# Report DIP Project
# Fast Super-Resolution using Weighted Median filtering

201431218 - Abhinav.Appidi

201431212 - Rekha Devi.

## Problem statement :

Fast Super-Resolution using Weighted Median Filtering.

## Motivation :

A 'non-iterative' method of image super-resolution based on weighted median filtering with Gaussian weights. Why weighted median filtering ? It reduces the errors caused by inaccurate motion vectors and since it is a non-iterative method,the implementation is fast compared to other iterative methods.

## Super-resolution :

Super-resolution is a technique that enhances the resolution of an image.

## Introduction :

Even after increase in resolution of modern camera sensors, image resampling is still a problem.Many image resampling algorithms use priori information which improves image quality only if priori information we have is true.

Super resolution is an alternative method to obtain better results.In this, several images of an object with subpixel shifts(low-resolution images) are used to construct a single high-resolution image. If the object motion and approximation functions are known, then the information from all images can be used to construct a single high-resolution image.

Main problem of Super-resolution is requirement of accurate motion estimation. But, in our approach, we implement a method stable enough to the errors of motion vectors estimation.

The goal of Super Resolution methods is to recover a high resolution image from one or more low resolution input images. In the classical multi-image SR, a set of low-resolution images of the same scene are taken at subpixel misalignments. Each low resolution image imposes a set of linear constraints on the unknown high resolution intensity values.If enough low-resolution images are available (at subpixel shifts), then the set of equations becomes determined and can be solved to recover the

high-resolution image.Practically, however, this approach is numerically limited only to small increases in resolution
.

***Even small errors in motion estimation results in serious degradation of the reconstructed image.***

If the implementing algorithm doesn't take into account the inaccuracy of motion vectors, it give ineffective results.In order to account for the inaccuracy of motion estimators, we use different algorithm in our implemented method.

The super-resolution algorithm with weighted median filtering, can be posed as an inverse problem.The low-resolution images $u_k$ are produced from high-resolution image(z) by applying motion transformation and Downsampling.

$$A_k z = u_k \text{ , } k = 1,2,3,4,.....,N$$

here the operator $A_k$ in general is represented as

$$A_k = D.(H_{cam}).(F_k).(H_{atm}).z + n.$$

$H_{atm}$ is atmosphere blur, $F_k$ is motion operator, $H_{cam}$ is the camera lens blur, D is downscaling operator, n is a noise.

The atmosphere and camera lens blurs are modeled by a single Gauss filter H, and the system of equations takes the form

$$A_k.z = D.F_k.H.z = u_k, \text{ } k=1,2,3,...,N.$$

In other methods we can use average sum of upsampled and motion compensated low resolution images.

## What to do (brief):

We consider the super resolution with z and $u_k$ defined on discrete sets {(i,j): i,j belongs to Z}.The motion transform $F_k$ defines a set of correspondences between the coordinates of points of source image and points of motion transformed image. Operator D performs downscaling. We process several upsampled low-resolution images into a single image which is our super-resolution image.

$$F_k z(i,j) = z(x^{\sim k}_{i,j} \text{ , } y^{\sim k}_{i,j} ).$$

$$D.z(x,y) = z(sx,sy).$$ (s -> is a downscaling factor)

Combining both $DF_k z(x,y) = z(x^{\sim k}_{si,sj} \text{ , } y^{\sim k}_{si,sj}) \rightarrow z(x^k_{i,j} \text{ , } y^k_{i,j})$ .

$$(Hz)(x^k_{i,j} \text{ , } y^k_{i,j}) = u^k_{i,j} .$$

We combine all low-resolution images $u_i$ into a single image and rewrite $(x^k_{i,j} \text{ , } y^k_{i,j} \text{ , } u^k_{i,j})$ as $(x_n \text{ , } y_n \text{ , } w_n)$. Here we have,

$$\mathbf{(Hz)(x_n , y_n) = w_n .}$$

$W_n$ corresponds to the pixel value of each pixel $(x_n , y_n)$ of the blurred high-resolution image (Hz). now we apply deblurring to the (Hz) image for final output of super-resolution image.

