**Bonus 1**

1. This is my cat:



Image size 1600x900x3

1. This is a grayscale of my cat:

A picture containing mammal, standing, posing, staring

Description automatically generated

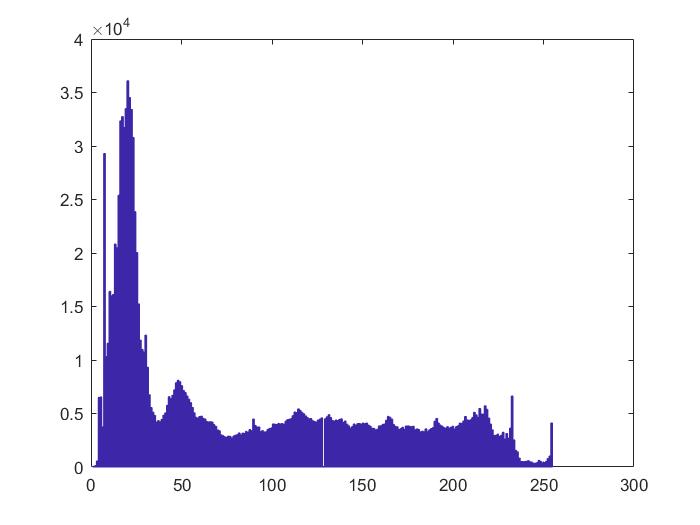
I took the red channel from RGB of his picture

1. Down here you can see a random permute of my cat

A black and white photo of a rug

Description automatically generated with low confidence

1. Histogram of my cat:



We can see the above histogram is of the grayscale of my cat and below is the histogram the of random permute of my cat.   
there are the same.

1. Calculate its entropy:  
     
   J\_I1 = 7.4595

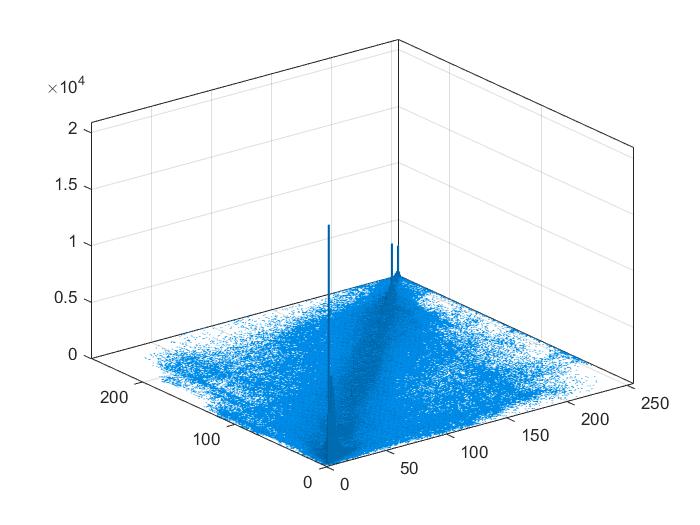
J\_permI1 = 7.4597

we can see almost the same as expected.

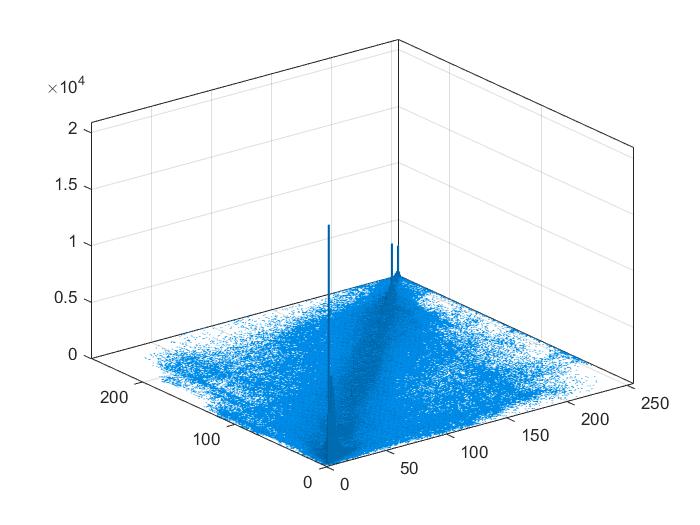
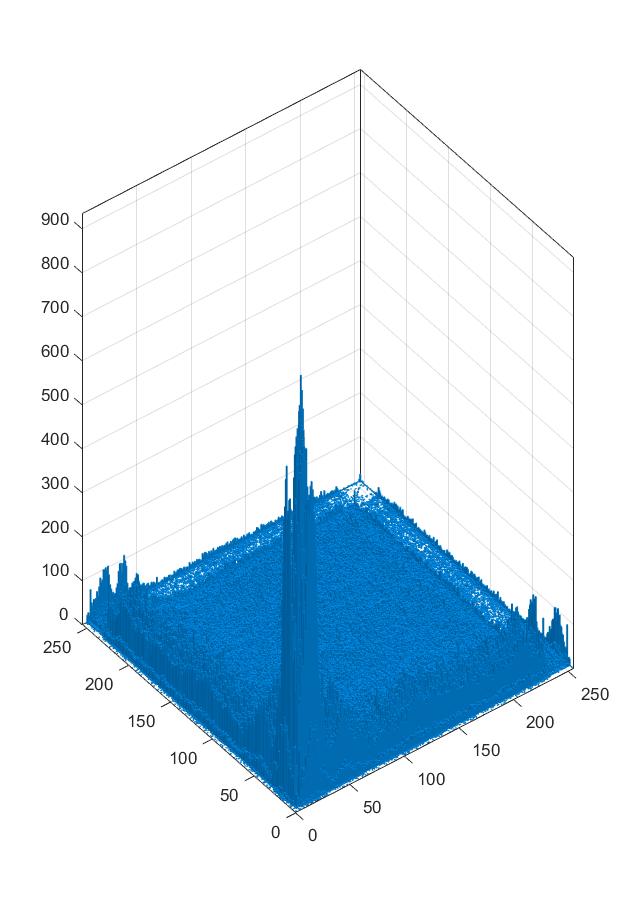
6) Calculate the Mutual Information between I1 pixels and their respective left-neighbors:

First I will show you the 2D histograms.

We can see the 2D histogram is really good it is around the middle.   
  
and we got: Mut\_I1 =



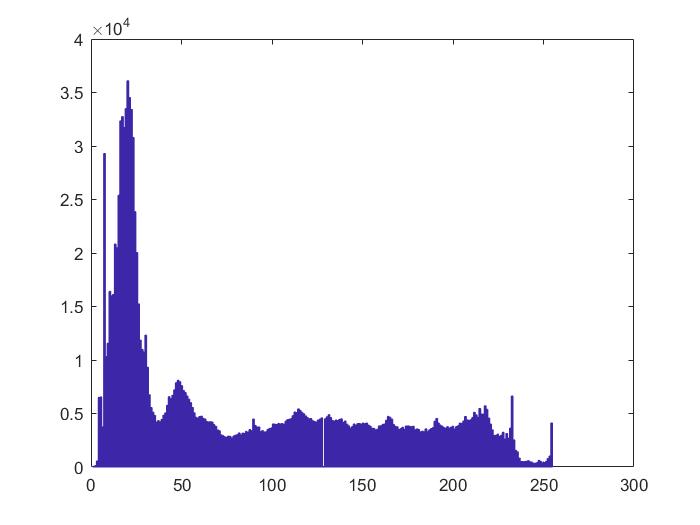
1.0900e+04  
that mine around 10000 bits are the same.   
we did a shift left to my cat image.



7) Calculate the Mutual Information between the permutation image’s pixels and their respective left-neighbors:

Here we see diversity in the histogram

Chart, histogram

Description automatically generated

From the middle as expected.   
  
we got Mut\_I1\_perm = 8.8746e+03

It is around 9000 bits of information so we can see a decrease in the mutual information from the regular left neighbours.

**Code**

clear all;

clc;

close all;

%% 1. Read an image -any image.

my\_cat = imread('my\_cat.jpeg');

whos

imshow(my\_cat);

%% 2. Present one of its RGB channels ? I1

%showing Red level

I1 = my\_cat(:,:,1);

figure(1);

imshow(I1);

%% 3. Permute I1 and present it.

rand\_order = randperm(numel(I1));

permI1 = I1(rand\_order);

whos

figure(2);

permI1\_im = reshape(permI1,size(I1));

imshow(permI1\_im)

%% 4. Present the histogram of I1.

I1\_doub = double(I1);

figure(3);

hist(I1\_doub(I1\_doub~= 0), 256);

%to get a PDF

P\_I1 = prob\_norm(I1);

figure(4);

permI1\_doub = double(permI1\_im);

hist(permI1\_doub(permI1\_doub~= 0), 256);

%% 5. Calculate its entropy.

J\_I1 = -sum(P\_I1(P\_I1~=0).\*log2(P\_I1(P\_I1~=0)))

%entropy(I1)

J\_permI1 = entropy(permI1\_im)

%% 6. Calculate the Mutual Information between I1 pixels and their respective left-neighbors.

I1\_last = I1(:,end);

I1\_mid = I1(:,[10:end]);

I1\_last\_mat = cast(ones(1,9),"uint8").\* I1\_last;

I1L = [I1\_mid, I1\_last\_mat];

imshow(I1L);

P\_I1\_L = prob\_norm(I1L);

figure;

hist = histogram2(double(I1L), double(I1), 256);

hist

P\_mat = hist.Values;

P\_mat = P\_mat./length(I1);

Mut\_I1 = mut\_info(P\_mat,P\_I1\_L,P\_I1)

%% 7. Calculate the Mutual Information between the permutation image?s pixels and their respective left-neighbors

I1perm\_last = permI1\_im(:,end);

I1perm\_mid = permI1\_im(:,[10:end]);

I1perm\_last\_mat = cast(ones(1,9),"uint8").\* I1perm\_last;

I1permL = [I1perm\_mid, I1perm\_last\_mat];

imshow(I1permL);

P\_Perm = prob\_norm(permI1\_im);

P\_PermL = prob\_norm(I1permL);

hist\_perm = histogram2(double(I1permL), double(permI1\_im), 256);

%P\_Mat\_perm = hist\_perm.Values;

P\_Mat\_perm = hist\_perm.Values./length(permI1\_im);

Mut\_I1\_perm = mut\_info(P\_Mat\_perm,P\_PermL,P\_Perm)

%% Functions section

function mut\_sum = mut\_info(Mat\_p,P1,P2)

%UNTITLED2 Summary of this function goes here

% Detailed explanation goes here

n = length(P1);

mut\_sum = 0;

for k = 1:n

for j = 1:n

if not(Mat\_p(k,j) == 0 ||P1(k) == 0||P2(j) == 0)

mut\_sum = mut\_sum + Mat\_p(k,j)\*log2(Mat\_p(k,j)/(P1(k)\*P2(j)));

end

end

end

end

function PDF = prob\_norm(I)

I = double(I(:));

h = hist(I(I~= 0), 256);

len = length(I);

PDF = h./len;

end