```
import cv2
import numpy as np
import sys
import os

def count_colored_pixels(image_path):
  """
  Counts the number of colored (non-grayscale) pixels in an image,
  and creates a new image highlighting these pixels in red.

  Args:
     image_path (str): The path to the input image file.
  """
  # 1. Load the image
  image = cv2.imread(image_path)
  if image is None:
     print(f"Error: Could not load image at '{image_path}'")
     return

  # 2. Convert the image from BGR to HSV colorspace
```

```
hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

# 3. Define the HSV range for the thermal colors (greens and blues).
# In OpenCV, Hue is in range [0, 179].
# We also set a minimum saturation and value to exclude grayscale and black pixels.
lower_bound = np.array([35, 50, 50])  # Lower bound for green (Hue, Sat, Val)
upper_bound = np.array([130, 255, 255]) # Upper bound for blue (Hue, Sat, Val)

# 4. Create a mask for pixels within the specified thermal color range
colored_mask = cv2.inRange(hsv_image, lower_bound, upper_bound)

# 5. Count the number of colored pixels
colored_pixel_count = cv2.countNonZero(colored_mask)
print(f"Number of colored pixels found: {colored_pixel_count}")

# 6. Create an output image to highlight the counted pixels
output_image = np.zeros_like(image)
output_image[colored_mask > 0] = (0, 0, 255) # Red in BGR

# 7. Save the output image
base_name = os.path.basename(image_path)
name, ext = os.path.splitext(base_name)
output_path = f"{name}_counted_pixels.png"

cv2.imwrite(output_path, output_image)
print(f"Output image saved to: {output_path}")
```

```
if __name__ == "__main__":
  if len(sys.argv) != 2:
      print("Usage: python pixel_counter.py <path_to_image>")
      sys.exit(1)

  input_path = sys.argv[1]
  count_colored_pixels(input_path)
```