

---

```

%Code to apply Shannon Fano coding to a grayscale image
clc;
clear all;
close all;
I=imread("vaibhav.jpg");
if size(I,3)==3
I=rgb2gray(I);
end
figure
imshow(I);
counts=imhist(I); %Finding frequency of each gray level intensity.
p=counts/sum(counts); %Normalizing histogram counts into probabilities.
symbols=find(p>0)-1; %Extracting only those intensity values that appear.
p=p(p>0); %Removing all zero probability gray levels.
[p_sorted,idx]=sort(p,'descend'); %Sorting probabilities from highest to
lowest.
symbols_sorted=symbols(idx); %Rearranging symbols in the same sorted order.
codes=strings(1,length(symbols_sorted));
%Creating an empty string array to store Shannon-Fano binary codes.
codes=shannon_fano(symbols_sorted,p_sorted,codes,1,length(p_sorted));
%Calling the recursive function that generates Shannon-Fano codes.
disp("Top 20 Shannon-Fano Codes for Image Symbols:");
disp("GrayLevel Probability Code");
disp("-----");
for i=1:min(20,length(symbols_sorted))
fprintf("%3d %.6f %s\n", ...
symbols_sorted(i),p_sorted(i),codes(i));
end
%Displaying only the most frequent gray levels and their corresponding codes.
Lavg=0;
for i=1:length(p_sorted)
Lavg=Lavg+p_sorted(i)*strlength(codes(i));
end
%Computing the average code length using probability weighted sum.
H=0;
for i=1:length(p_sorted)
H=H-p_sorted(i)*log2(p_sorted(i));
end
%Applying Shannon entropy formula  $H=-\sum(p_i \log_2(p_i))$  for binary coding.

disp("-----");
fprintf("Entropy(H)=%.4f bits/pixel\n",H);
fprintf("AverageCodeLength(Lavg)=%.4f bits/pixel\n",Lavg);
fprintf("CodingEfficiency=%.2f %%\n", (H/Lavg)*100);
%Efficiency indicates how close coding is to the theoretical entropy limit.
function codes=shannon_fano(symbols,p,codes,startIdx,endIdx) %shannon fano
recursive function
if startIdx>=endIdx
return;
end
%Stopping recursion when only one symbol remains.
totalProb=sum(p(startIdx:endIdx));

```

---

---

```

%Calculating total probability of the current symbol group.
runningSum=0;
splitIdx=startIdx;
for i=startIdx:endIdx
runningSum=runningSum+p(i);
%Finding cumulative probability until it reaches half of total.
if runningSum>=totalProb/2
splitIdx=i;
break;
end
end
for i=startIdx:splitIdx
codes(i)=codes(i)+"0";
end
%Assigning binary 0 to the first probability subset.
for i=splitIdx+1:endIdx
codes(i)=codes(i)+"1";
end
%Assigning binary 1 to the second probability subset.
codes=shannon_fano(symbols,p,codes,startIdx,splitIdx);
codes=shannon_fano(symbols,p,codes,splitIdx+1,endIdx);
%Recursively repeating the splitting until all symbols get a unique code.
end

```

*Top 20 Shannon-Fano Codes for Image Symbols:*

*GrayLevel Probability Code*

---

```

114 0.009462 0000000
105 0.009158 0000001
119 0.009092 0000010
121 0.008998 0000011
118 0.008982 0000100
107 0.008873 0000101
108 0.008828 0000110
113 0.008821 0000111
122 0.008749 0001000
110 0.008691 0001001
117 0.008689 0001010
116 0.008579 0001011
112 0.008572 0001100
120 0.008535 0001101
106 0.008430 000111
104 0.008429 0010000
109 0.008344 0010001
123 0.008338 0010010
103 0.008328 0010011
115 0.008306 0010100

```

---

*Entropy( $H$ )=7.7299 bits/pixel*

*AverageCodeLength( $L_{avg}$ )=7.7929 bits/pixel*

*CodingEfficiency=99.19 %*



*Published with MATLAB® R2025b*