Usage Example

Type	Technical Spec
Created	@October 20, 2024 3:01 AM
Last Edited Time	@October 22, 2024 8:19 PM
Last Edited By	Aman Agrawal @doubleCradle

Example Results

Original Frame

Figure 1: Original Frame from the Video

Modified Frame

Figure 2: Modified Frame with Macroblocking Artifacts

Sincere apologies to Rowan Atkinson

Description of Results

- **Original Frame**: This image represents a frame extracted from the original video. It shows the content before any modifications are applied.
- **Modified Frame**: This image illustrates the same frame after applying macroblocking artifacts. The artifacts are created by shifting blocks of pixels, resulting in a distorted appearance that simulates compression errors.

Conclusion

The modified frames can be used for training models to detect macroblocking artifacts in video streams. By comparing the original and modified frames, you can visually assess the effectiveness of the synthetic data generation process.

Usage Example for Macroblocking Data Generation

Step-by-Step Guide

1. Prepare Your Environment

Ensure you have the following:

- Python installed on your machine.
- Required libraries: `opencv-python`, `numpy`, `tqdm`.
- `ffmpeg` installed and accessible from your command line.

2. Organize Your Directory Structure

Directory structure is as follows:

```
|   configurations.py
|   datagen.py
|   main.py
|   README.md
|   requirements.txt
|
+---dataset
|   |   dataset.csv
|   |
|   +---extracted_frames
|   |       (empty or pre-filled with images)
|   |
|   \---modified_frames
|           (will be populated after processing)
|
\---videos
        (place your video files here, e.g., test.mp4)
```

3. Configure Parameters

Open `configurations.py` and modify the parameters as needed. For example:

```
pythonCopy Code
DEMO_CONFIG = {
    'data': 'dataset',
    'input': 'videos',
    'frames': 'dataset/extracted_frames',
    'mod_frames': 'dataset/modified_frames',
    'params': {
        'block_size_min': 10,
        'block_size_max': 50,
        'max_shift_value_min': 5,
        'max_shift_value_max': 20,
        'artifact_percentage_min': 0.1,
        'artifact_percentage_max': 0.7,
    },
    'variations': 5,  # Number of variations for each frame
    'useSeed': False,
    'seed': 43,
    'fps': 1,  # Extract 1 frame per second
}
```

4. Add Video Files

Place any video files you want to process in the `videos` folder. You can add as many videos as you like.

5. Run the Script

Open a terminal or command prompt, navigate to the directory containing your scripts, and run the following command:

```
python main.py
```

6. Results

- **Frame Extraction**: The script will extract frames from the videos in the `videos` folder and save them in the `dataset/extracted_frames` folder.
- **Frame Modification**: The script will then apply macroblocking artifacts to each extracted frame based on the parameters defined in `configurations.py`. Each modified frame will be saved in the `dataset/modified_frames` folder.
- **CSV File Generation**: After processing all frames, a CSV file named `dataset.csv` will be created in the `dataset` folder. This file will contain rows for each modified image, including:
  - `image_name`: The name of the modified image.
  - `block_size`: The size of the blocks used for macroblocking.
  - `max_shift_value`: The maximum shift value applied to the blocks.
  - `artifact_percentage`: The percentage of blocks that were modified.
  - `horizontal`: Direction of the shift (True for horizontal, False for vertical).
  - `start_from_top_or_left`: Starting point for the block copying (True for top/left, False for bottom/right).

Example CSV Row

The generated CSV file will look something like this:

```
image_name,block_size,max_shift_value,artifact_percentage,h
orizontal,start_from_top_or_left
test_0001.jpg,25,10,0.5,True,False
test_0002.jpg,15,7,0.3,False,True
...
```

Conclusion

By following these steps, you can easily generate synthetic data with macroblocking artifacts from your video files. Adjust the parameters in `configurations.py` to customize the artifact generation process according to your needs.
