```
import cv2
import numpy as np
import sys


def count_pixels_in_frame(frame):
  """Counts colored pixels in a single video frame using the established HSV range."""
  hsv_image = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
  lower_bound = np.array([35, 50, 50])
  upper_bound = np.array([130, 255, 255])
  colored_mask = cv2.inRange(hsv_image, lower_bound, upper_bound)
  return cv2.countNonZero(colored_mask)


def analyze_video_pixels(video_path):
  """
  Processes an entire video to gather pixel count statistics for each frame.
  """
  cap = cv2.VideoCapture(video_path)
  if not cap.isOpened():
      print(f"Error: Could not open video file {video_path}")
```

```
    return

frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
print(f"Analyzing video: {video_path} ({frame_count} frames)")

pixel_counts = []
frame_idx = 0

while True:
    ret, frame = cap.read()
    if not ret:
        break  # End of video

    count = count_pixels_in_frame(frame)
    pixel_counts.append(count)

    frame_idx += 1
    if frame_idx % 100 == 0:
        print(f"Processed {frame_idx}/{frame_count} frames...")

cap.release()
print("\nVideo analysis complete.")
print("-" * 30)

if not pixel_counts:
    print("No frames were processed.")
    return
    counts_array = np.array(pixel_counts)

    max_count = np.max(counts_array)
    min_count = np.min(counts_array)
    mean_count = np.mean(counts_array)
    median_count = np.median(counts_array)
    std_dev = np.std(counts_array)

    print("Pixel Count Statistics:")
    print(f"  - Maximum: {max_count:,} pixels")
    print(f"  - Minimum: {min_count:,} pixels")
    print(f"  - Average (Mean): {mean_count:,.2f} pixels")
    print(f"  - Median: {median_count:,.2f} pixels")
    print(f"  - Standard Deviation: {std_dev:,.2f}")
    print("-" * 30)


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

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