## Problem Solution :

In our method instead of constructing the high resolution image (Hz) which satisfies our required conditions in every point (x,y) we use the following algorithm :

For every pixel (i,j) of the high resolution (Hz) we take all points $(x_n,y_n)$ from a small neighbourhood of (i,j) and perform averaging of these values. we can use averaging method based on gaussian filtering but it results in image blur. (here **$\sigma$** is Gaussian radius. )

$$\mathbf{(Hz)(i,j) = [ \textstyle\sum_n w_n . \exp( -1*( (x_n-i)^2 + (y_n -j)^2 ) / (2\sigma^2) ) ] / [ \sum_n \exp( -1*( (x_n-i)^2 + (y_n -j)^2 ) / (2\sigma^2) ) ]}$$

A robust approach based on median filtering upsamples the low-resolution images and combining the upsampled images using median averaging.It applies the median averaging to the values of all points in the neighbourhood of the target pixel.Median averaging produces sharp edges but it doesnot use the spatial distribution of points $(x_n,y_n)$.

$$(Hz)(i,j) = \text{med}(w_n : ((x_n-i)^2 + (y_n-j)^2 ) < R^2 ).$$

So we take the benefits of both Gaussian averaging and median averaging and use a combined method based on weighted median averaging.In weighted median every pixel value $w_n$ has a weight $c_n$. we choose weight $c_n$ from Gauss weights.

$$c_n = \exp( -1*( (x_n-i)^2 + (y_n -j)^2 ) / (2\sigma^2) ) .$$

We calculate the weighted median as weighted average with $w_n$ taken $c_n$ times when $c_n$ are natural numbers.In general case, take pairs $(w_n,c_n)$ and sort them in ascending order of $w_n$, we find value of m which satisfies the conditions

$$\textstyle\sum^{m-1}_{k=1} c_k \le S/2 , \sum^{m}_{k=1} c_k > S/2 , S = \sum_k c_k .$$

and take the corresponding value Wm as the result of weighted median averaging for that pixel.

The obtained result is the approximation of the blurred image (Hz).The results can be improved by varying the gauss radius(sigma) and and the radius(R) for different pixels of the image.

## How we proceeded :

Method 1 :

First we take 'n' low resolution images with motion transformed, random shifts , noisy and blurry images. These low resolution images are in downsampled format. In our first implementation , we proceeded with taking gaussian average of all low resolution images, and combining them to a single super resolution image.

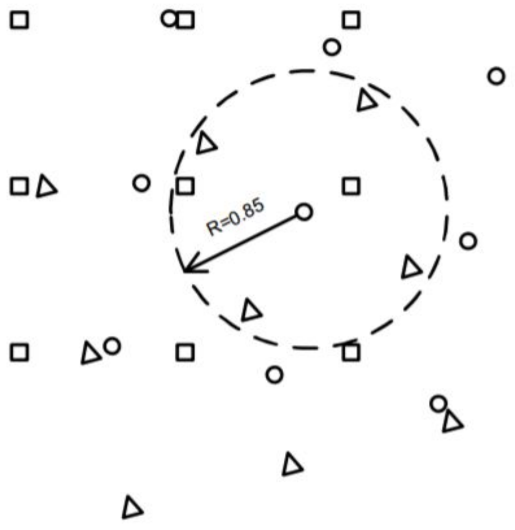
$$(Hz)(i,j)= [ \sum_n w_n . \exp( -1*( (x_n-i)^2 + (y_n -j)^2 ) / (2\sigma^2) ) ] / [ \sum_n \exp( -1*( (x_n-i)^2 + (y_n -j)^2 ) / (2\sigma^2) ) ]$$

Where radius of gaussian (σ) is chosen from experimenting the output image. But the Gaussian averaged is like a low-pass filter and obtained super resolution image is blurred and is ineffective at the edges.

