Interpolation and internal painting are one of the basic approaches in image internal painting, which is used to eliminate undesirable parts that occur in digital images or to enhance faulty parts. This study was designed to compare the interpolation algorithms used in image in-painting in the literature. Errors and noise generated on the colour and grayscale formats of some of the commonly used standard images in the literature were corrected by using Cubic, Kriging, Radial based function and High dimensional model representation approaches and the results were compared using standard image comparison criteria, namely, PSNR (peak signal-to-noise ratio), SSIM (Structural SIMilarity), Mean Square Error (MSE). According to the results obtained from the study, the absolute superiority of the methods against each other was not observed. However, Kriging and RBF interpolation give better results both for numerical data and visual evaluation for image in-painting problems with large area losses.

Keywords: Image in-painting, Interpolation, Cubic interpolation, Kriging interpolation, Radial based function, High dimensional model representation

1. Introduction
When analogue cameras were widely used, the photographs we kept in print were at risk of aging, fading, wear, and thus loss of information. With the advent of digital cameras, the development of computer and storage, and even cloud storage, our habit of storing photographs in print has evolved accordingly. However, this did not eliminate the risk of information loss of our photos. Errors or loss of information were observed in digital photographs during the acquisition and transmission of the photograph. Some of these may be related to the direct quality of the photograph, such as blur and noise, and in some cases, loss of information in certain areas of the picture. In addition, there may be errors such as the unintentional incorporation of undesirable objects into the photo frame during the photo shoot.

Some methods have been developed due to requirements such as eliminating the lack of information in photographs or eliminating unwanted areas. Bertalmio et al. (2000) developed in-painting terminology. In-painting is the art of modifying a picture or video in a way that cannot be easily detected by an ordinary observer and has become a major research area in image processing (Bugeau et al. 2010). The missing or undesirable portion of the image is completed by using intensity levels on adjacent pixels. However, the image in-painting will not be able to restore the original form of the missing part in the picture, it will only fill the missing or undesirable parts, close to the original (Amasidha et al. 2016).

The methods developed for in-painting can be grouped under three main headings.

- **Texture synthesis:** The basis of this approach is the self-similarity principle. It is based on the assumption that similar structures in a picture are often repeated (Bugeau et al. 2010).
- **Exemplar-based approach:** The missing region is filled with information from the known region at the patch level.
- **Partial differential equations and variation-based diffusion techniques:** The method of partial differential equations fills the regions to be uniformly propagated along the isophot directions from neighboring regions along the direction of isophot. The method, which gives good results for small areas, causes turbidity in larger areas. Variation methods address the problem in the form of finding the extremes of energy
functions. However, these models only aim at dealing with non-textural in-painting. The difficulty of real in-painting problems is due to the rapid changes of the isophot and the roughness of the image functions (Chang and Chongxiu 2011).

In addition to this, methods have been developed that consider interiors as an interpolation problem and apply different interpolation techniques to complete the missing areas in the images (Guzel, 2015 and Mutlu et. al., 2014).

In this study, we aimed to evaluate the interpolation methods to handle the image in-painting process as an interpolation problem as the main contribution. State-of-the-art methods were tested on a generated dataset which was subject to both noise and corruption. These methods were then assessed based on their SSIM, PNSR and MSE in a quantitative evaluation. A qualitative evaluation was also performed on the results to ensure that the interpolation approach yielded visually appealing images (Bostanci, 2014 and Seref et. al, 2021).

The rest of the paper is structured as follows: Section 2 presents the current literature and background on the interpolation approaches employed in the paper. This section is followed by Section 3 where the dataset and the evaluation approach is elaborated. Section 4 presents both quantitative and qualitative results, finally the paper is concluded in Section 5.

2. Literature Review and Background
2.1 Cubic Interpolation
This interpolation is one of the common methods used to estimate unknown points. It generates results with smoother transitions and lower error rate than other interpolation techniques with polynomials. It is a commonly used method for filling lost pixels. When an interpolation of surface values is desired in a two-dimensional field, it can be formulated as follows.

