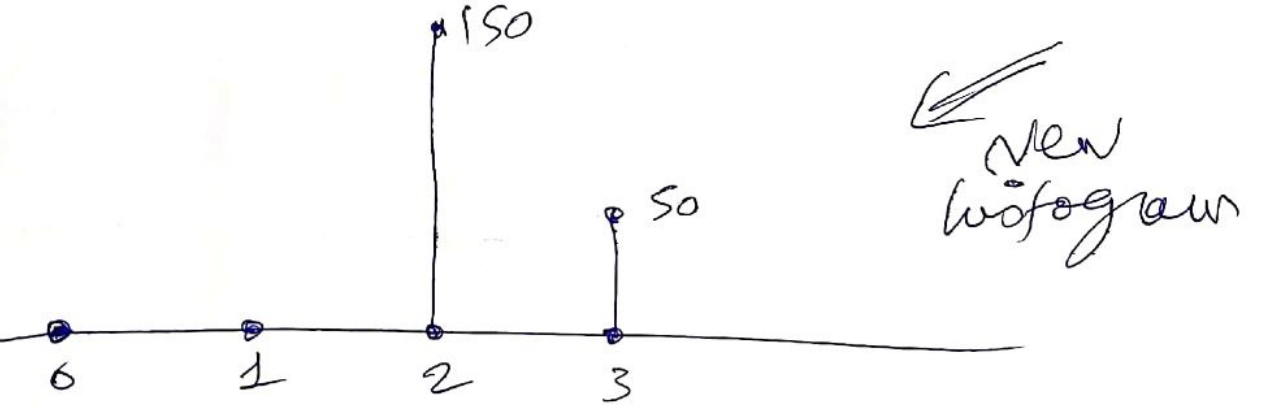
$$S_2 = 3 * (0.75 + 0.2 + 0.05) = \boxed{3}$$

$$S_3 = 3 * (0.75 + 0.2 + 0.05 + 0) = \boxed{3}$$

(9)

New histogram

50 pixel value	No. of occurrences
0	0
1	0
2	150 (r_0)
3	(r_1 40 + r_2 10 + r_3 0) = 50



New histogram