SD TO HD VIDEO CONVERSION TOOL USING SRGAN

REPORT

PREPARED BY

NAME	**: MANOJ N**
DEGREE	**: BE.CSE**
REGISTER NUMBER	**: 711720104040**
COLLEGE	**: KGISL INSTITUTE OF TECHNOLOGY**

ABSTRACT

The demand for high-definition (HD) video content has grown rapidly with the advancement of digital media consumption. However, there is a significant number of existing videos in Standard Definition (SD), which calls for efficient and effective resolution enhancement techniques. This report presents a video conversion tool that utilizes the Super-Resolution Generative Adversarial Network (SRGAN) to upscale video resolution from SD to HD. The SRGAN model is well-known for its ability to generate high-quality images by learning intricate details from training data. The project consists of several critical stages. It begins with data acquisition and preparation, where necessary datasets and pre-trained model weights are downloaded and organized. The input video is then processed using the OpenCV library to extract individual frames. Each frame undergoes super-resolution processing through the SRGAN model, which utilizes deep convolutional neural networks and residual blocks to enhance image quality. The model's performance is further improved using a combination of content loss, adversarial loss, and perceptual loss, ensuring a high-fidelity output. Once all frames have been super-resolved, they are reassembled into a video format, maintaining the original frame rate to provide a seamless viewing experience. The final result is a high-definition video that exhibits significantly improved resolution and visual quality compared to the original SD version. This comprehensive report provides a detailed account of each step in the implementation process. It discusses the architecture of the SRGAN model and highlights the challenges faced during development, along with their corresponding solutions. The tool showcases the practical application of advanced deep learning techniques in video processing, offering a valuable solution for upgrading legacy video content to meet contemporary HD standards.

TABLE OF CONTENTS

1. INTRODUCTION

The objective of this project was to create a video conversion tool that improves the quality of videos by increasing their resolution from Standard Definition (SD) to High Definition (HD). To accomplish this, the Super Resolution Generative Adversarial Network (SRGAN) model was utilized. This report provides an overview of the tool's design and implementation, including the workflow, the SRGAN model, and the methodology employed.

2. TOOL DESIGN

2.1 ARCHITECTURE OVERVIEW

1. **Data acquisition and preparation:** Data acquisition and preparation involve the retrieval and arrangement of essential datasets and model weights.
2. **Frame extraction:** Frame extraction is the process of isolating individual frames from the input video for further analysis.
3. **Super-resolution processing:** Frame extraction entails the utilization of the SRGAN model on each frame to improve its resolution.
4. **Frame compilation:** Frame compilation involves reconstructing the enhanced frames into a high-definition video.
5. **Output generation:** Output generation includes the preservation of the final high-definition video.

3. IMPLEMENTATION

3.1 DATA ACQUISITION AND PREPARATION

The initial phase entailed obtaining the essential data, such as the input video, and retrieving the pre-trained SRGAN model weights. This data was structured and decompressed to simplify accessibility throughout the processing phases.

3.2 FRAME EXTRACTION

The video input was processed through the OpenCV library, where each frame was isolated and stored as a separate image file. This process was essential as the SRGAN model functions on individual images rather than continuous video feeds. By segmenting the video into distinct frames, we were able to independently apply the model to each frame.

3.3 SUPER-RESOLUTION PROCESSING

The resolution of the extracted frames was enhanced using the SRGAN model, which is a type of Generative Adversarial Network (GAN). The SRGAN model consists of two main components: the Generator and the Discriminator.

Generator: The Generator component is responsible for generating high-resolution images from low-resolution inputs. It achieves this by utilizing deep convolutional neural networks with residual blocks, which contribute to its high performance and stability during training.

Discriminator: On the other hand, the Discriminator component plays a crucial role in distinguishing between real high-resolution images and the images generated by the Generator. It provides adversarial feedback, thereby encouraging the Generator to produce more realistic images.