Method 2 :

So, we thought of using alternative method to preserve edges. For this to produce sharp edges we used median averaging of all low-resolution images. In this we up-sampled the low-resolution images and combined them using median averaging to the values of pixels in the neighbourhood of the target pixel.

$$(Hz)(i,j) = med(w_n : ((x_n-i)^2 + (y_n-j)^2 ) < R^2 ).$$



( Median filter for irregular samples. Samples for three different images are represented. The point in the centre of the circle is substituted by the median of all points in the circle, including itself. )

We take all pixels that fall inside the circle for all low-resolution images and median is calculated for all those pixel values and that obtained value is substituted as the pixel value at the centre of the circle.

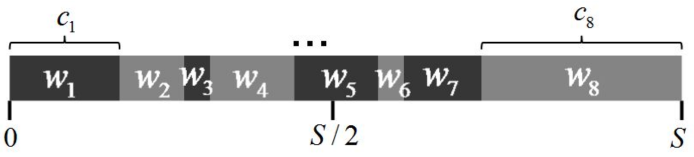
<u>Method 3 :</u>

Median averaging produces sharp edges but it does not use the spatial distribution of points $(x_n , y_n)$. So, to take the benefits of both Gaussian averaging and median averaging we implemented the final method based on weighted median averaging. In weighted median $w_{med}(w_n , c_n)$ , every pixel value $w_n$ has a weight $c_n$ associated with it. We choose the weights $c_n$ as in Gaussian averaging.

$$\mathbf{c_n = exp( -1*( (x_n-i)^2 + (y_n -j)^2 ) / (2\sigma^2) ) .}$$

To find the result of the median averaging $w_{med}(w_n , c_n)$, the pairs $(w_n , c_n)$ are sorted in the ascending order of $w_n$ . Next , we find the value of '**m**' which satisfies the below conditions : And take the value $w_m$ as the result of the weighted median averaging.

$$\sum_{k=1}^{m-1} c_k \le S/2 \, , \sum_{k=1}^{m} c_k > S/2 \, , S = \sum_k c_k \, .$$



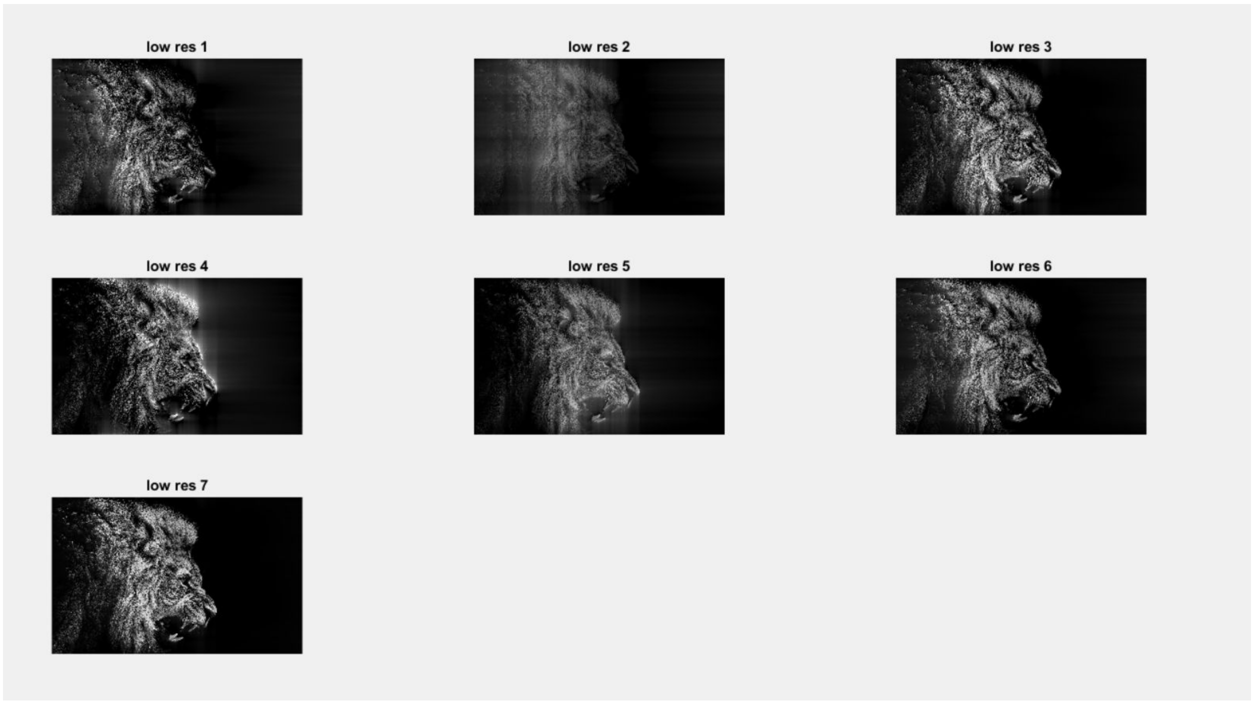
Weighted median averaging procedure.

