In [15]:

```
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt
import os
from helperFunctions import add_impulse_noise
from imfPython import imf
from skimage.metrics import peak_signal_noise_ratio as psnr
from skimage.metrics import structural_similarity as ssim

data_dir = 'test_images'

#set matplotlib size
plt.rcParams['figure.figsize'] = [10, 10]
```

In [16]:

```
# set image
image_name = 'lena_gray_512.tif'
```

In [17]:

```
# load image
img_path = os.path.join(data_dir,image_name)
if not os.path.exists(img_path):
    raise FileExistsError

img = cv.imread(img_path,0)
plt.imshow(img,cmap='gray')
```

Out[17]:

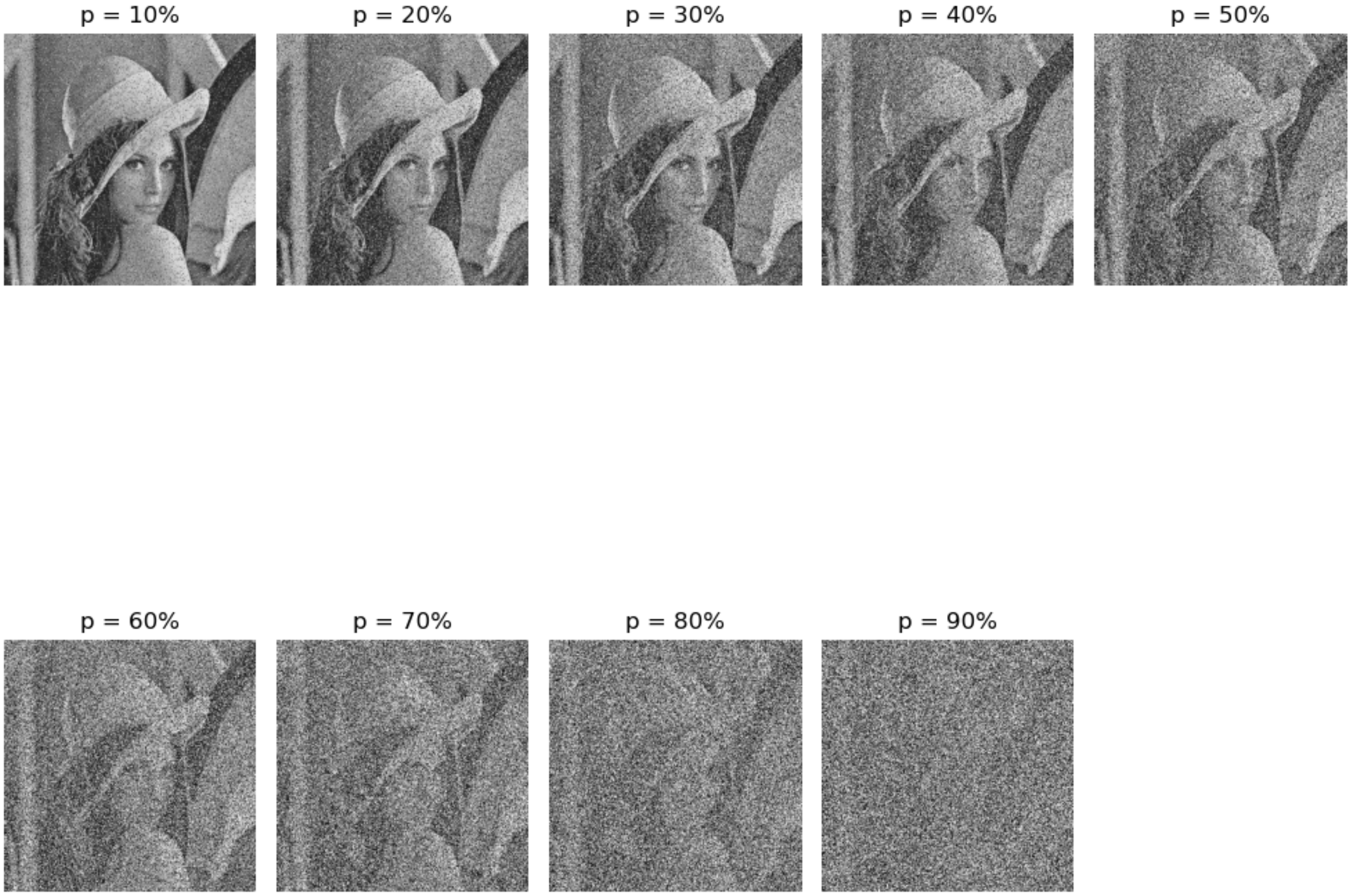
```
<matplotlib.image.AxesImage at 0x1fc33e86e00>
```