On a unit square, if the surface values at the corners and the partial derivative values at these points are known, the convergence values of the points within the square can be obtained by polynomial (1)

\[ p(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j \]  

(1)

The known corner values of the surface and the partial derivative values \((f_x, f_y, f_{xy})\) in the x and y direction which can be obtained from the corner points. The polynomial is obtained by replacing the required equations with unknown coefficients for \(a_{ij}\) which can be solved using matrix format to express the polynomial exactly.

2.2 Kriging Interpolation
Kriging is a geostatistics interpolation method that takes into account the distance and degree of variation between known points when estimating values at unknown points (Firas and Jassim 2013). This approach uses the values of the entire sample to calculate an unknown value. Kriging assumes that the distance or direction between sample points reflects a spatial correlation that can be used to explain variation on the surface. In kriging interpolation, a mathematical function is applied to all points of a specified number or a specified radius to determine the estimation value of each unknown point. The closer the point is, the higher value the weights have. Kriging is the most appropriate approach when there is a spatially related distance or directional trend in the data (spatial auto-correlation) and calculated as follows:

\[ \hat{P} = \sum_{i=1}^{N} \lambda_i P_i \]  

(2)

where \(N: k \times k\) (sample) is the total number of intact pixels, \(\hat{P}\) : pixel to be interpolated, \(P_i: k \times k\) intact pixels in the block, \(\lambda_i: k \times k\) weights of the intact pixels in the block \(\left(\sum_{i=1}^{N} \lambda_i = 1\right)\)
Kriging interpolation, chooses values of \( \lambda_i \) in order to minimize the interpolation variance \( \sigma^2 = E \left[ \left( P - \hat{P} \right)^2 \right] \).

Two steps are required for Kriging interpolation:

1. Dependency rules: Constructing variograms and covariance functions to predict statistical dependence (spatial autocorrelation) based on the autocorrelation model (fitting a model).

   The Variogram is a function of distance and direction that separates the two positions used to measure dependence. Variogram is defined as the variance of the difference between two variables at two different points and is calculated as:

   \[
   2\gamma(h) = \frac{1}{n} \sum_{i=1}^{N} \left[ P(x_i) - P(x_i + h) \right]^2 ; \quad P(x_i), P(x_i + h) ; \quad x_i, x_i + h
   \]

   (3)

2. Estimation: Estimating unknown values.

   The data is used twice to perform these two steps.

Jassim (2013) creates damaged images by using 4 different masks on 10 different grayscale images and uses Kriging interpolation to remove them. Sapkal and Kadbe (2016) also do inpainting using Kriging interpolation. In this study, the results are obtained by using the same masks on 5 different grayscale images.

Sapkal et al. (2016) used Kriging interpolation to remove several types of masks. In the study, the same dataset was used with a different set of masks.

Awati et al. (2017) conducted a study on troubleshooting colour images using modified Kriging interpolation. They separate colour images into RGB components and apply separate interpolation to each component. In practice, 3x3 matrices are used. The masks they use consist of vertical, horizontal and curved lines. They make separate trials for 1,2,3 and 4 pixel thicknesses as line thicknesses in each mask and evaluate the results.

### 2.3 Radial Basis Functions

Chang and Chongxiu (2011) define a mapping between the coordinates and colours of the image pixels, and implement an algorithm based on radial-based functions to generate the best approximation of this mapping in a given neighborhood. Radial-based functions are means of approximating a multivariable function as a linear combination of univariate functions. It is one of the methods that provides good results for the interpolation of scattered data. In order to increase the accuracy of their solutions and reduce the complexity of the algorithm, researchers create the pixel-by-pixel zoom function. For larger loss areas, interpolation of different overlapping coefficients is used.

Wang and Qin (2006) propose an algorithm for image inpainting based on compactly supported radial basis functions (CSRBF). The algorithm transforms the 2-D image inpainting problem from a 3-D point set into a surface reconstruction problem. First, a covered surface is constructed for approximation to the set of dots obtained from the damaged image using radial based functions. The values of the lost pixels are then calculated using this surface.

