# Hands-On 02 - DFT and DCT using OpenCV

*Readme*

Vineet Gattani

## Folder Structure:

This is the readme/report file.
The main folder is named DFT_DCT_Gattani.
The code folder also contains 1 cpp file (main.cpp), 1 header file, 1 .jpg file which is the input image and one folder named 'Results'.
The 'Results' folder is sub-divided into two folders :

1. DFT
    1.1. This folder contains screenshots of the DFT being masked based on variable parameter and it's corresponding reconstructed images.
2. DCT
    2.1. This folder contains screenshots of the DCT being masked based on variable parameter and it's corresponding reconstructed images.

## Procedure:

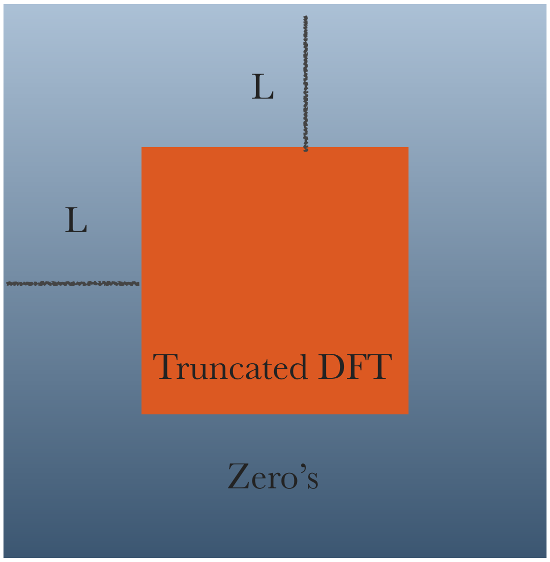
### *1. Setting up the project:*

Create a new project and set it up for OpenCV use. Add all the necessary .dylib files to the project. Make sure the input image is in the same location as the project folder.

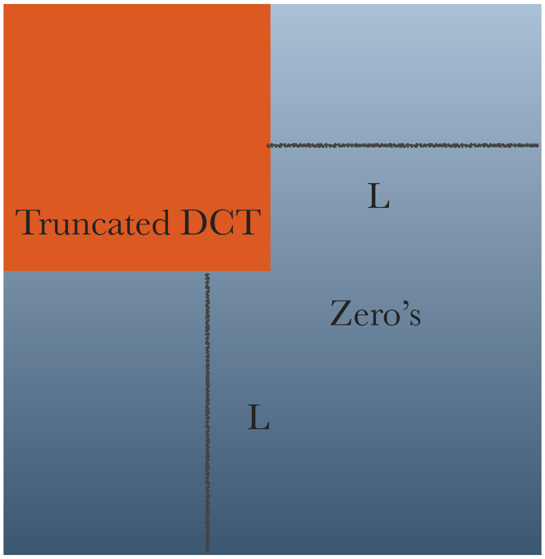
### *2. Compiling and running the code*

Now, copy the code from the main.cpp file provided and paste it in the empty main.cpp file in the project. Make sure the image file zebra.jpg is also placed in the same folder as the main code.
Create a header file in the same location as main.cpp and copy the code from variables.h into the newly created header file.
Compile the project by going to `Product → Build.`
Now the build is succeeded, run the project by going to `Product → Run.`
As the code is running, it prompts the user to enter $L_{min}$ for DFT and DCT. This decides the truncation of the DFT and the DCT.
Enter the values required and press enter.
Before we show results, let us establish a reference from where this $L_{min}$ is measured in both DFT and DCT.

Assume that the DFT of the image is going to look like the image below. The blue region is the part which will be truncated or masked. The center of the square will have the high frequency coefficients. Thus we find a threshold value of L for which the reconstruction is satisfactory. Increasing the L beyond this point would only decrease the quality of the reconstructed image.

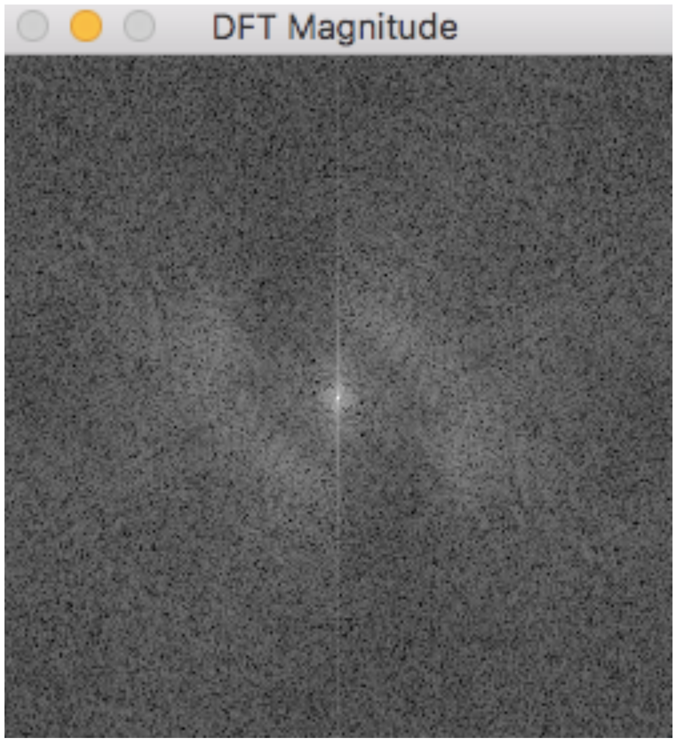



Similarly, visualize a DCT like below. We are going to find a minimum value of L for the masked DCT below, for which the reconstruction is satisfactory. Increasing the value of L beyond this value will make help the reconstruction of the image but the changes will not be significant.

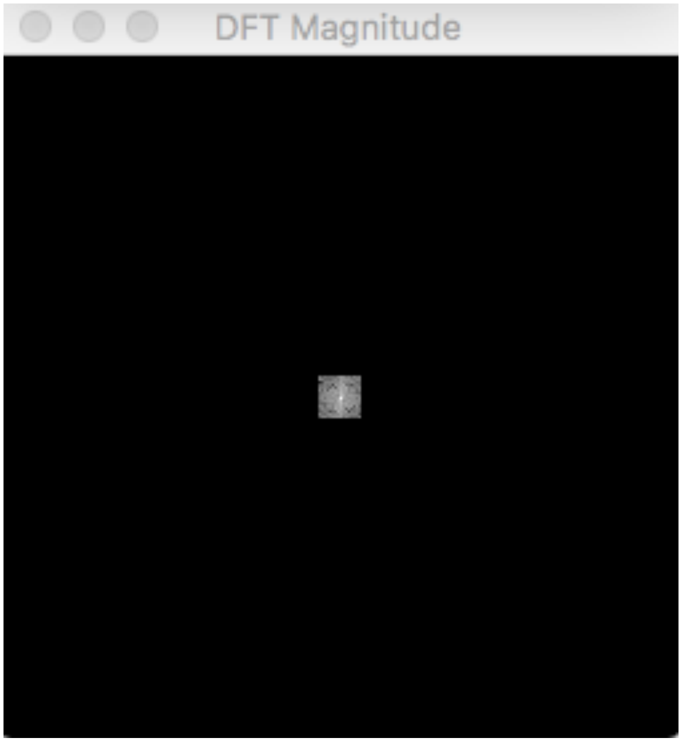


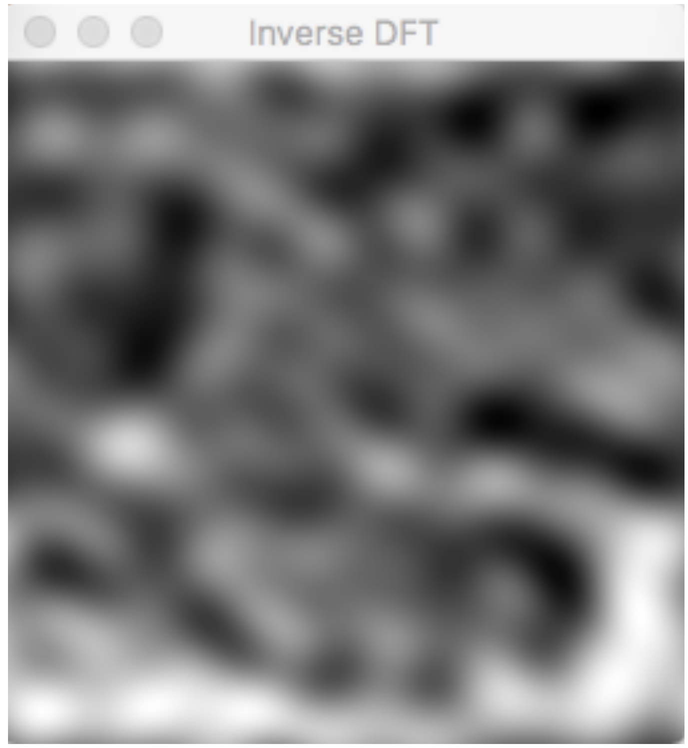
DFT masking :

Below is unmasked magnitude of DFT.

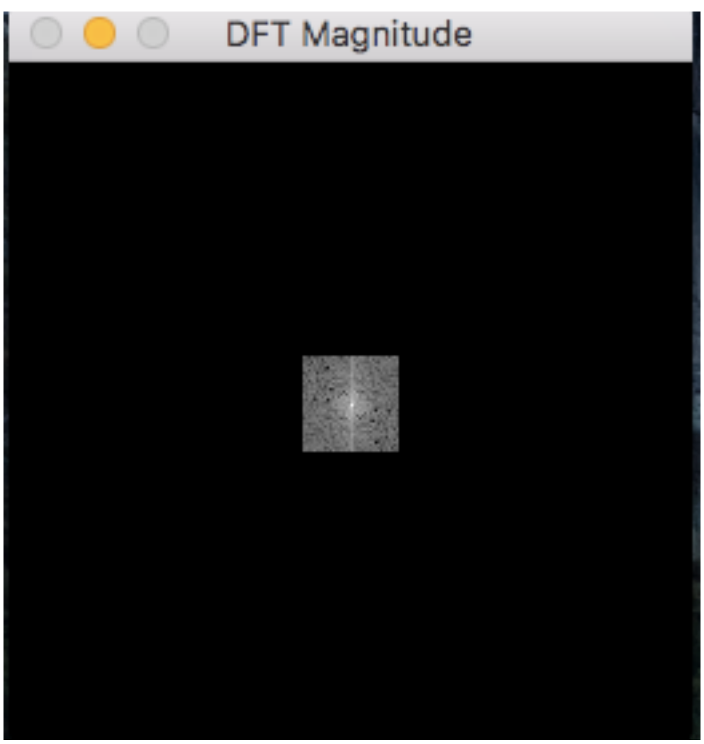


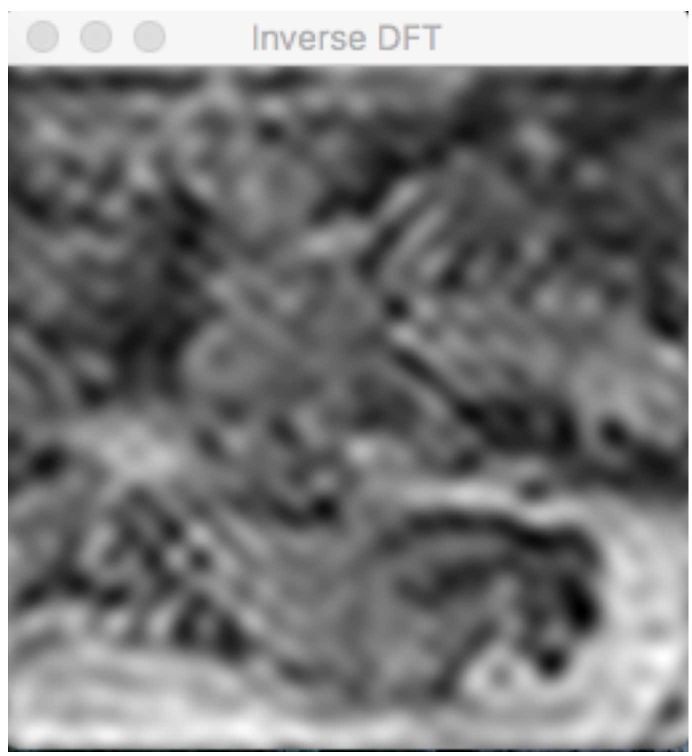
L = 120; MSE = 0.0507



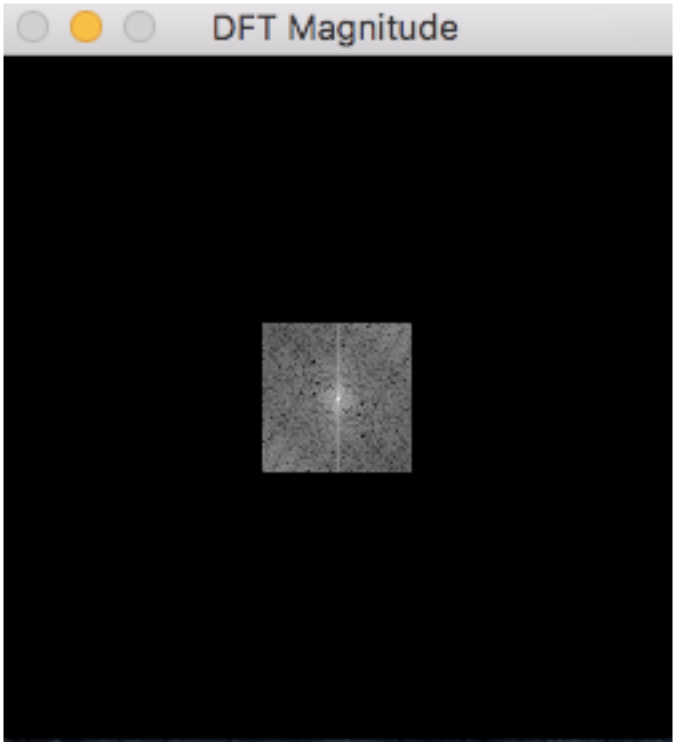



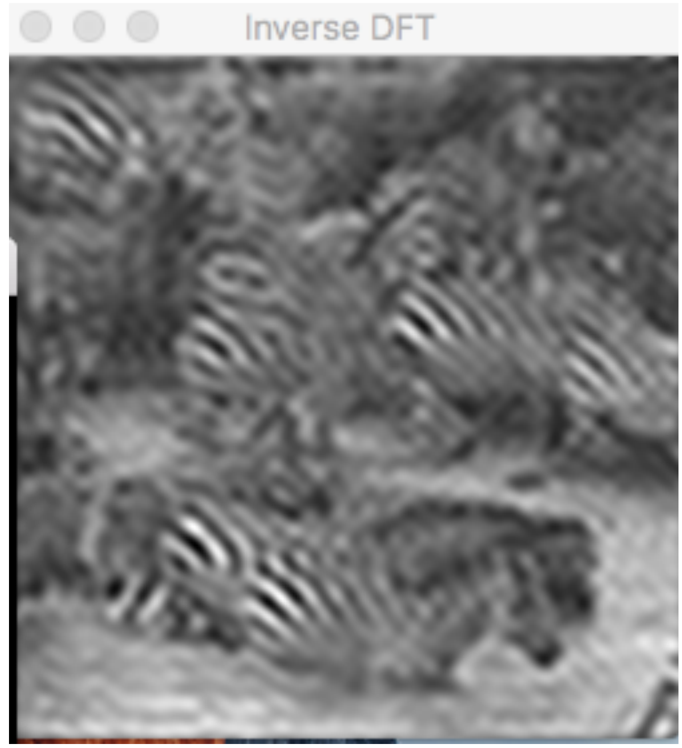

L = 110; MSE = 0.0431



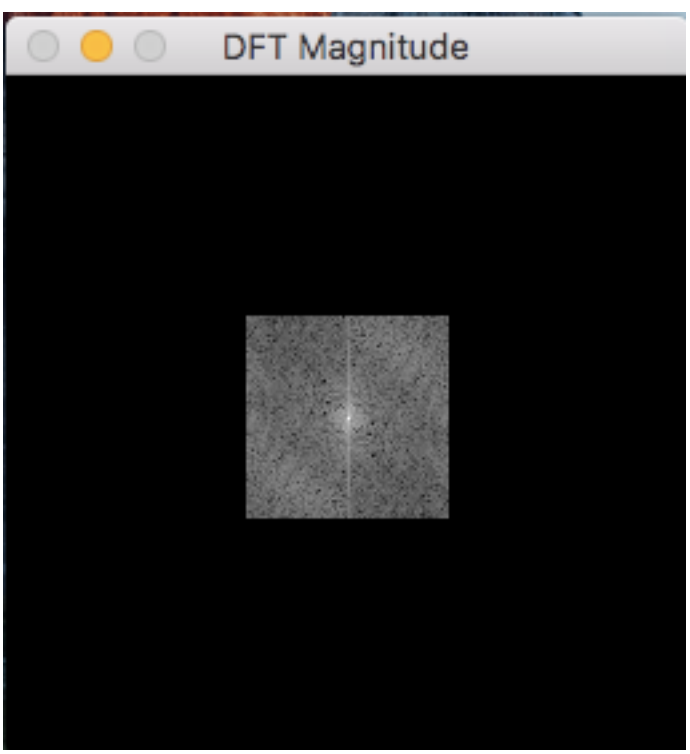



L = 100; MSE 0.04011