In [18]:

```
# add impulse noise from 10% to 90%
noisyImages = list()
for i in range(1,10):
    timg = np.copy(img)
    noisyImages.append(add_impulse_noise(timg,0.1*i))
```

In [19]:

```
# plot the images in noisyImages as a 2x5 grid, noise_ratio starts from 0.1 to 0.9
fig, axs = plt.subplots(2,5)
ctr = 0
for i in range(2):
    for j in range(5):
        if ctr == 9:
            axs[i,j].axis('off')
            break
        axs[i,j].imshow(noisyImages[i*5+j],cmap='gray')
        axs[i,j].set_title('p = {}%'.format(10*(i*5+j+1)))
        axs[i,j].axis('off')
        ctr+=1
plt.tight_layout()
```

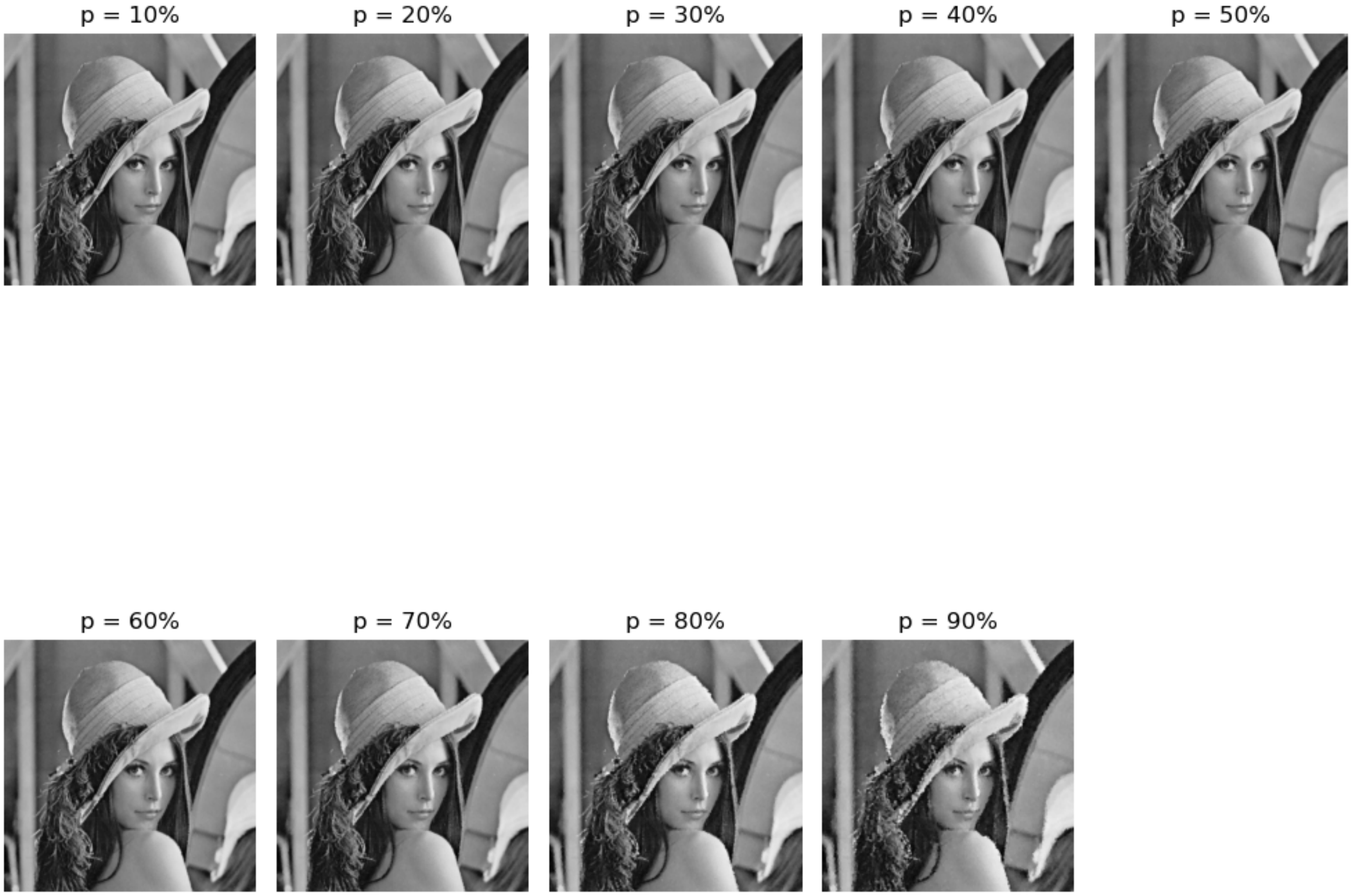


In [ ]:

```
filteredImages = list()
for i in range(9):
    filteredImages.append(imf(noisyImages[i]))
```

