28-11-24
Parallel Cellular Algorithm
Algorithm:-

```
import numpy as np

def noise_reduction(grid, iterations):
    """
    Perform noise reduction on a 2D grid using cellular automata rules.
    Each cell is updated by averaging itself and its neighbors.
    """
    rows, cols = grid.shape
    for _ in range(iterations):
        # Create a copy of the grid to apply updates
        new_grid = grid.copy()
        for i in range(1, rows - 1):
            for j in range(1, cols - 1):
                # Update each cell by averaging its value with neighbors
                neighbors = [
                    grid[i - 1][j], grid[i + 1][j], grid[i][j - 1], grid[i][j + 1],
                    grid[i - 1][j - 1], grid[i - 1][j + 1], grid[i + 1][j - 1], grid[i + 1][j + 1]
                ]
                new_grid[i][j] = int((grid[i][j] + sum(neighbors)) / (len(neighbors) + 1))
        grid = new_grid  # Update the grid for the next iteration
    return grid


def main():
    # Take user input for grid dimensions and noise level
    rows = int(input("Enter number of rows for the grid: "))
    cols = int(input("Enter number of columns for the grid: "))
    iterations = int(input("Enter the number of iterations for noise reduction: "))

    # Generate a random noisy grid
    np.random.seed(0)  # For reproducibility
    grid = np.random.randint(0, 256, (rows, cols))
    print("\nOriginal Grid (Noisy Image):")
    print(grid)

    # Perform noise reduction
    reduced_grid = noise_reduction(grid, iterations)
    print("\nReduced Noise Grid (After Noise Reduction):")
    print(reduced_grid)
```

```
    print("\nOutput by Devanshi Slathia: Noise reduction completed successfully!")


if __name__ == "__main__":
    main()
```

Output:-

```
Enter number of rows for the grid: 5
Enter number of columns for the grid: 5
Enter the number of iterations for noise reduction: 2

Original Grid (Noisy Image):
[[172  47 117 192  67]
 [251 195 103   9 211]
 [ 21 242  36  87  70]
 [216  88 140  58 193]
 [230  39  87 174  88]]

Reduced Noise Grid (After Noise Reduction):
[[172  47 117 192  67]
 [251 122 116 119 211]
 [ 21 134 113 122  70]
 [216 118 108 114 193]
 [230  39  87 174  88]]

Output by Devanshi Slathia: Noise reduction completed successfully!
```