The obtained result is the approximation of the blurred image (Hz) . This method produces sharp images. This is because of the behaviour of median filtering. Then deblur algorithm is applied to get the final super-resolution image.

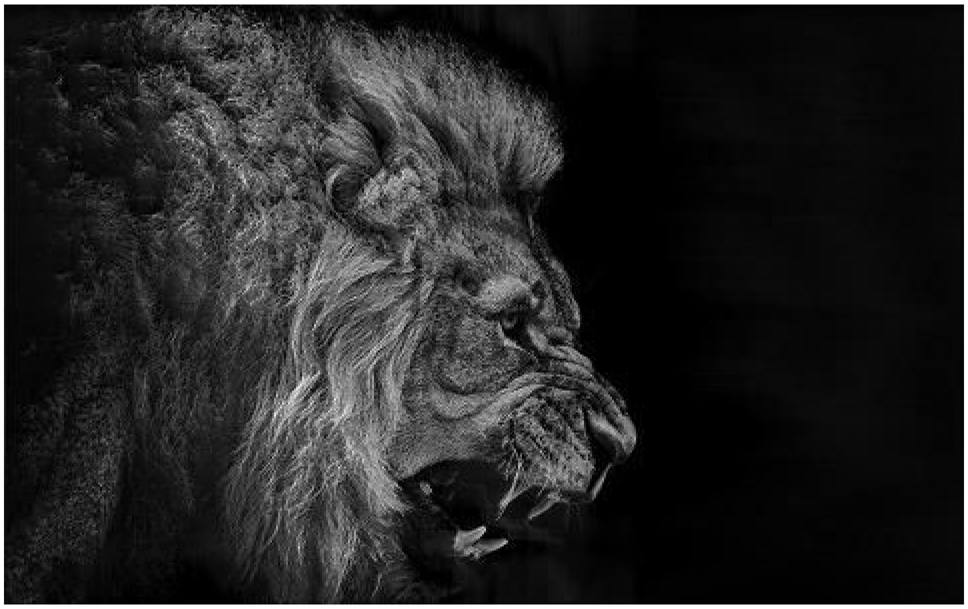
( If we increase the radius of median filtering(R) we do not increase the image blur of the resulting image and hence deblur algorithms need not be applied.). So , in our applied method of weighted median filtering , it reduces the errors caused by the inaccurate motion vectors.