In [21]:

```
# plot filtered images
fig, axs = plt.subplots(2,5)
ctr = 0
for i in range(2):
    for j in range(5):
        if ctr == 9:
            axs[i,j].axis('off')
            break
        axs[i,j].imshow(filteredImages[i*5+j],cmap='gray')
        axs[i,j].set_title('p = {}%'.format(10*(i*5+j+1)))
        axs[i,j].axis('off')
        ctr+=1
fig.tight_layout()
```



p = 10%

p = 20%

p = 30%

p = 40%

p = 50%

p = 60%

p = 70%

p = 80%

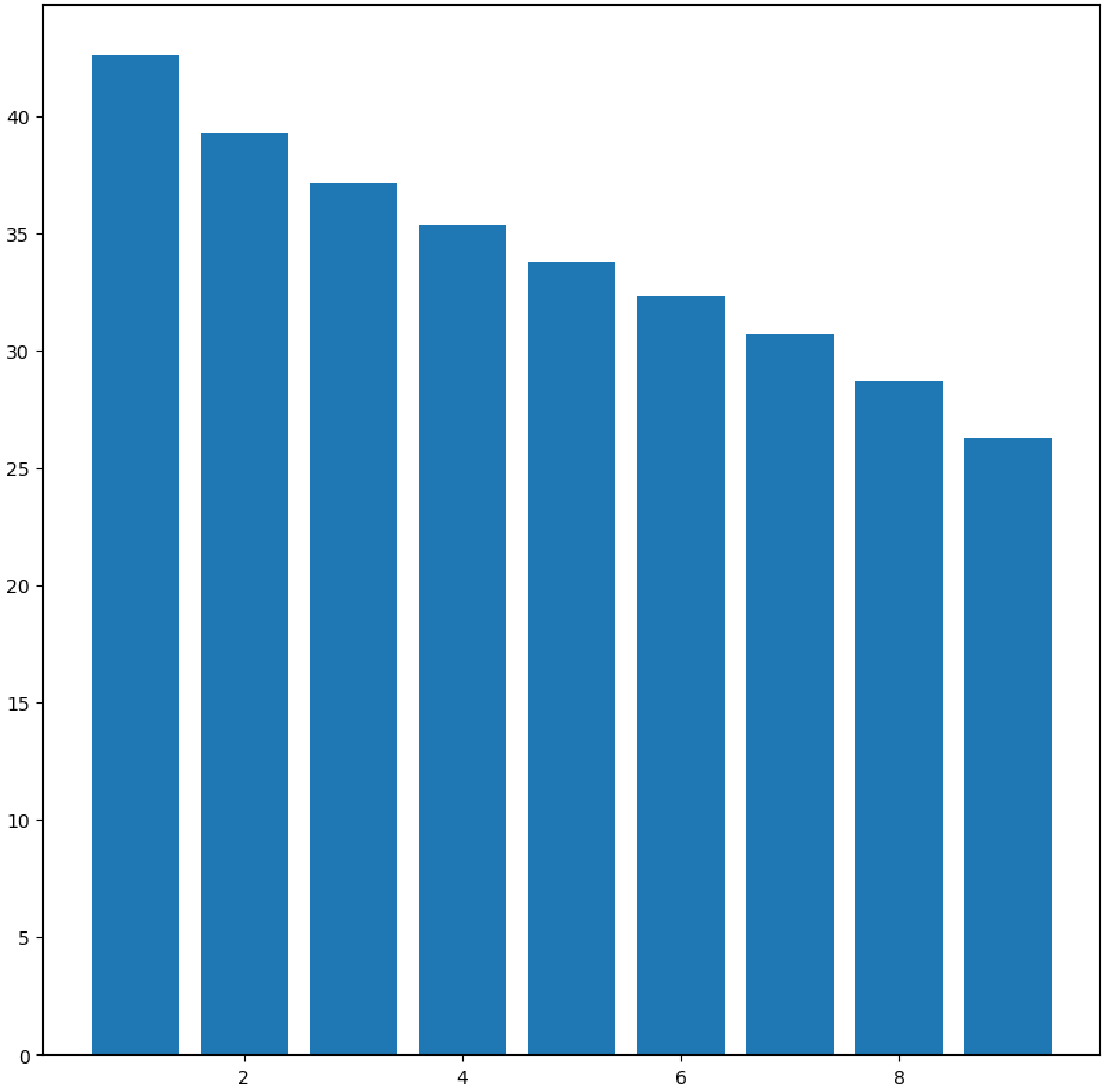
p = 90%

In [22]:

```
psnr_list = list()
for i in range(9):
    psnr_list.append(psnr(img,filteredImages[i]))
plt.bar(range(1,10),psnr_list)
```

