```
% Shannon-Fano Encoding for Grayscale Images
clc;
clear all;
close all;

% Load and prepare image
img_data = imread("Lin-Dan.jpg");

% Convert to grayscale if needed
if size(img_data, 3) == 3
    img_data = rgb2gray(img_data);
end

% Display original image
figure;
imshow(img_data);

% Calculate histogram and probability distribution
intensity_freq = imhist(img_data);
prob_dist = intensity_freq / sum(intensity_freq);

% Extract non-zero intensity levels
active_pixels = find(prob_dist > 0) - 1;
prob_dist = prob_dist(prob_dist > 0);

% Sort by probability (descending order)
[sorted_prob, sort_indices] = sort(prob_dist, 'descend');
sorted_pixels = active_pixels(sort_indices);

% Initialize code storage
binary_codes = strings(1, length(sorted_pixels));

% Generate Shannon-Fano codes
binary_codes = encode_shannon_fano(sorted_pixels, sorted_prob, binary_codes,
1, length(sorted_prob));

% Display results
disp("Top 20 Shannon-Fano Codes for Image Symbols:");
disp("GrayLevel    Probability    Code");
disp("-----------------------------------------");

for k = 1:min(20, length(sorted_pixels))
    fprintf("%3d          %.6f        %s\n", ...
        sorted_pixels(k), sorted_prob(k), binary_codes(k));
end

% Calculate average code length
avg_length = 0;
for k = 1:length(sorted_prob)
    avg_length = avg_length + sorted_prob(k) * strlength(binary_codes(k));
end
```

```
% Calculate entropy
entropy_val = 0;
for k = 1:length(sorted_prob)
    entropy_val = entropy_val - sorted_prob(k) * log2(sorted_prob(k));
end

% Display statistics
disp("----------------------------------------");
fprintf("Entropy (H) = %.4f bits/pixel\n", entropy_val);
fprintf("Average Code Length (L_avg) = %.4f bits/pixel\n", avg_length);
fprintf("Coding Efficiency = %.2f %%\n", (entropy_val / avg_length) * 100);

% Shannon-Fano recursive encoding function
function code_array = encode_shannon_fano(pixel_vals, probs, code_array,
left, right)
    % Base case: single element
    if left >= right
        return;
    end

    % Calculate total probability for current range
    total_weight = sum(probs(left:right));

    % Find optimal split point
    cumulative = 0;
    partition_point = left;

    for j = left:right
        cumulative = cumulative + probs(j);
        if cumulative >= total_weight / 2
            partition_point = j;
            break;
        end
    end

    % Assign '0' to upper partition
    for j = left:partition_point
        code_array(j) = code_array(j) + "0";
    end

    % Assign '1' to lower partition
    for j = partition_point + 1:right
        code_array(j) = code_array(j) + "1";
    end

    % Recursively encode both partitions
    code_array = encode_shannon_fano(pixel_vals, probs, code_array, left,
partition_point);
    code_array = encode_shannon_fano(pixel_vals, probs, code_array,
partition_point + 1, right);
end

Top 20 Shannon-Fano Codes for Image Symbols:
GrayLevel    Probability    Code
```

```
----------------------------------------
  1          0.210962        000
  0          0.095074        001
  3          0.042223        01000
  2          0.042117        01001
  4          0.028201        0101
 15          0.021884        011000
  5          0.017167        011001
 16          0.016694        01101
  6          0.013240        011100
 14          0.012426        011101
  7          0.012316        01111
 13          0.011535        1000000
 17          0.011094        1000001
 20          0.010703        100001
  8          0.010239        1000100
 19          0.010146        1000101
 18          0.009715        100011
  9          0.008736        10010000
 10          0.007622        10010001
 11          0.007085        1001001
----------------------------------------
Entropy (H) = 6.0289 bits/pixel
Average Code Length (L_avg) = 6.1540 bits/pixel
Coding Efficiency = 97.97 %
```

Published with MATLAB® R2025b