```
import cv2
import numpy as np

def skeletonize(image_path):
    # Load the image in grayscale
    image = cv2.imread(image_path, 0)

    # Apply a binary threshold to obtain a binary image
    _, binary_image = cv2.threshold(image, 127, 255, cv2.THRESH_BINARY)

    # Invert the binary image
    inverted_image = cv2.bitwise_not(binary_image)

    # Initialize the skeleton image
    skeleton = np.zeros_like(inverted_image)

    # Apply morphological operations until the image is completely skeletonized
    while cv2.countNonZero(inverted_image) > 0:
        # Erode the inverted image
        eroded = cv2.erode(inverted_image, None)

        # Open the eroded image
        opened = cv2.morphologyEx(eroded, cv2.MORPH_OPEN, None)

        # Subtract the opened image from the eroded image
        subtracted = cv2.subtract(eroded, opened)

        # Bitwise OR the subtracted image with the skeleton image
        skeleton = cv2.bitwise_or(skeleton, subtracted)

        # Set the inverted image to the eroded image for the next iteration
        inverted_image = eroded.copy()

    # Show the skeleton image
    cv2.imshow("Skeleton", skeleton)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

# Example usage
image_path = "path/to/your/image.jpg"
skeletonize(image_path)
```