### 2.4 High Dimensional Model Representation and Lagrange Interpolation

Karaca and Tunga (2016), who consider image in-painting as an interpolation problem, have designed this problem by using the HDMR method and Lagrange interpolation, which allows a multivariate function to be expressed as the sum of multiple functions with less variables.

Normally grayscale images are represented as functions with two variables \( f(x, y) \), \( x \): number of rows, \( y \): number of columns. Similarly, coloured images are represented by a function of three variables \( f(x, y, z) \). In order to apply HDMR, the copy of the image itself is added as an
additional dimension. In other words, a grayscale image, \( f(x, y, n) \), \( n = 1, 2 \); and the color image is expressed as \( f(x, y, z, n) \), \( z = 1, 2, 3, \ n = 1, 2 \) (RGB channels).

HDMR expansion for a multivariate function; fixed term is defined as the sum of functions of one variable, functions of two variables and others.

\[
f(x_1, x_2, ..., x_N) = f_0 + \sum_{i=1}^{N} f_i(x_i) + \sum_{i,j=1}^{N} f_{ij}(x_i, x_j) + ... + f_{12...N}(x_1, x_2, ..., x_N)
\]

In general, functions are represented up to univariate or bivariate functions, and the remainder is ignored as an approximation error. The function created for grayscale images can be fully represented when it is extended up to two-variable functions in the HDMR expansion (Altın and Tunga 2014). The researchers presented a representation by making up to three variable functions for in-painting with color images and aimed to estimate the lost pixels by applying them with Lagrange interpolation.

In another study, Karaca and Tunga (2016) also studied the in-painting of a rectangular area using the same method. They tried 5x5, 10x10 and 20x20 pixel dimensions on different images for the in-painting area.

The algorithms used for in-painting in the literature have also been used for noise removal. Jassim (2013) tried the Kriging algorithm for salt & pepper noise reduction. For noise of varying intensity, it first detects noise using an 8x8 pixel filter on the image, and then applies Kriging interpolation to correct incorrect pixel values in this area.

3. Material and Method
3.1 Material
The internally stained images obtained from the methods used in the studies and their originals were presented with PSNR and SSIM criteria. In addition, although the masks used in the studies seem similar to each other, they have differences. This is a limitation for the exact comparison of algorithms. In order to make a full comparison of the algorithms proposed in this part of the study, we compared the results obtained by using the same masks and images and each of the methods.

256x256 grayscale and color images were used to compare the algorithms. The images used were selected from the images commonly used in image processing research: Lena, Mandril, Peppers, Jetplane and House. The masks applied to the images are created as follows:

- Mask 1: simple curve, drawn with 4 pixel-thick pencil
- Mask 2: A non-condensed font consisting of several lines between 12-19 fonts,
- Mask 3: Intense font created with 12 font letters,
- Mask 4: Intense scratches of oblique, horizontal and vertical lines drawn with a 4-pixel-thick pencil,
- Mask 5: Frame created with a size of 40x40 pixels.

In addition, to test the noise reduction efficiency of interpolation methods, masks created with salt & pepper noise were applied. Their density levels are Noise 1: 10%, Noise 2: 30%, Noise 3: 50%, Noise 4: 70% and Noise 5: 90%, respectively. For instance, the presentation of the grayscale Lena image with the masks to be used is demonstrated in Table 1.

Table 1 Display of the masks used in the study on a sample image

| Mask 1 | Mask 2 | Mask 3 | Mask 4 | Mask 5 |
|--------|--------|--------|--------|--------|


In all of the algorithms, corrected images were obtained by calculating only unknown pixel values and the obtained images were compared with the original image using PSNR (peak signal-to-noise ratio) and SSIM (Structural SIMilarity) criteria. In order to determine the unknown pixel values, the difference between the pixel values between the original image and the false image was taken and the non-0 pixel values were tried to be estimated by the algorithms. In the comparison of the results with the actual image, the mean square error (MSE), which is a part of the SSIM and PSNR criteria, is indicated.