Out[22]:

```
<BarContainer object of 9 artists>
```

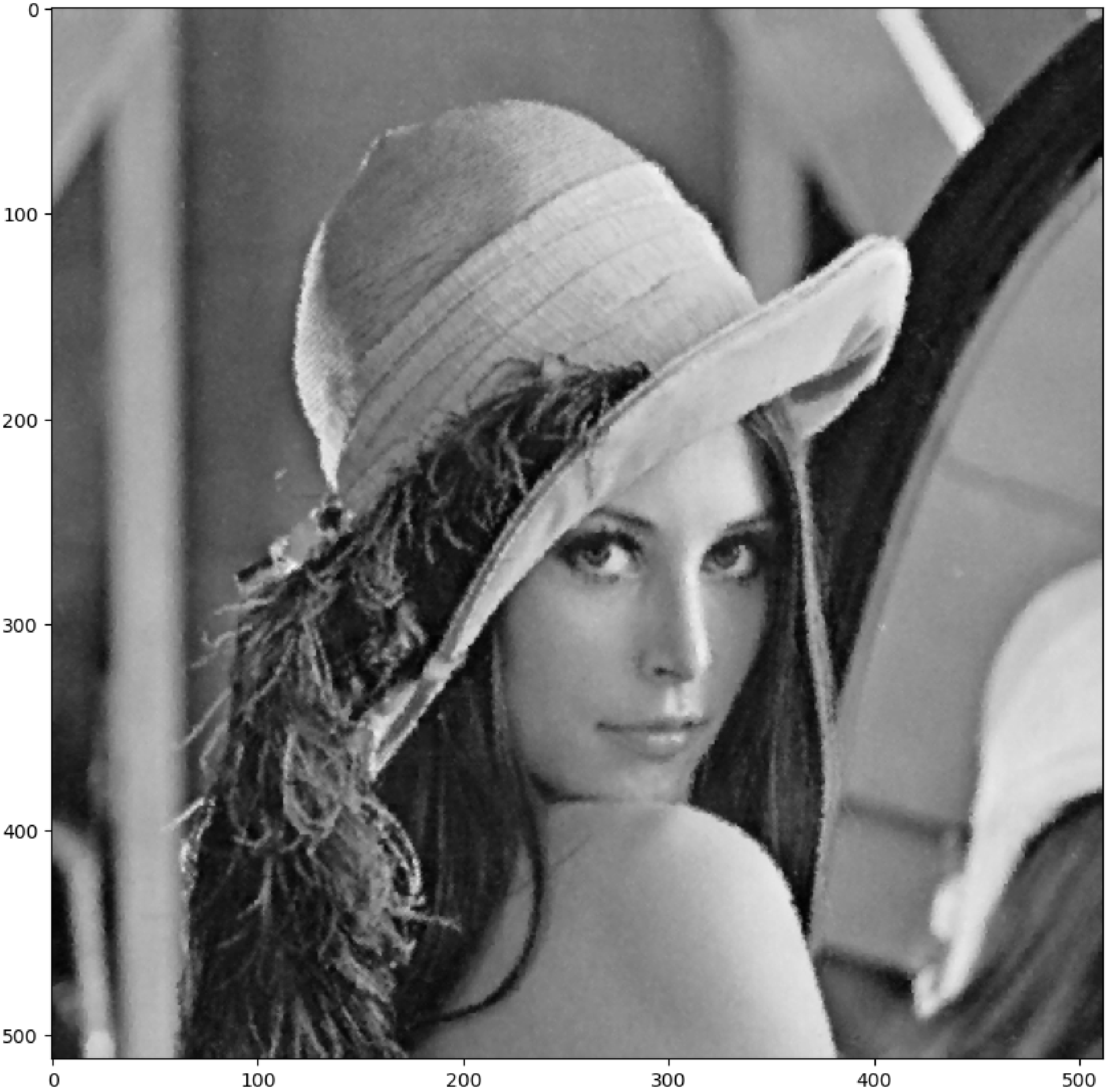


In [23]:

```python
plt.imshow(filteredImages[5],cmap='gray')
```

Out[23]:

```
<matplotlib.image.AxesImage at 0x1fc33edfcd0>
```



In [14]: