
```
% Code to demonstrate Arithmetic Coding for an image
% Date: 29 Jan 2026
clc;
clear all;
close all;

inputImg = imread("Harvey.jpg");
% Reading the input image

if size(inputImg,3) == 3
    inputImg = rgb2gray(inputImg);
end
% Converting RGB image to grayscale if required

figure;
imshow(inputImg);
% Displaying the grayscale image

histCounts = imhist(inputImg);
probabilities = histCounts / sum(histCounts);
% Computing probability of each gray level using histogram normalization

symbolSet = find(probabilities > 0) - 1;
probabilities = probabilities(probabilities > 0);
% Retaining only symbols that appear in the image (non-zero probability)

cumulativeProb = cumsum(probabilities);
lowerBoundProb = [0; cumulativeProb(1:end-1)];
upperBoundProb = cumulativeProb;
% Creating cumulative probability intervals for arithmetic coding

symbolSequence = symbolSet(1:10);
% Selecting a small symbol sequence for demonstration

lowerLimit = 0;
upperLimit = 1;
% Initializing arithmetic coding range [0, 1]

for idxSeq = 1:length(symbolSequence)

    currentSymbol = symbolSequence(idxSeq);
    symbolIndex = find(symbolSet == currentSymbol);
    % Finding index of current symbol in probability table

    intervalRange = upperLimit - lowerLimit;
    % Computing current interval width

    upperLimit = lowerLimit + intervalRange * upperBoundProb(symbolIndex);
    lowerLimit = lowerLimit + intervalRange * lowerBoundProb(symbolIndex);
    % Updating interval bounds based on symbol probability

end
```

```
encodedValue = (lowerLimit + upperLimit) / 2;  
% Selecting a value inside final interval as encoded tag  
  
disp("Arithmetic Coding Output:");  
fprintf("Final Lower Bound = %.10f\n", lowerLimit);  
fprintf("Final Upper Bound = %.10f\n", upperLimit);  
fprintf("Encoded Tag Value = %.10f\n", encodedValue);  
% Displaying final arithmetic code results
```

Arithmetic Coding Output:
Final Lower Bound = 0.3939314176
Final Upper Bound = 0.3939314176
Encoded Tag Value = 0.3939314176



Published with MATLAB® R2025b