3.2 Method
The evaluation presented in this study employs a number of various interpolation techniques, namely, Two-dimensional cubic interpolation, Kriging interpolation, Interpolation with radial based functions (RBF), Interpolation using High Dimensional Model Representation. These techniques are detailed in the following.

3.2.1 Two-dimensional cubic interpolation
For this method, x, y: row, column coordinate values of known pixels, v: pixel values, x_q, y_q: coordinate values of the desired pixels, cubic: to be used as input parameters for the interpolation. After the information of the two-dimensional image is put into the form that the function can use, the function is executed and the new pixel values are updated.

3.2.2 Kriging interpolation
For this method, kriging (x_i, y_i, z_i, x, y) and auxiliary functions presented by Schwanghart and Kuhn (2010) were used (x_i, y_i: row, column coordinate values of known pixels, z_i: pixel values, x, y: to be the value of the coordinate values of the desired pixels). 256x256 pixel images were used along with masks of size 16x16 pixels masks and noise densities defined in 8x8 pixels neighbourhood. The unknown values in each sub-image were calculated using Kriging function.

In this way, it is ensured that the unknown points in the sub-images are related to all pixel values in the sub-images. The 8x8 pixel size has enough data for noise reduction. However, the Kriging algorithm did not work for Noise5 with a 90% density (see Table 1). Therefore, only 16x16 pixels are used for Noise5. On the other hand, although the calculation method by sub-images is suitable for scattered errors, it will not be suitable for an internal painting problem as in Mask5. Therefore, for the case in Mask5, the image in-painting is reduced to a 90x90 pixel image size that will take the center of the lost frame area (40x40 pixels) and the function is run on this 90x90 pixel neighbourhood.

3.2.3 Interpolation with radial based functions (RBF)
The approach of (Foster 2009) was adopted in order to compute the RBF function in a similar fashion with the Kriging interpolation where sub-images are used for the computation.

3.2.4 Interpolation using High Dimensional Model Representation
For this method, following the approach of Tunga and Koçanoğulları (2017) study; constant, one-variable and two-variable functions representing grayscale images; The fixed, one, two and three variable functions representing the coloured images were found and the missing areas on these functions were corrected by interpolation approach. In this section, spline for interpolation of one-dimensional functions and cubic interpolation for interpolation of two-dimensional functions were implemented.
These methods can be applied directly to grayscale images which can be expressed as two variable functions. Interpolation of coloured images were obtained by applying the above interpolations separately to the three layers of the images. At this point, there is no difference between the method applied to grayscale images. For the calculations, the same algorithms were applied to the red, green, blue (RGB) layers three times and then combined to produce corrected colour images.

4. Results
In-painting results were obtained using the various interpolation methods discussed above. All the results obtained for a sample are presented in Figures 1 (in grayscale) and 2 (in colour).

The numerical comparison results obtained from all images are summarized in Table 2 for grayscale and in Table 3 for colour images. The results of two-dimensional cubic interpolation, Kriging interpolation, RBF interpolation and YBMG interpolation are shown in four large blocks. Orange, blue, green intracellular staining was used to compare the outputs of the four methods. For example, in the PSNR values of 4 different methods used to remove Mask 1 in the colour Lena image, 2-dimensional cubic interpolation has the highest value. This cell was stained with orange. The highest SSIM values of the four methods were painted with blue, while the lowest MSE values of the four methods were painted with green. By means of this staining, it is better to observe which method gives better results in which situation.

In addition, the data of four different methods were compared with the one-way ANOVA test to determine whether there was a difference between the methods in terms of PSNR, MSE and SSIM. There was no difference between the methods in terms of PSNR, MSE and SSIM values in the evaluation (p = 0.997, p = 0.998, p = 0.986 for gray images, p = 0.974, p = 0.994, p = for gray images, respectively), 0.988).