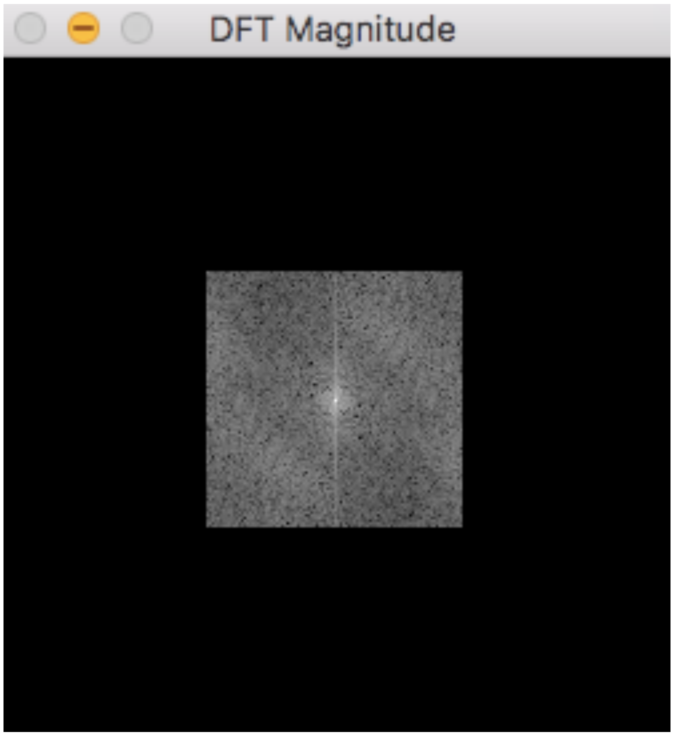



L = 90; MSE= 0.0344

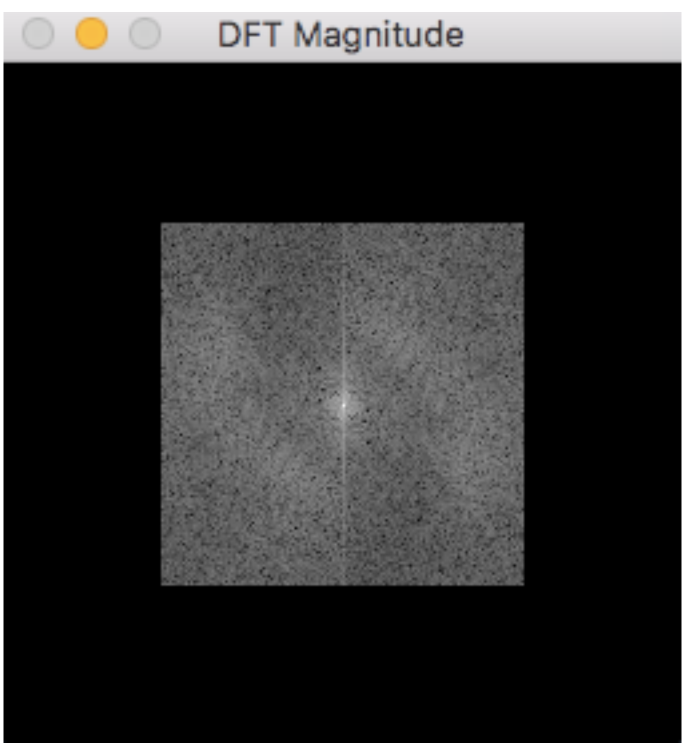


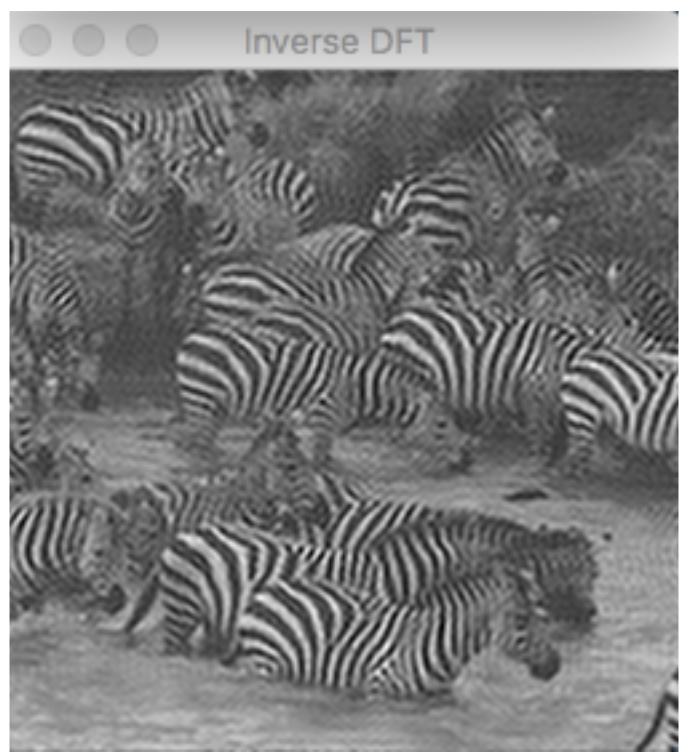

L = 80; MSE = 0.0292



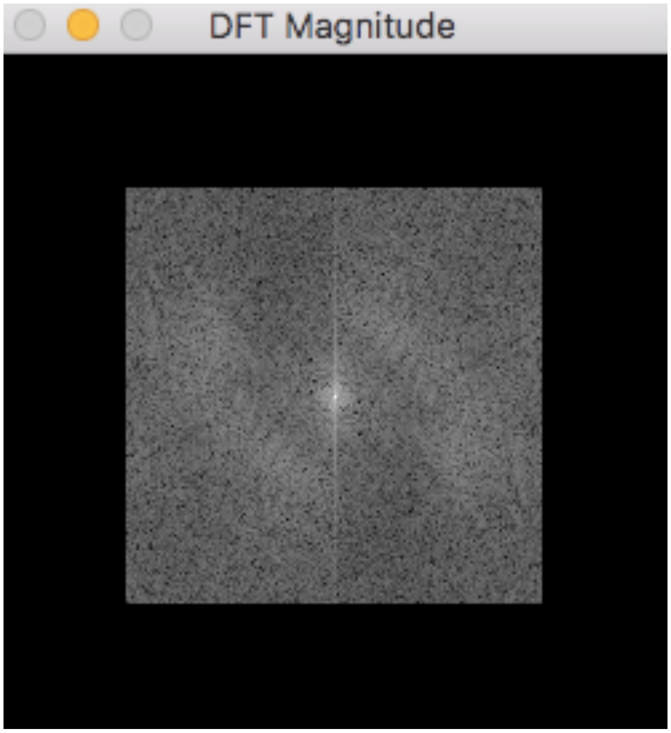


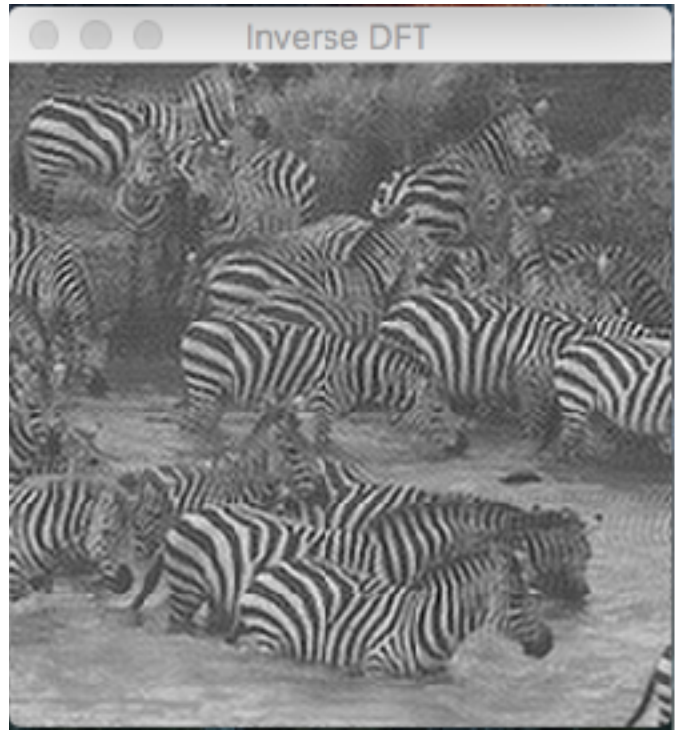
