
```
% Arithmetic Coding Demonstration for Grayscale Image
% 29 Jan 2026

clc;
clear;
close all;

img = imread("SS.jpeg");

% Convert to grayscale if RGB
if size(img,3) == 3
    grayImg = rgb2gray(img);
else
    grayImg = img;
end

figure;
imshow(grayImg);
title("Input Grayscale Image");

% Histogram and probability calculation
histValues = imhist(grayImg);
probDist = histValues / sum(histValues);

% Extract valid symbols
validSymbols = find(probDist > 0) - 1;
symbolProb = probDist(probDist > 0);

% Cumulative probability ranges
cumDistribution = cumsum(symbolProb);
lowerRange = [0; cumDistribution(1:end-1)];
upperRange = cumDistribution;

% Select a short symbol sequence for demo
symbolSeq = validSymbols(1:10);

% Initialize arithmetic interval
lowBound = 0;
highBound = 1;

for i = 1:length(symbolSeq)

    currentSymbol = symbolSeq(i);
    symbolIndex = find(validSymbols == currentSymbol);

    intervalWidth = highBound - lowBound;

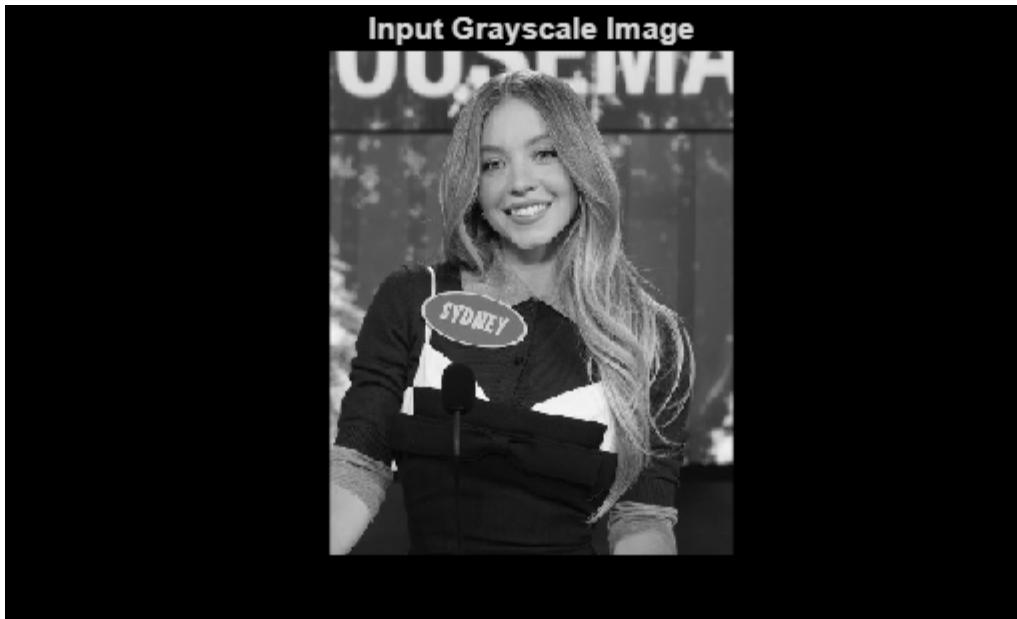
    highBound = lowBound + intervalWidth * upperRange(symbolIndex);
    lowBound = lowBound + intervalWidth * lowerRange(symbolIndex);
end

% Final encoded value
```

```
encodedValue = (lowBound + highBound) / 2;

disp("Arithmetic Coding Output:");
fprintf("Final Low Value = %.10f\n", lowBound);
fprintf("Final High Value = %.10f\n", highBound);
fprintf("Encoded Tag      = %.10f\n", encodedValue);

Arithmetic Coding Output:
Final Low Value = 0.0000000000
Final High Value = 0.0000000000
Encoded Tag      = 0.0000000000
```



Published with MATLAB® R2025b