```python
# importing the necessary libraries (OpenCV and NumPy)
import cv2
import numpy as np

# Load the image in grayscale
image_path = 'images.jpeg'
image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

# Get the dimensions of the image
height, width = image.shape

# Create a video writer to save the output video
output_video = '220969.mp4'
fourcc = cv2.VideoWriter_fourcc(*'mp4v')  # Define the codec for the video
fps = 2  # Frames per second
video_writer = cv2.VideoWriter(output_video, fourcc, fps, (width, height), False)

# Function to extract bit planes from the image
def extract_bit_planes(image):
    bit_planes = []  # List to store individual bit planes
    for i in range(8):
        # Extract the i-th bit plane
        bit_plane = (image & (1 << i)) >> i
        bit_planes.append(bit_plane)
    return bit_planes

# Extract bit planes from the image
bit_planes = extract_bit_planes(image)

# Initialize image to accumulate bit planes for reconstruction
accumulated_image = np.zeros_like(image)

# Add bit planes and save each step as a video frame
for i in reversed(range(8)):  # Start from most significant bit
    # Accumulate current bit plane to reconstruct the image
    accumulated_image += bit_planes[i] * (1 << i)

    # Normalize the accumulated image to be within the range 0-255
    frame = cv2.normalize(accumulated_image, None, 0, 255, cv2.NORM_MINMAX)

    # Write frame to the video
    video_writer.write(frame.astype(np.uint8))

# Release video writer to finalize the video
video_writer.release()
```