L=60;MSE=0.02326



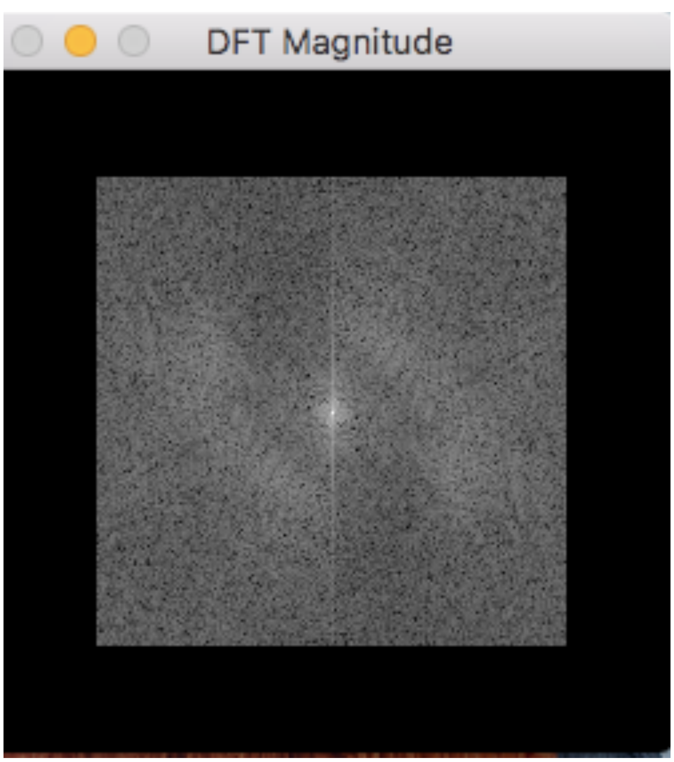



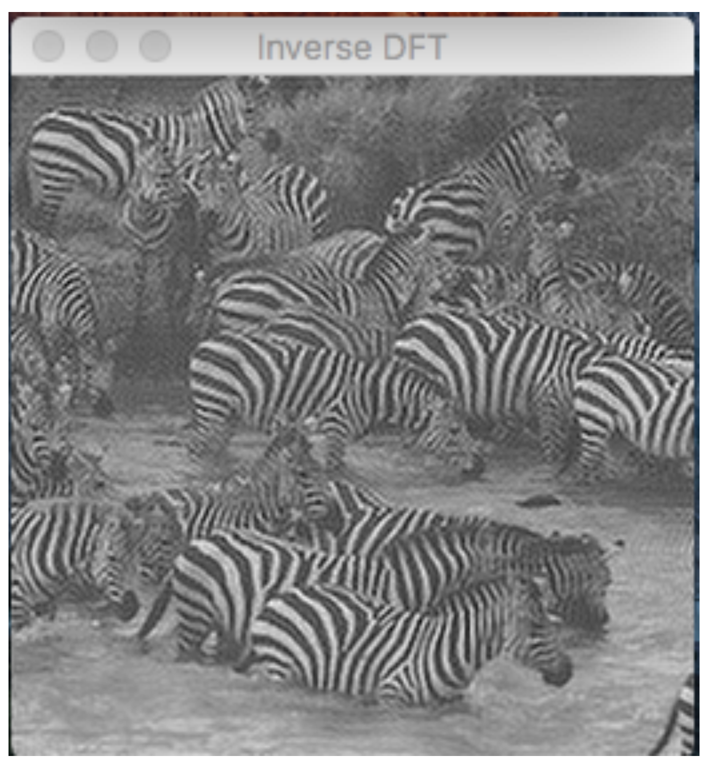
L= 50;MSE =0.0248




L = 40; MSE =0.02313

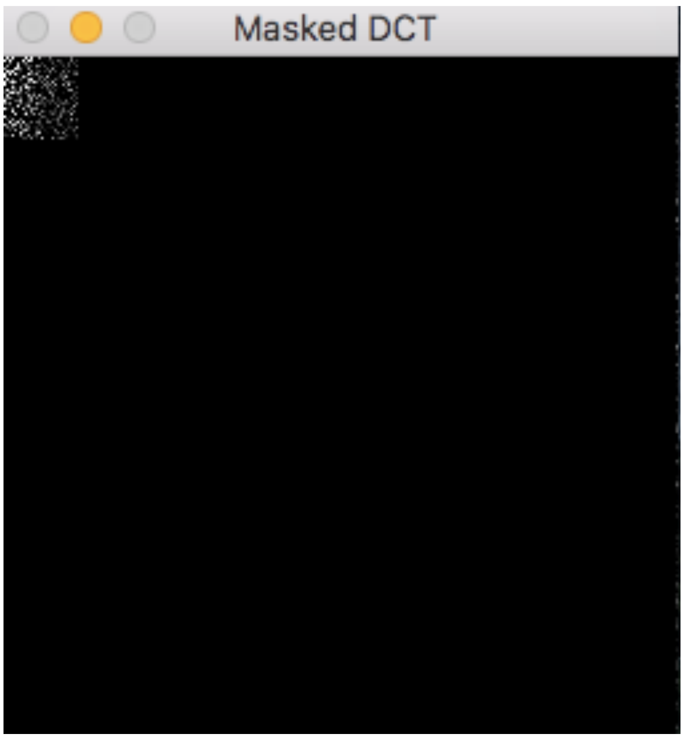


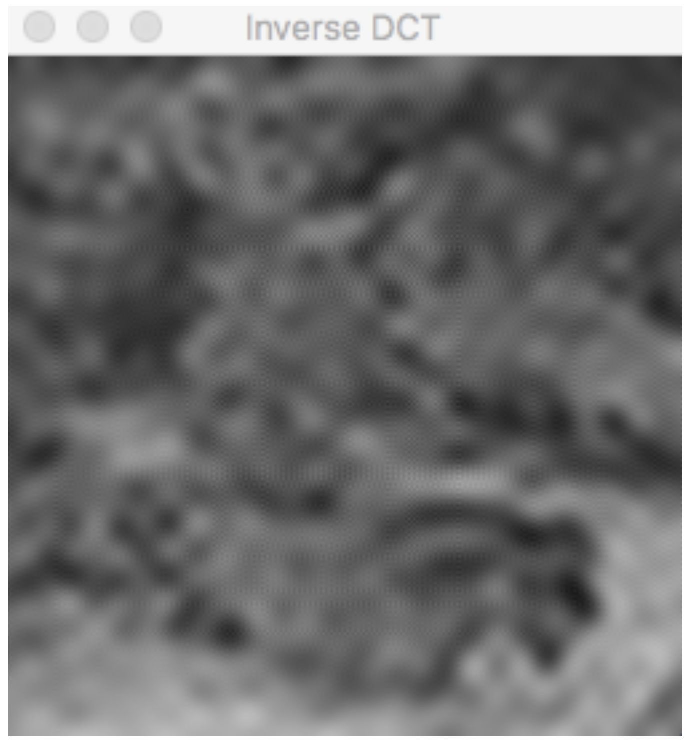



Thus, from the above visual inspection of the