## Results :

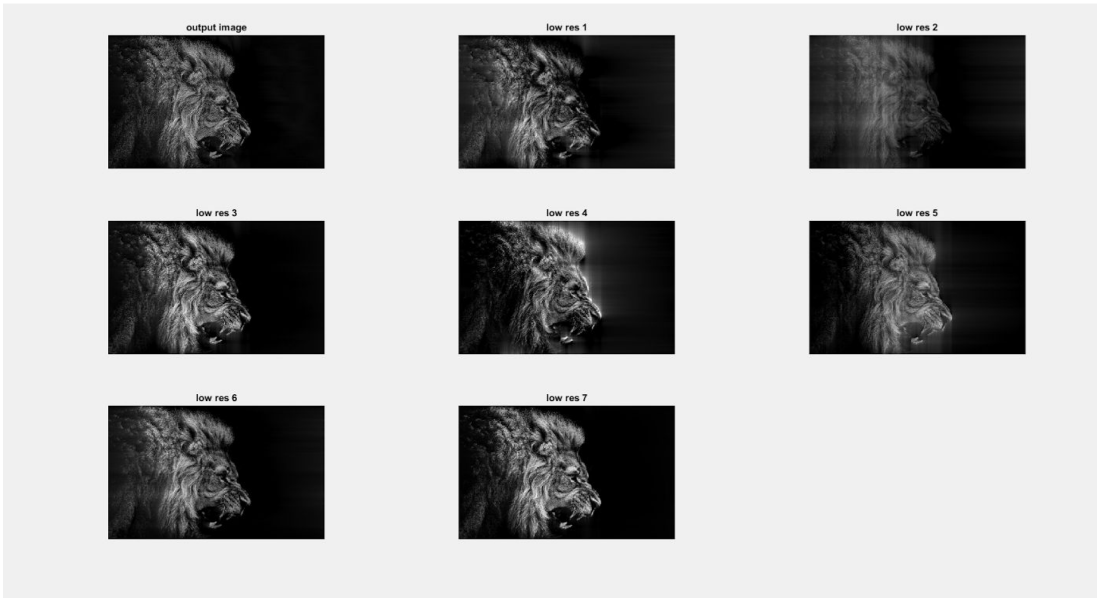
We took about 7 low resolution images as dataset



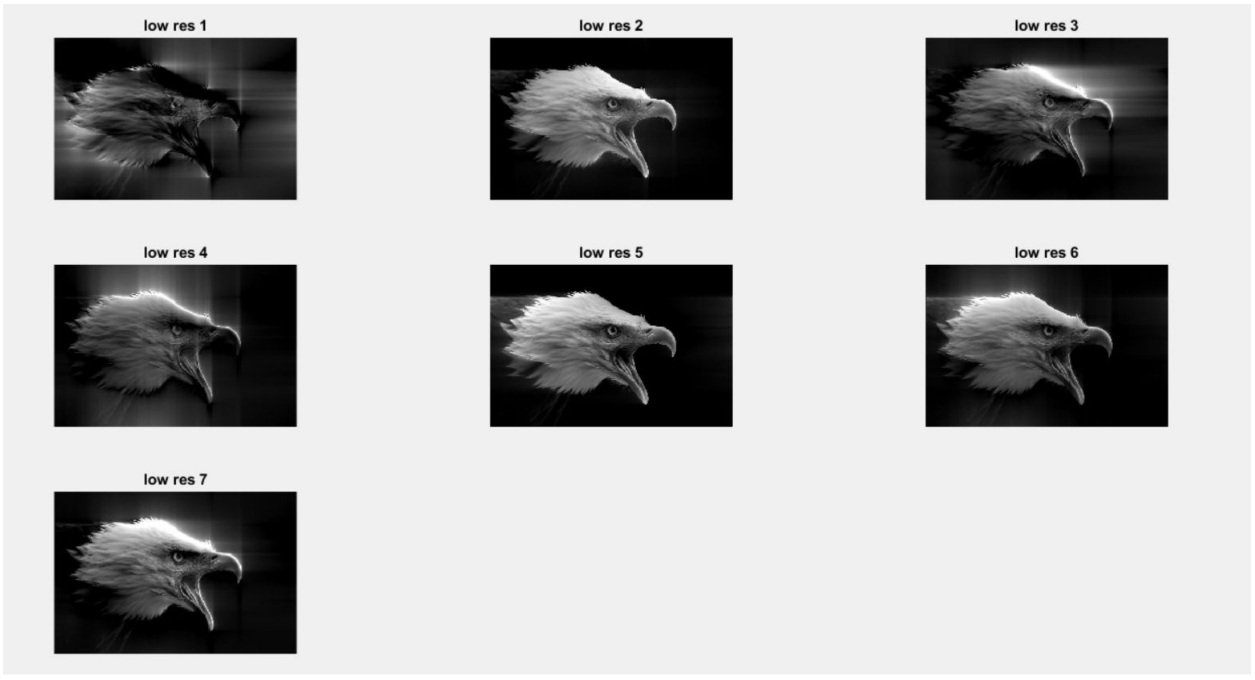
Sample for one dataset is shown above. And the output super-resolution image is