In addition, in order to compare the numerical results of the four methods, the difference between the maximum value and the smallest value of the PSNR and SSIM results obtained for each image and mask was attempted to be observed as a percentage change. These ratios are presented as Figures 3 and 4.

PSNR and MSE values are directly related to each other. As it is known, the MSE value is part of the PSNR criterion. The PSNR value is inversely proportional to the logarithm of the square root of MSE (5) Therefore, it is essential to have the lowest MSE when PSNR is highest.

\[
PSNR = 20 \log_{10}(\frac{255}{\sqrt{MSE}})
\]  
(5)

In addition, the formula of MSE (6) is built on differences in density levels between the two images.

\[
MSE = \frac{1}{NM} \sum_{x=1}^{N} \sum_{y=1}^{M} (f(x,y) - g(x,y))^2
\]  
(6)

In other words, the closer the pixel values in the same position in the two images are, the closer the MSE value is to 0. We understand that the smaller the MSE value and the higher the PSNR, the better the results. On the other hand, the SSIM criterion was formulated to measure luminance, contrast and structural correlation between the two images. The results obtained from functions comparing these three criteria separately are multiplied to find the SSIM value. Each function takes values in the range 0-1, as a result SSIM has values from 0 to 1. The SSIM value approaches to unity when the correlations are high, i.e. the images are similar. Although the PSNR and SSIM calculations do not seem to be alike, there are studies showing that these are analytically related (Horé and Ziou 2010). Therefore, SSIM and PSNR values are expected to be higher in a well obtained in-painting.
As a result of this evaluation; In Table 2 and Table 3, it will be natural for the cells stained with orange, blue and green to coexist. However, in some cases it is observed that this association is not preserved, but this is due to very small differences that can be ignored. Therefore, the outputs we obtained do not contradict theoretical knowledge.

For each image and method in Figure 3 and Figure 4, it was tried to observe the percentage change between the methods by the ratio of the highest and lowest obtained score differences to the lowest value. As can be observed here, the numerical comparison results for images obtained by in-painting are generally very close to each other. The difference in SSIM values does not exceed 5%. PSNR values are mostly within the 5% change band.

In addition, according to the statistics obtained from the one-way ANOVA test, it was confirmed that there was no difference between the methods in numerical evaluation scales. Therefore, it is observed that the methods do not have absolute superiority to each other in terms of in-painting results.

If the methods of application are taken into consideration, the in-painting problem in Mask 5 will need to be evaluated separately. In other masks, the problem of internal painting is scattered throughout the image, where it focuses on a specific area. Therefore, in Kriging and RBF interpolation methods, we focused on the 90x90 pixel area in order to keep the lost area in the middle. It is considered that the selected area would provide sufficient data in order to fill the missing area.