To ensure the quality and similarity to the ground truth, the SRGAN model incorporates a combination of content loss, adversarial loss, and perceptual loss. Content loss measures the difference between the generated image and the ground truth high-resolution image using a pre-trained VGG network. Adversarial loss, on the other hand, encourages the Generator to produce images that are indistinguishable from real images. Lastly, perceptual loss combines both content and adversarial losses to ensure high perceptual quality and similarity to the ground truth. In the process, each extracted frame is passed through the SRGAN generator, resulting in a super-resolved version of the frame.

3.4 FRAME COMPILATION

Once all frames had undergone super-resolution, they were aggregated into a video format. This process entailed establishing a video writer object, configuring the suitable frame rate and resolution, and systematically appending each super-resolved frame to the video writer.

3.5 OUTPUT GENERATION

The high-definition video was stored in a designated format. This resultant video was created by merging all the super-resolved frames, leading to a notable enhancement in resolution and visual clarity in contrast to the initial SD video.

4. CONCLUSION

The video conversion tool, which converts SD to HD videos, effectively utilized the SRGAN model to enhance the resolution of the videos. The primary objective of the design was to divide the video into smaller frames, implement advanced super-resolution techniques, and then reconstruct these frames to create a coherent HD video. This tool successfully showcased the efficacy of employing GANs, particularly SRGAN, to enhance video quality. Furthermore, it emphasized the possibility of future advancements in video processing technologies.

5. SOURCE CODE

```
import gdown
import os
import cv2
from model import resolve_single
from utils import load_image
from model.srgan import generator
from keras.preprocessing.image import array_to_img, save_img

# Step 1: Data Acquisition and Preparation
# Downloading the dataset
file_id = '150FNex0OfXOJzKciXRdo8Yu-lYV5ijUA'
url = f'https://drive.google.com/uc?id={file_id}'
output = 'dataset.zip'
gdown.download(url, output, quiet=False)
os.system('unzip dataset.zip')

# Downloading the model weights
file_id = '1rft4jXu1Fs_vMMPIK8yXeEHvKlL1hJyp'
url = f'https://drive.google.com/uc?id={file_id}'
output = 'weights.zip'
gdown.download(url, output, quiet=False)
os.system('unzip weights.zip')

# Step 2: Frame Extraction
# Downloading the input video
```

```
file_id = '11pj6mGTgZgWEF_P8cILgEtaGoEqeVe9K'
url = f'https://drive.google.com/uc?id={file_id}'
output = 'sample.mp4'
gdown.download(url, output, quiet=False)

# Reading the video from specified path
cam = cv2.VideoCapture("sample.mp4")
fps = cam.get(cv2.CAP_PROP_FPS)
print(fps)

# Creating a folder named data
if not os.path.exists('data'):
    os.makedirs('data')

# Frame extraction
currentframe = 0
arr_img = []
while True:
    ret, frame = cam.read()
    if ret:
        name = './data/frame' + str(currentframe) + '.jpg'
        cv2.imwrite(name, frame)
        currentframe += 1
        arr_img.append(name)
    else:
        break
```

```
cam.release()
cv2.destroyAllWindows()

# Step 3: Super-Resolution Processing
# Load SRGAN model
model = generator()
model.load_weights('weights/srgan/gan_generator.h5')

# Super-resolve frames
arr_output = []
for img_path in arr_img:
    lr = load_image(img_path)
    sr = resolve_single(model, lr)
    arr_output.append(sr)

# Step 4: Saving Super-Resolved Frames
s_res = []
for i, sr in enumerate(arr_output):
    out_name = f'./data/output_images/frame{i:03d}.jpg'
    img_pil = array_to_img(sr)
    save_img(out_name, img_pil)
    s_res.append(out_name)

# Step 5: Creating the Output Video
# Video properties
fps = 20
```

```
size = (arr_output[0].shape[1], arr_output[0].shape[0])


# Create video writer
out = cv2.VideoWriter('output.mp4', cv2.VideoWriter_fourcc(*'DIVX'), fps, size)


# Write frames into video
for img_path in s_res:
    img = cv2.imread(img_path)
    out.write(img)
out.release()
```