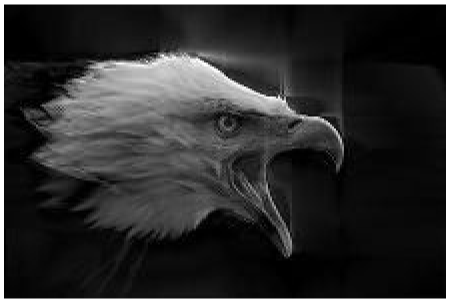
Comparing output image with all input low-resolution images :



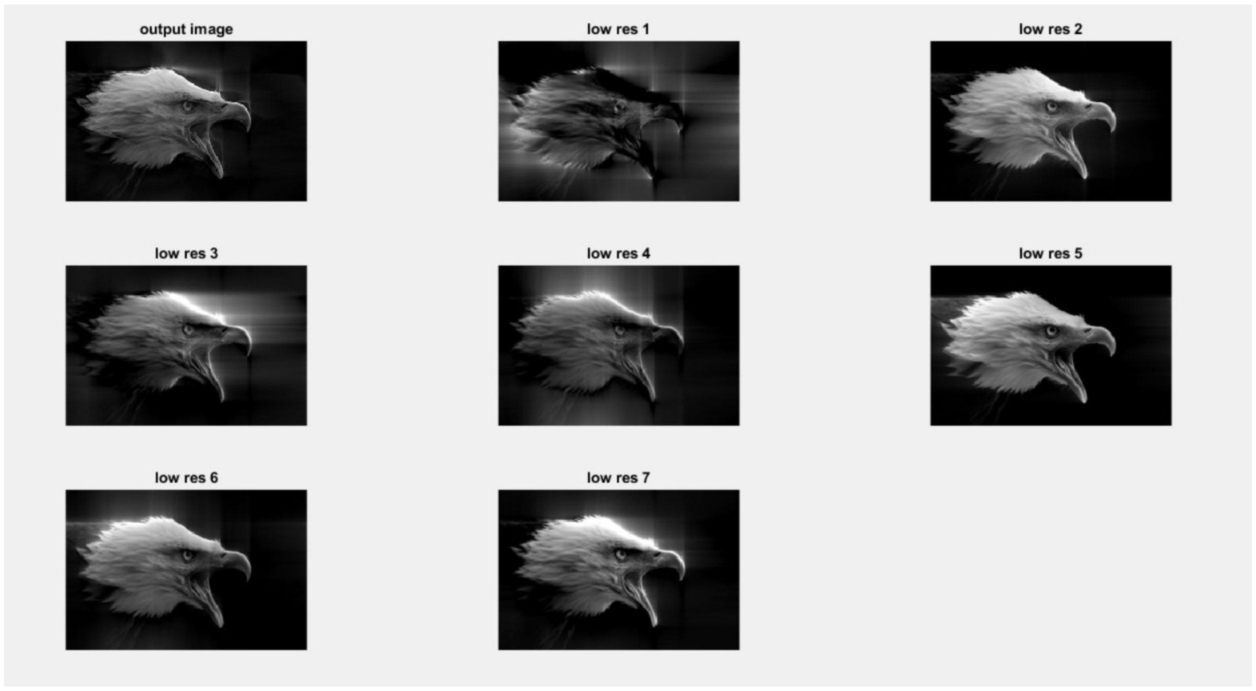
For dataset two we took 7 input low-resolution images. The low-resolution input images are shown below