reconstructed image and MSE values, we can say that L =60 is maximum/threshold value required to reconstruct the image successfully without much loss. Any value below 60 (50,40…) would also lead to similar results as above.

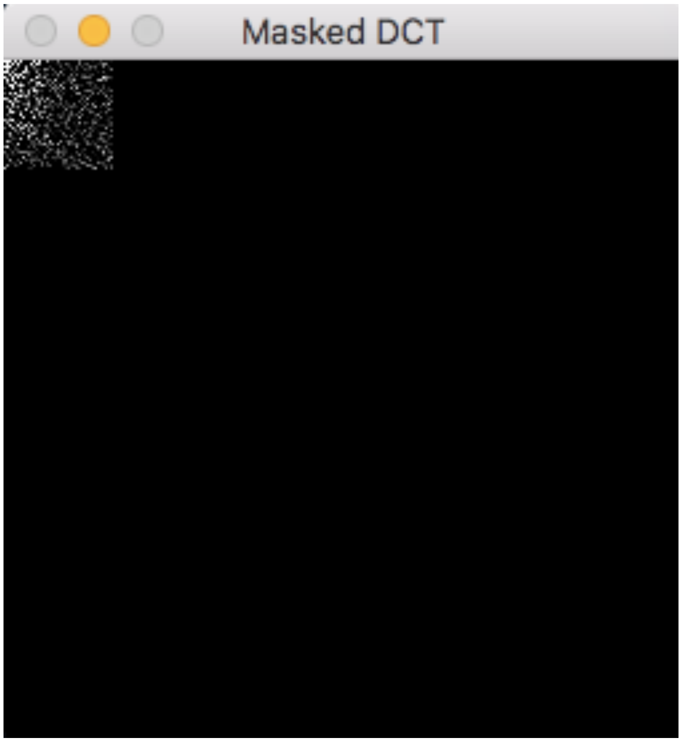
DCT masking:

