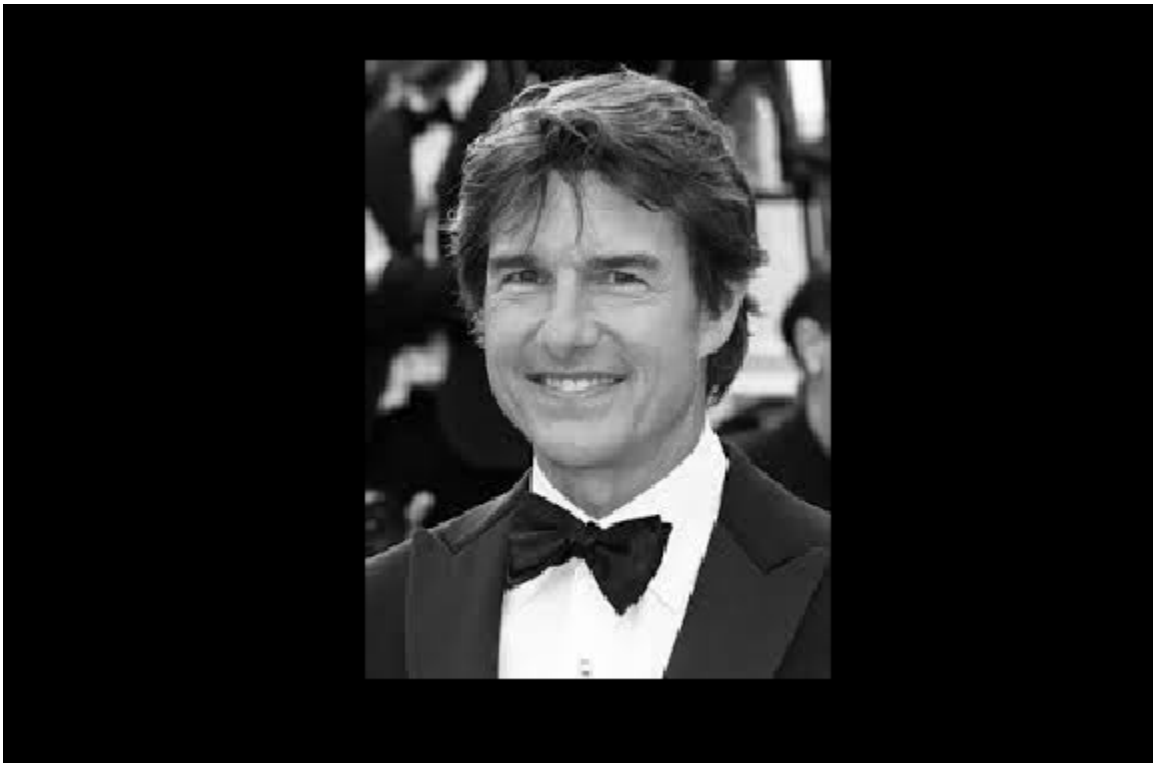
```
%Code to apply Shannon Fano coding to a grayscale image
clc;
clear all;
close all;

I=imread("tom.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 = -Σ(p*log2(p)) 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*
20	*0.013642*	*0000000*
48	*0.013444*	*0000001*
49	*0.013048*	*000001*
52	*0.012514*	*000010*
35	*0.011504*	*000011*
47	*0.011345*	*0001000*

```
 37         0.011048       0001001
 50         0.010672       000101
 54         0.010534       0001100
 56         0.010058       0001101
 53         0.009940       000111
 55         0.009920       0010000
 51         0.009742       0010001
 34         0.009583       0010010
 10         0.009563       0010011
 32         0.009484       0010100
  8         0.009266       0010101
 38         0.009148       001011
 22         0.009049       0011000
 46         0.008831       0011001
-------------------------------------------------
Entropy (H)=7.7062 bits/pixel
AverageCodeLength (Lavg)=7.7754 bits/pixel
CodingEfficiency=99.11 %
```



Published with MATLAB® R2025b