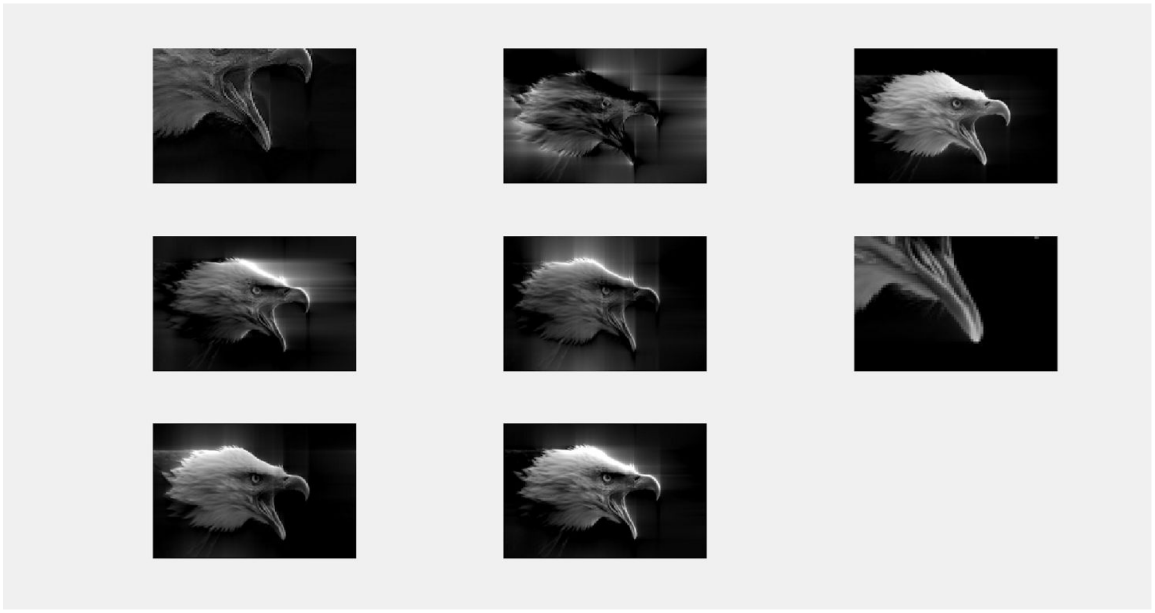
Output super resolution image for the above dataset is



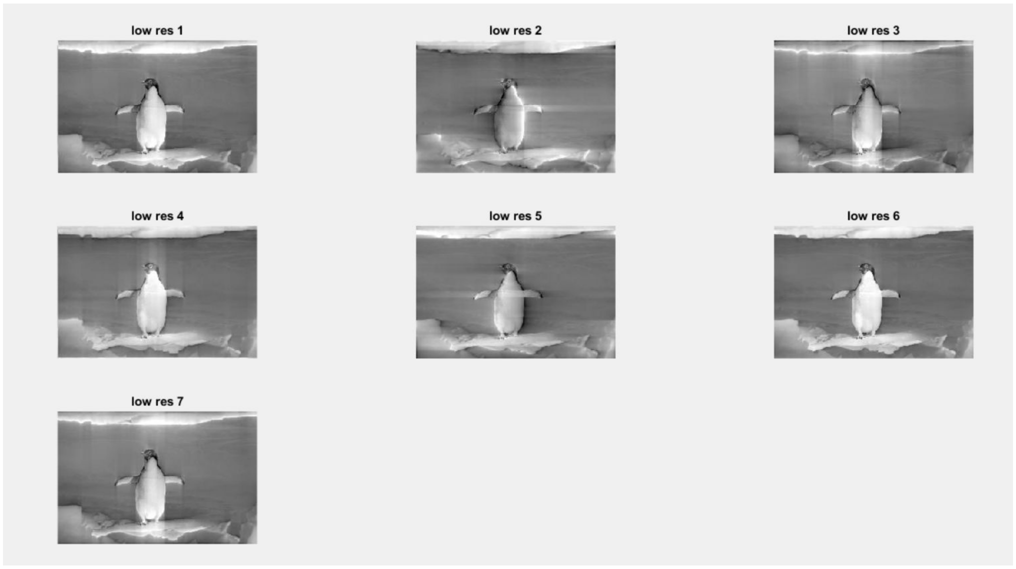
And comparing output image with all low-resolution images is shown below.