In fact, in the first trials, the whole image was tested using the Kriging and RBF functions at once for both the Mask 5 and the 4 other masks. However, the functions needed during the operation of the array and the required meshgrid structures that need to be produced. When the results obtained for Mask 5 are evaluated; it is observed that Kriging and RBF methods produce better outputs. On the other hand, this advantage will be determined in the evaluation made through observation. The results obtained in Mask 5 can be compared visually with the study of Karaca and Tunga (2016). It is observed that the results obtained for staining of the quadratic region are much larger than the MSE values.
Figure 1 Grayscale sample image with in-painting.
| Corrupt Image | 2D Cubic Interpolation | Kriging Interpolation | RBF Interpolation | HDMR Interpolation |
|---------------|------------------------|----------------------|------------------|-------------------|
| noisy image   | PSNR, SSIM, MSE        | PSNR, SSIM, MSE      | PSNR, SSIM, MSE  | PSNR, SSIM, MSE   |
|               | 37.3694, 0.99567, 11.8163 | 36.3132, 0.99529, 15.1673 | 36.1625, 0.99566, 15.6616 | 37.3481, 0.99565, 11.9147 |
| noisy image   | PSNR, SSIM, MSE        | PSNR, SSIM, MSE      | PSNR, SSIM, MSE  | PSNR, SSIM, MSE   |
|               | 39.0467, 0.99774, 6.3986 | 37.8562, 0.99718, 10.6477 | 37.5792, 0.99662, 11.5342 | 39.0161, 0.99773, 8.1672 |
| noisy image   | PSNR, SSIM, MSE        | PSNR, SSIM, MSE      | PSNR, SSIM, MSE  | PSNR, SSIM, MSE   |
|               | 33.0549, 0.99038, 32.1803 | 32.5236, 0.98854, 36.3678 | 32.4008, 0.9882, 37.4114 | 33.0294, 0.98832, 32.4369 |
| noisy image   | PSNR, SSIM, MSE        | PSNR, SSIM, MSE      | PSNR, SSIM, MSE  | PSNR, SSIM, MSE   |
|               | 32.8171, 0.99604, 33.9917 | 31.5567, 0.98766, 48.1294 | 31.2143, 0.98711, 49.1843 | 32.8124, 0.99604, 34.0384 |
| noisy image   | PSNR, SSIM, MSE        | PSNR, SSIM, MSE      | PSNR, SSIM, MSE  | PSNR, SSIM, MSE   |
|               | 36.1827, 0.99591, 15.6608 | 36.3019, 0.99621, 15.2367 | 36.7859, 0.99937, 13.63 | 36.1634, 0.9959, 15.7303 |
| noisy image   | PSNR, SSIM, MSE        | PSNR, SSIM, MSE      | PSNR, SSIM, MSE  | PSNR, SSIM, MSE   |
|               | 39.9505, 0.99376, 6.577 | 41.0628, 0.99835, 5.091 | 40.443, 0.99812, 5.8719 | 39.9123, 0.99785, 6.6352 |
| noisy image   | PSNR, SSIM, MSE        | PSNR, SSIM, MSE      | PSNR, SSIM, MSE  | PSNR, SSIM, MSE   |
|               | 34.9742, 0.99032, 20.685 | 34.9698, 0.99342, 20.7063 | 34.9297, 0.99277, 20.9197 | 34.9944, 0.9933, 20.7796 |
| noisy image   | PSNR, SSIM, MSE        | PSNR, SSIM, MSE      | PSNR, SSIM, MSE  | PSNR, SSIM, MSE   |
|               | 31.5036, 0.98569, 45.0223 | 31.2815, 0.98493, 48.469 | 31.1229, 0.98414, 50.2096 | 31.5827, 0.98587, 45.1682 |
| noisy image   | PSNR, SSIM, MSE        | PSNR, SSIM, MSE      | PSNR, SSIM, MSE  | PSNR, SSIM, MSE   |
|               | 26.6613, 0.97254, 68.501 | 27.2329, 0.96815, 104.7379 | 26.9026, 0.96827, 102.9969 | 26.6690, 0.97252, 68.5462 |

Figure 2 Colour sample image with in-painting
|          | Lena    | Boose   | Peppers | Mandarin | Japane |
|----------|---------|---------|---------|----------|--------|
| PSNR     | 35.471  | 36.639  | 35.190  | 35.487   | 36.784 |
| PSNR     | 0.98133 | 0.98506 | 0.98543 | 0.98642  | 0.98039 |
| SSIM     | 3.374   | 3.337   | 3.336   | 3.337    | 3.336  |
| SSIM     | 0.98454 | 0.98485 | 0.98522 | 0.98522  | 0.98536 |
| Noise1   | 0.98522 | 0.98522 | 0.98522 | 0.98522  | 0.98522 |
| Noise1   | 46.201  | 41.668  | 49.689  | 11.168    | 33.784 |
| Noise1   | 13.115  | 13.406  | 13.959  | 13.959    | 13.959 |
| Noise1   | 11.823  | 12.682  | 13.693  | 13.693    | 13.693 |
| Noise1   | 10.614  | 10.847  | 10.847  | 10.847    | 10.847 |
| Noise5   | 37.872  | 37.872  | 37.872  | 37.872    | 37.872 |
| Noise5   | 0.98752 | 0.98752 | 0.98752 | 0.98752   | 0.98752 |
| Noise4   | 46.504  | 46.504  | 46.504  | 46.504    | 46.504 |
| Noise4   | 11.844  | 11.844  | 11.844  | 11.844    | 11.844 |
| Noise4   | 46.201  | 46.201  | 46.201  | 46.201    | 46.201 |
| Noise3   | 14.1    | 14.1    | 14.1    | 14.1      | 14.1   |
| Noise3   | 18.45   | 18.45   | 18.45   | 18.45     | 18.45  |
| Noise3   | 18.45   | 18.45   | 18.45   | 18.45     | 18.45  |
| Noise3   | 18.45   | 18.45   | 18.45   | 18.45     | 18.45  |
| Noise3   | 18.45   | 18.45   | 18.45   | 18.45     | 18.45  |
| Noise2   | 18.45   | 18.45   | 18.45   | 18.45     | 18.45  |
| Noise2   | 18.45   | 18.45   | 18.45   | 18.45     | 18.45  |
| Noise2   | 18.45   | 18.45   | 18.45   | 18.45     | 18.45  |
| Noise2   | 18.45   | 18.45   | 18.45   | 18.45     | 18.45  |
| Noise2   | 18.45   | 18.45   | 18.45   | 18.45     | 18.45  |
| Noise1   | 18.45   | 18.45   | 18.45   | 18.45     | 18.45  |
| Noise1   | 18.45   | 18.45   | 18.45   | 18.45     | 18.45  |
| Noise1   | 18.45   | 18.45   | 18.45   | 18.45     | 18.45  |
| Noise1   | 18.45   | 18.45   | 18.45   | 18.45     | 18.45  |
| Method       | Lena        | House       | Peppers      | Mandrill     | Apocalypse   |
|--------------|-------------|-------------|--------------|--------------|--------------|
|             | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| Cubic        | 0.8977 | 0.9781 | 0.8863 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 |
| Kriging      | 0.9003 | 0.9781 | 0.8863 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 |
| RBF          | 0.9003 | 0.9781 | 0.8863 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 |
| HDRM         | 0.8977 | 0.9781 | 0.8863 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 | 0.9781 | 0.9297 |
According to the results, the square area is still clearly visible and filled with diagonal lines. In our results, the square area (for Kriging and RBF interpolation) was less prominent and no diagonal lines were observed in the faulty area. The difference between grayscale and colour images of the two studies may be inaccurate, but the interiors of in-painting characters can still be observed. Although relatively, the results obtained in our study can be more visually appealing.
5. Conclusion

The literature lacks a detailed evaluation on the use of interpolation methods for the in-painting problem. Accordingly, this study aimed to evaluate state-of-the-art approaches, namely, two dimensional cubic, Kriging, RBF and YBMG based Lagrange interpolation methods, which are recently studied in the literature.

A benchmark dataset was generated in order to perform a detailed evaluation of the in-painting algorithms. Several comprehensive experiments were conducted in order to make a fair assessment. According to the results obtained, it is observed that the methods do not have absolute superiority to each other in terms of in-painting results. However, Kriging and RBF interpolation produce better results both for numerical data and visual evaluation for image inpainting problems having large area losses. Studies on this subject have also yielded good results for large area in-painting, but Kriging and RBF interpolation outputs look better in terms of visual quality.

References

Altın, E. M., & Tunga, B. (2014). High Dimensional Model Representation in Image Processing. *Proceedings of the 14th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE*, 55–64. https://doi.org/10.1063/1.4904687

Amasidha, K. S., Awati, A. S., Deshpande, S. P., & Patil, M. R. (2016). Digital Image Inpainting: A Review. *International Research Journal of Engineering and Technology (IRJET)*, 03(01), 851–854. Tarihinde adresinden erişildi https://www.irjet.net/archives/V3/i1/IRJET-V3I1150.pdf

Awati, A. S., Deshpande, S. P., Belagal, P. Y., & Patil, M. R. (2017). Digital Image Inpainting Using Modified Kriging Algorithm. *2nd International Conference for Convergence in Technology (I2CT)*, 945–950.

