```
%Code to apply Shannon Fano coding to a grayscale image
clc;
clear all;
close all;
I=imread("C:\Users\hp\Downloads\images.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=-#(p*log2(p)) for binary coding. 1
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
--------------------------------------
254 0.033403 000000
253 0.029786 000001
252 0.026110 00001
251 0.022295 00010
 21 0.019116 00011
 19 0.019056 001000
 20 0.017804 001001
 18 0.017387 001010
250 0.016036 001011
 22 0.014943 001100
249 0.014446 001101
 17 0.012817 001110
243 0.012022 001111
248 0.011704 0100000
245 0.010591 0100001
247 0.010094 0100010
 23 0.009240 0100011
 16 0.009160 0100100
246 0.008743 0100101
244 0.008544 010011
--------------------------------------
Entropy(H)=7.5065 bits/pixel
```

```
AverageCodeLength(Lavg)=7.5887 bits/pixel
CodingEfficiency=98.92 %
```

Published with MATLAB® R2021a