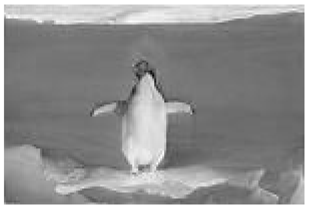
If we are unable to find difference between output any near equal low-resolution image check below image ;(we can see blur in low-res image when zoomed in and edges preserved in super resolution output image)



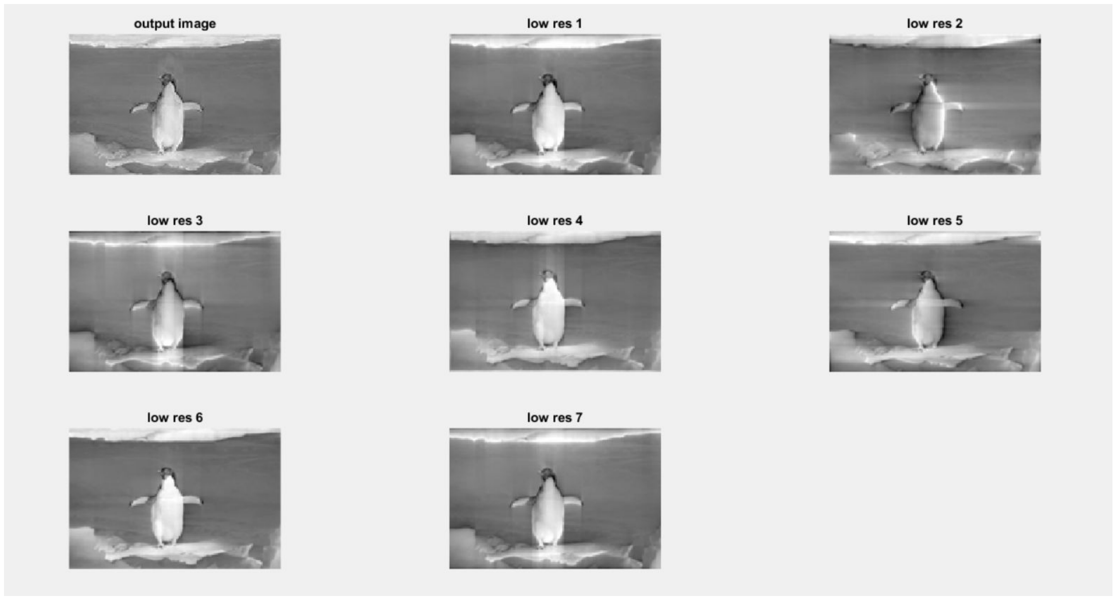
For dataset 3 we took 7 low resolution images as dataset ,



Output for dataset 3 is :



Comparing output with all low-resolution input images :



The perfection in the output super-resolution image depends mutually on the value of Gaussian radius(σ) in deblurring and the value of R we choose.If R is small, the image would be blurred and hence we need to apply deblurring algorithm on the obtained image in order to get the high resolution image whereas when the R is high, the resulting image tends to be piecewise flat and hence no deblurring algorithm is required to get the final super-resolution image.
By changing the values of R and σ over and over in the process of implementing super resolution algorithm, using above data we arrived at a optimal values of R and σ for both the datasets.