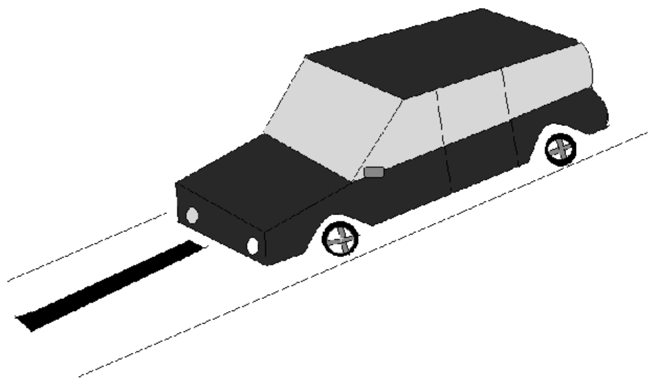
```
clc;
clear;
close all;

% Read image
img = imread("car1.png");

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

imshow(img);

% Histogram and probability
freq = imhist(img);          % frequency of each gray value
prob = freq / sum(freq);     % probability

% Keep only symbols that appear
symbols = find(prob > 0) - 1;
prob = prob(prob > 0);

% Cumulative probability
cumProb = cumsum(prob);
lowRange  = [0; cumProb(1:end-1)];
highRange = cumProb;

% Take first 10 symbols for demo
pixels = img(:);       % flatten image
seq = pixels(1:10);    % real sequence from image

% Initial range
low = 0;
high = 1;

% Arithmetic encoding loop
for i = 1:length(seq)

    currentSymbol = seq(i);
    index = find(symbols == currentSymbol);

    range = high - low;

    high = low + range * highRange(index);
    low  = low + range * lowRange(index);
end

% Final encoded number
encodedValue = (low + high) / 2;

disp("Arithmetic Coding Result:");
fprintf("Lower Bound : %.10f\n", low);
fprintf("Upper Bound : %.10f\n", high);
fprintf("Encoded Tag : %.10f\n", encodedValue);
```

```
Arithmetic Coding Result:
Lower Bound : 0.7333667351
Upper Bound : 1.0000000000
Encoded Tag : 0.8666833675
```



Published with MATLAB® R2021a