Bertalmio, M., Sapiro, G., Caselles, V., & Ballester, C. (2000). Image Inpainting. *SIGGRAPH Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 417–424.

Bostanci, E., (2014). Is Hamming distance only way for matching binary image feature descriptors?, Electronics Letter 50(11), 806-808

Bugeau, A., Bertalmio, M., Caselles, V., & Sapiro, G. (2010). A Comprehensive Framework for Image Inpainting. *IEEE Transactions on Image Processing, 19*(10), 2634–2645.

Chang, L., & Chongxiu, Y. (2011). New Interpolation Algorithm for Image Inpainting. İçinde *International Conference on Physics Science and Technology (ICPST)* (C. 22, ss. 107–111). Elsevier B.V. https://doi.org/10.1016/j.phpro.2011.11.017

Foster, M. (2009). MATLAB Multivariate Interpolation Toolbox. Tarihinde 01 Eylül 2018, adresinden erişildi https://github.com/mattfoster/matlab-interpolation-toolkit

Guzel M.S. (2015) , Performance evaluation for feature extractors on street view images, The Imaging Science Journal, 64(1), 26-33.

Horé, A., & Ziou, D. (2010). Image quality metrics : PSNR vs . SSIM. İçinde *International Conference on Pattern Recognition* (ss. 2366–2369). https://doi.org/10.1109/ICPR.2010.579

Jassim, F. A. (2013). Image Inpainting by Kriging Interpolation Technique. *World of
Jassim, F. A. (2013). Kriging Interpolation Filter to Reduce High Density Salt and Pepper Noise. Computer Science and Information Technology Journal (WCSIT), 3(5), 91–96.

Karaca, E., & Tunga, M. A. (2016a). A Method for Inpainting Rectangular Missing Regions using High Dimensional Model Representation and Lagrange Interpolation. 24th Signal Processing and Communication Application Conference (SIU) Signal Processing and Communication Application Conference (SIU).

Karaca, E., & TUNGA, M. A. (2016b). Interpolation-Based Image Inpainting in Colour Images Using High Dimensional Model Representation. 24th European Signal Processing Conference (EUSIPCO), 2425–2429.

https://doi.org/10.1109/EUSIPCO.2016.7760684

Mutlu, B., Haciomeroglu, M., Guzel, M.S. et al. (2014). Silhouette Extraction from Street View Images. International Journal of Advanced Robotic Systems, 1-12 .

Sapkal, M. S., Kadbe, P. (2016). Kriging Interpolation Technique Basis Image Inpainting. International Journal of Latest Trends in Engineering and Technology (IJLTET), 6(3), 467–470.

Sapkal, M. S., Kadbe, P. K., & Deokate, B. H. (2016). Image Inpainting By Kriging Interpolation Technique For Mask Removal. International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 310–313.

Seref , B., Bostanci, G. E, Guzel, M.S. (2021). Evolutionary neural networks for improving the prediction performance of recommender systems, Turkish Journal of Electrical Engineering & Computer Sciences, 29(1), 62-77.

Schwanghart, W., & Kuhn, N. J. (2010). TopoToolbox : A set of Matlab functions for topographic analysis. Environmental Modelling and Software, 25, 770–781. https://doi.org/10.1016/j.envsoft.2009.12.002

Tunga, B., & Koçanaoğulları, A. (2017). Digital image decomposition and contrast enhancement using high-dimensional model representation. Signal, Image and Video Processing, 299–306. https://doi.org/10.1007/s11760-017-1158-8

Wang, W., & Qin, X. (2006). An Image Inpainting Algorithm Based on CSRFB Interpolation. Dianzi Yu Xinxi Xuebao/Journal of Electronics and Information Technology, 890–894.