
```

%Code to apply Shannon Fano coding to a grayscale image
% Name- Akshay(BT23ECE050)
clc;
clear all;
close all;

I=imread("image_1.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
<hr/>		
31	0.014325	0000000
25	0.014314	0000001
29	0.014167	000001
22	0.014112	000010
30	0.014111	000011

```
17      0.014046    0001000
18      0.014015    0001001
28      0.013948    000101
24      0.013931    000110
26      0.013919    000111
20      0.013888    0010000
13      0.013833    0010001
23      0.013771    001001
15      0.013746    001010
19      0.013723    001011
27      0.013709    001100
16      0.013543    001101
21      0.013521    001110
12      0.013282    001111
32      0.013231    0100000
```

*Entropy(H)=7.3241 bits/pixel
AverageCodeLength(Lavg)=7.4016 bits/pixel
CodingEfficiency=98.95 %*



Published with MATLAB® R2025b