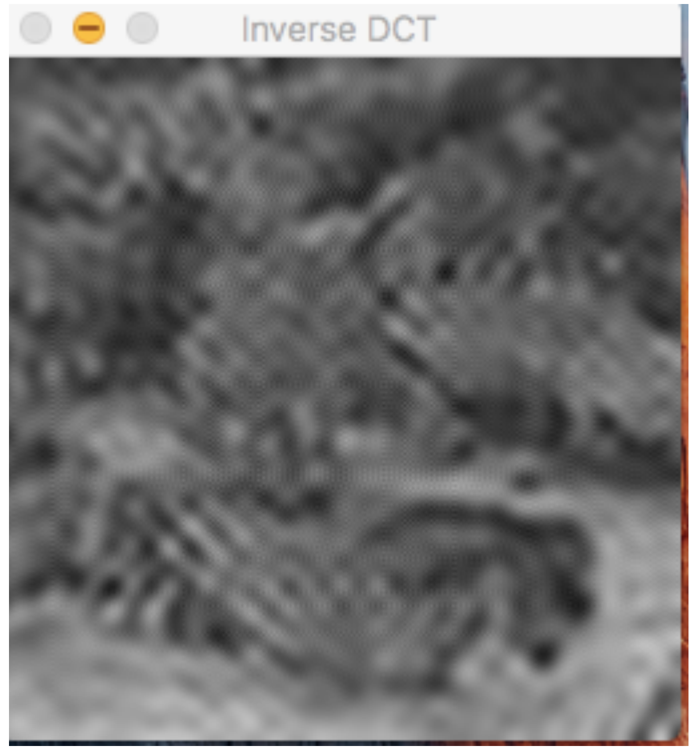
L=30; MSE = 0.038



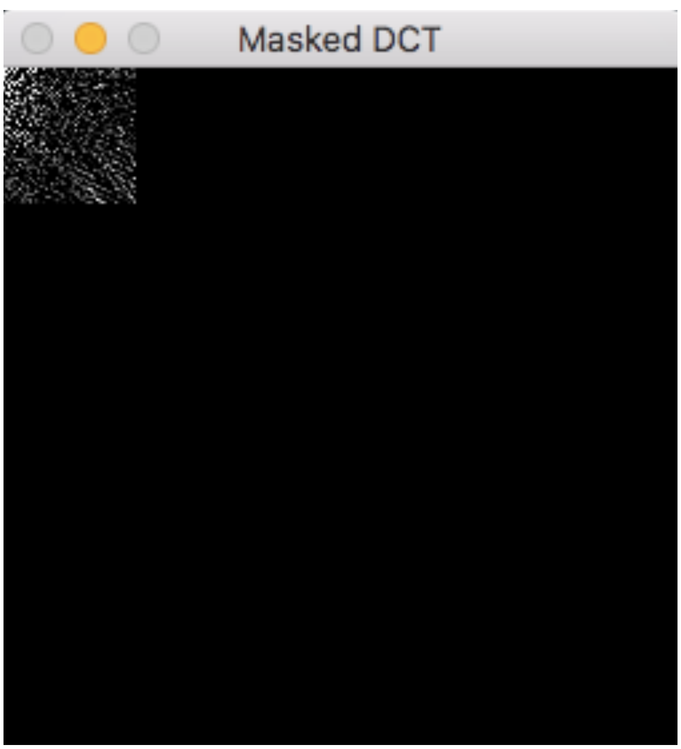



L=40; MSE = 0.037

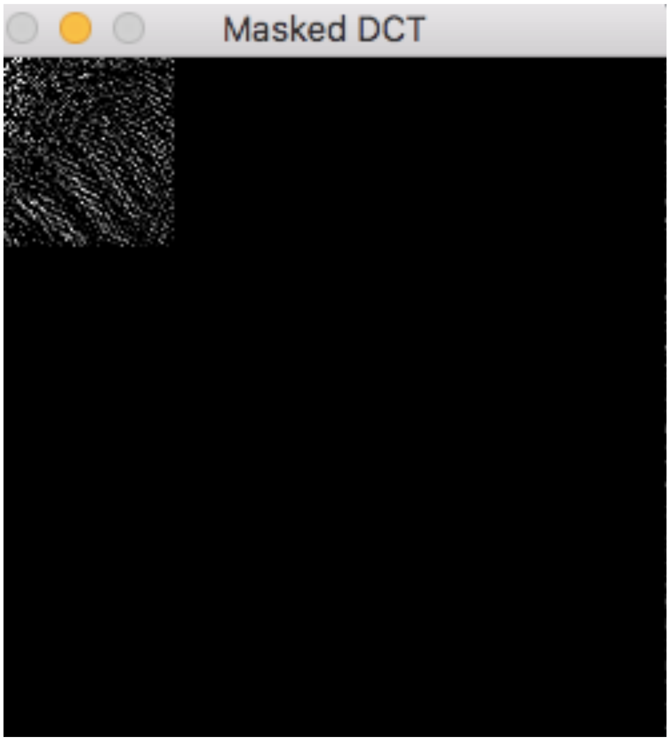



L=50; MSE = 0.0356

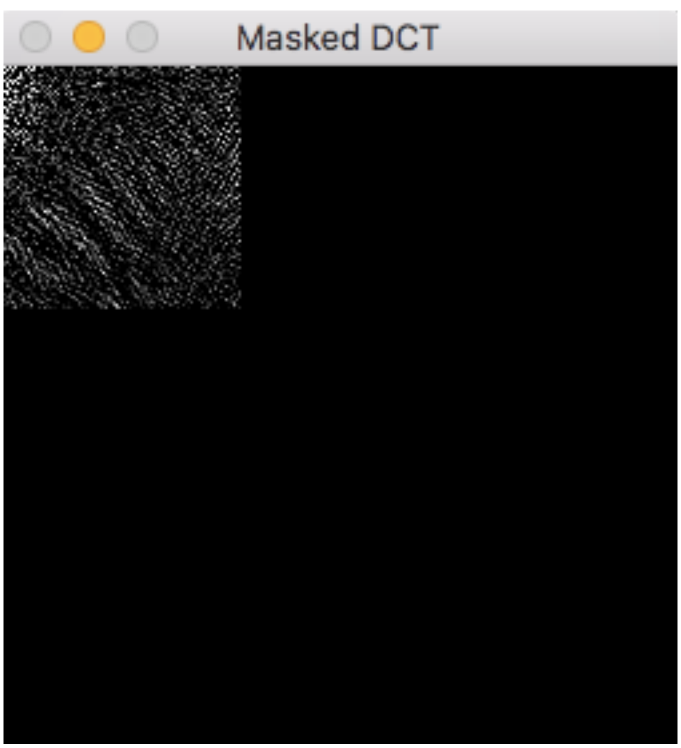



L = 70; MSE = 0.0318

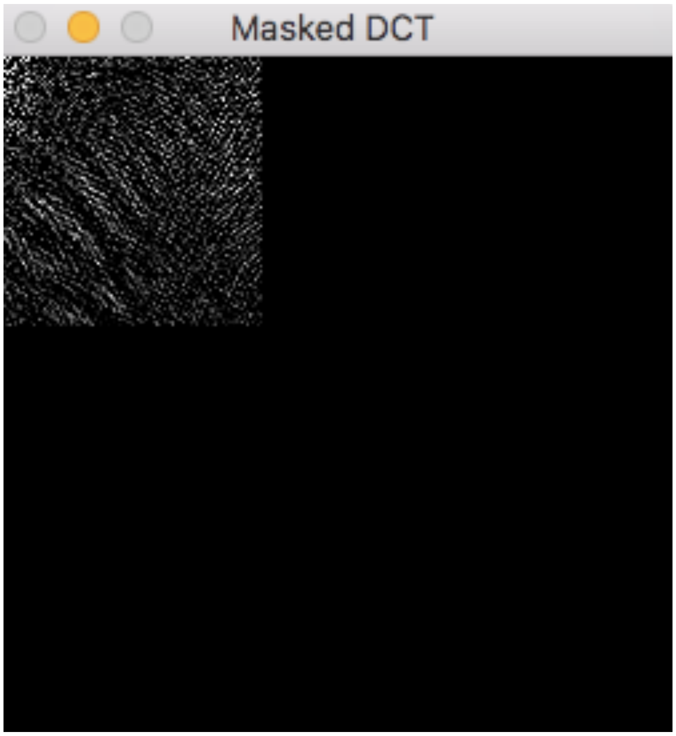




L=90; MSE=0.0276

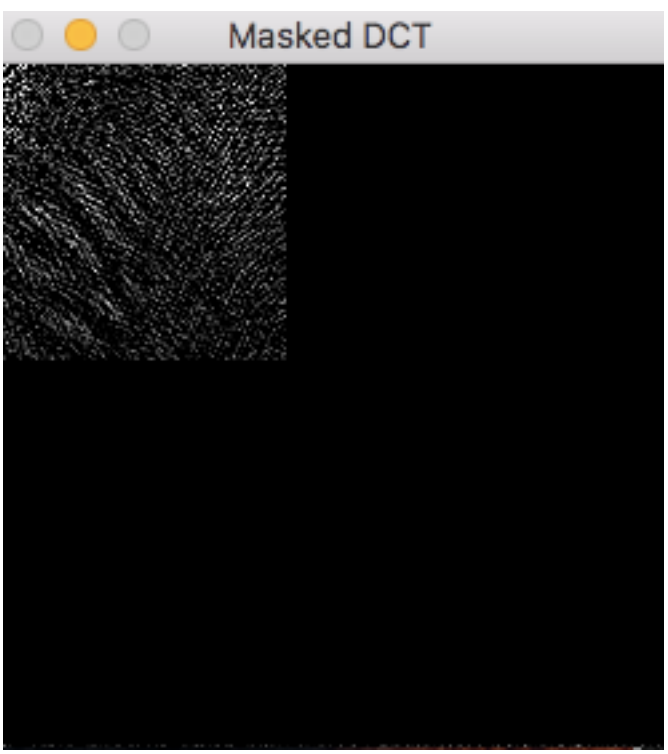



L=100; MSE =0.0252

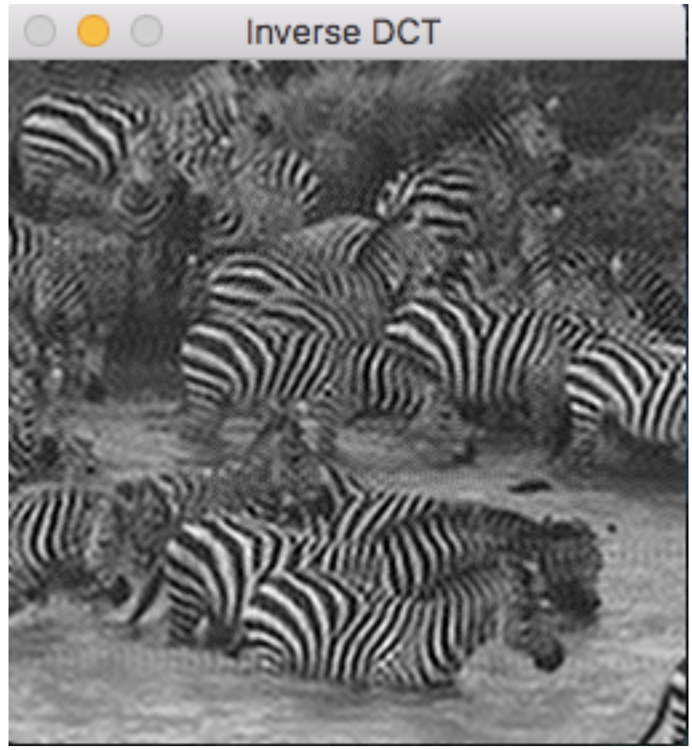




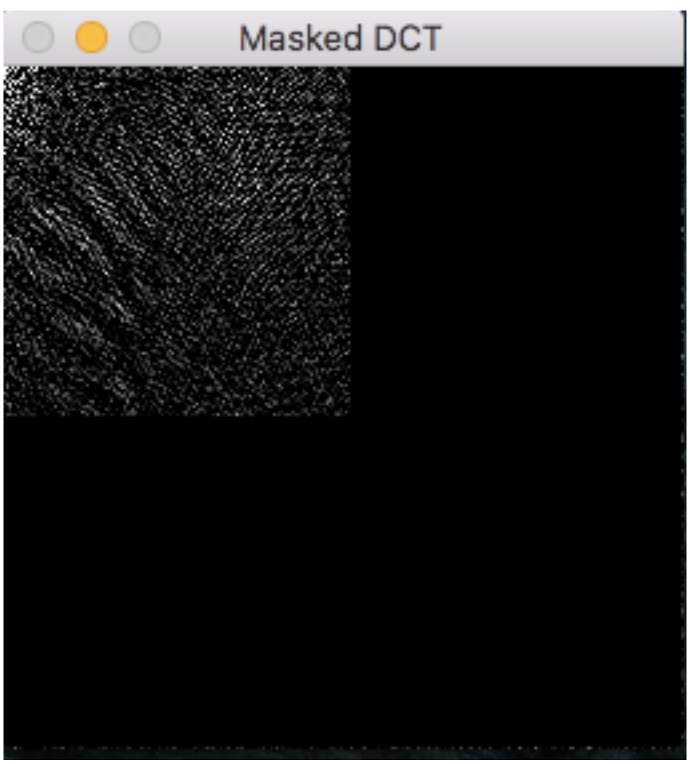
L=110; MSE = 0.0232

