Deep Learning Project , Group 3

Name: Guruksha Gurnani, G27849047

Introduction

The increasing prevalence of online video content has brought the challenge of identifying and moderating sensitive materials like–Violence. Simply put, the project aims to create a video classification system capable of identifying violent content in video and producing annotated outputs to highlight violent and nonviolent segments.
Our initial thought process as a team was limited to providing a binary classification results but later was enhanced to include annotated video outputs for better interpretability

Dataset overview

The dataset used in this project was sourced from Kaggle contains:

- 1000 Violence videos : These include real street fight situations in various environments and conditions
- 1000 Non-Violence Videos: These depict diverse human activities like sports, eating , walking and other non violent actions

The videos were collected from YouTube and cover a wide range of scenarios to ensure robustness, each video was labeled as either violent or non violent and preprocessing steps involved frame extractions and resizing to ensure compatibility with the model

Team Contributions
Raghav Agarwal & Anirudh Rao: Understanding and developing the static model.
Dhanush Bhargav & Guruksha: Developing a model that incorporates delay to evaluate how the sequence of frames over time enhances the model's ability to detect violence in videos

Individual contribution:

Frontend Development:

Individually, I worked on developing frontend interface using StreamLit to create intuitive and interactive user experience. The interface (taking inspiration from ChatGPT) allows users to upload a video, view progress of processing and visualize the final annotated video output.

Features:

Video Upload: Users can upload the videos in .mp4 format

Annotated Video Playback : The processed video annotated with labels(‘Violence’ or ‘Non violent') is displayed in a red or a green box for violence and non violence respectively on the web interface. The annotated video is saved as well after the processing is completed.

Custom CSS was implemented to enhance the aesthetics, including improved fonts, subtle backgrounds, and responsive elements. Clear instructions are added to guide users on how to use the interface effectively. The frontend extracts and displays video metadata such as duration, resolution, and FPS (frames per second) to ensure user transparency. A loading spinner provides visual feedback during video processing. A sidebar is included in the interface to display the technical details of the backend model, such as architecture, clip length, overlap, and device type.

Challenges:

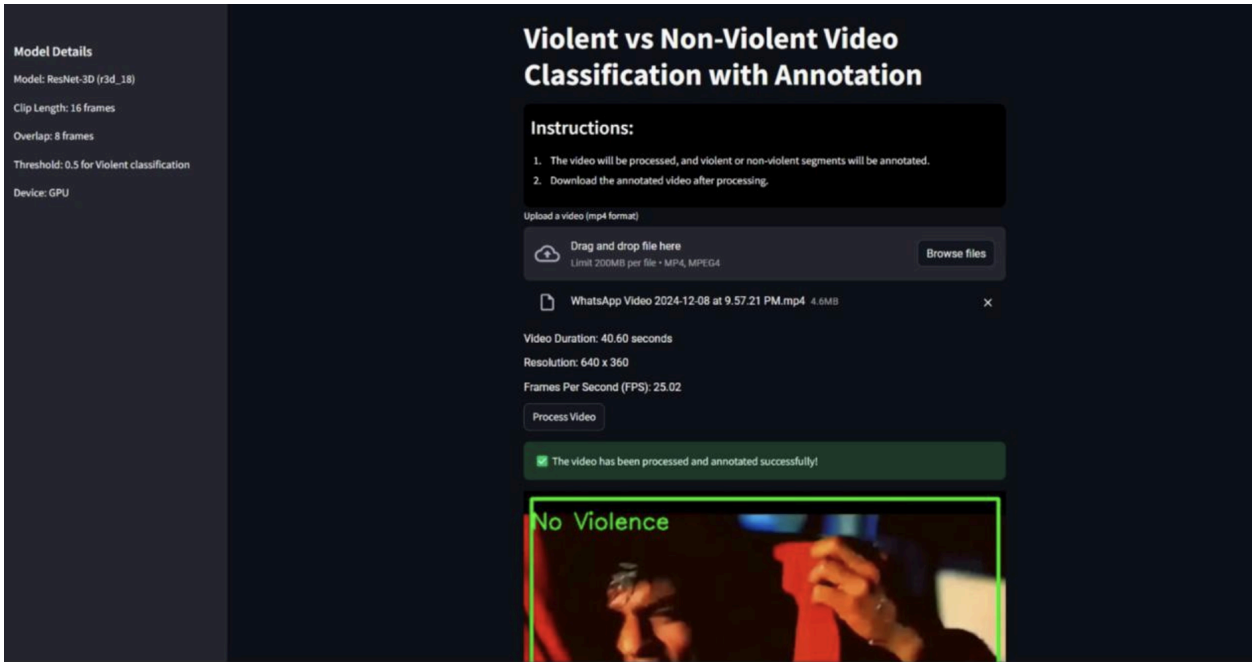
- Integrating the annotating functionality with the frontend while ensuring a seamless user experience
- Optimizing the runtime to minimize delays in processing large videos.

Here’s a sample frontend code:

```
# UI Elements
st.title("Violent vs Non-Violent Video Classification with Annotation")
st.markdown(
   """
   <div style="background-color: #000000; padding: 10px; border-radius: 10px;">
       <h3 style="color: white;">Instructions:</h3>
       <ol style="color: white;">
           <li>The video will be processed, and violent or non-violent segments will
be annotated.</li>
           <li>Download the annotated video after processing.</li>
       </ol>
   </div>
   """,
   unsafe_allow_html=True,
)

# Sidebar for Model Details
st.sidebar.header("Model Details")
st.sidebar.write("Model: ResNet-3D (r3d_18)")
st.sidebar.write("Clip Length: 16 frames")
st.sidebar.write("Overlap: 8 frames")
st.sidebar.write("Threshold: 0.5 for Violent classification")
st.sidebar.write(f"Device: {'GPU' if torch.cuda.is_available() else 'CPU'}")
```

Output:



Backend:

The earlier developed backend was responsible for:

- Extracting frames from videos and batching them for processing— each video was clipped into 16 frames batches and if any batches was “Violent” the video was labeled violent (same for non violence case) and then classifying them as violent or nonviolent based on the prediction. The output would simply be a text “The video is Violent” or “The video is Non Violent

Here’s a sample code:

```
def classify_video(video_path, model_path, batch_size=16):
    model = load_model(model_path)
    frame_batches = extract_frames_from_video(video_path, batch_size=batch_size)
    for batch in frame_batches:
        with torch.no_grad():
            outputs = model(batch)
            probabilities = softmax(outputs, dim=1)
            predictions = torch.argmax(probabilities, dim=1)
        if 1 in predictions:
            return "Violent"
    return "Non-Violent"
```

This backend file is no longer part of the the project for our goals evolved and the team felt the need to enhance the output to produce annotated video , which required integration of inference with video annotation and overlaying of predictions on video frames with clear labels and colors which was handled by the other team members.

Workflow:

Users upload a video via Streamlit Interface
Frames are extracted and preprocessed (resized, normalized)
Inference is performed on frame batches using the ResNet 3D model
Frames are annotated with labels and stitched back into annotated video
The annotated video is displayed to the user on the interface and saved in .mp4 format.

Conclusion

This project developed a system that identifies violent content in videos and provides annotated outputs. My contributions spanned both frontend and backend development, with a focus on user experience and system efficiency.

References

https://docs.streamlit.io/
(Streamlit documentation)