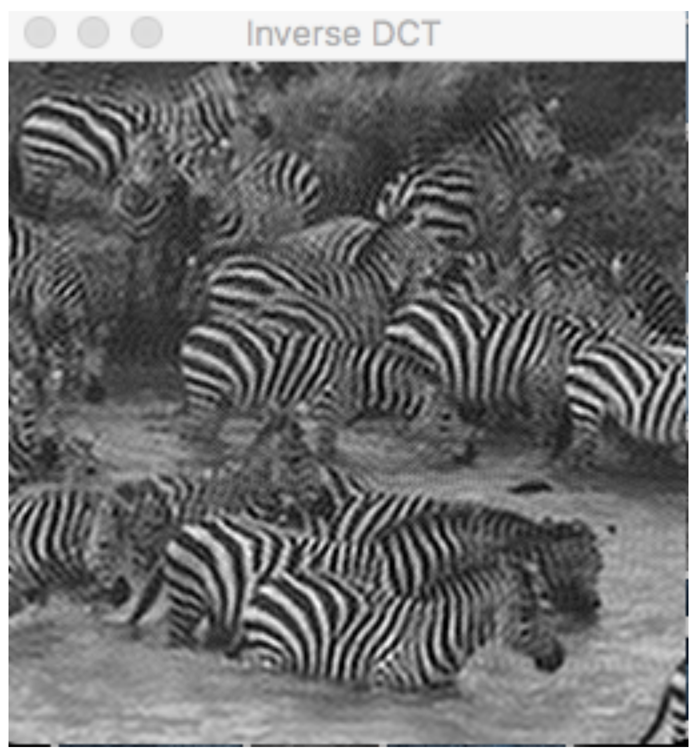



L=120;MSE=0.0214


L=130; MSE 0.0197



Thus, from the above visual inspection of the reconstructed image and MSE values, we can say that L =120 is minimum/threshold value required to reconstruct the image successfully without much loss. Any value above 120 (130,140…) would also lead to similar results as above.

Note: The above instructions for compilation are for Xcode users. However, the build and run procedures vary for different OS's.