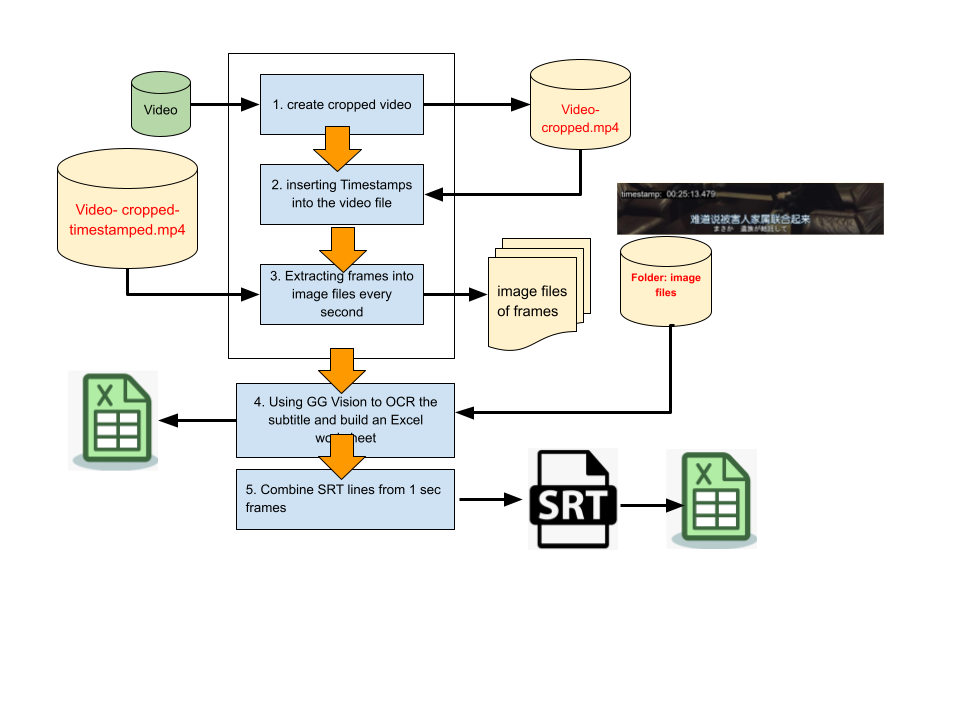
This is a set of FFMPEG commands and Python scripts to extract hardcoded subtitles from any video.

## **The steps are:**

1. FFMPEG: crop the video of the lower portion where the subtitles are
2. FFMPEG: timestamp the video every second for subtitle timing
3. FFMPEG: Extract frames (per second or seconds per frame) for OCR
4. Python: reduce the number of frames to be OCR as much as possible
5. Python: Using GG Vision to OCR the subtitle and build an Excel worksheet
6. Manually inspect and edit the Excel sheet
7. Python: build an SRT file

**Process Flow Diagram**



Here are some utilities that you might find useful:

<https://mediaarea.net/en/MediaInfo> to see media metadata

You must install FFMPEG <https://ffmpeg.org/>

Python will tell you what packages are needed.

## **Step by step**

**Step 1**: FFMPEG Cropping the original video file to a smaller size.

Run python cropnframe.py

This will create movie frames of 1 second per frame in FRAME folder and metadata.txt of file names and video time.

frame\_0001.png, 0.00

frame\_0002.png, 1.00

frame\_0003.png, 2.00

frame\_0004.png, 3.00

## **Step 2: Time Stamping the video**

In this step, the video-cropped.mp4 will be time-stamped and used as the reference time for the subtitle after the frames have been extracted.

Execute this command as is

**ffmpeg -i "D:\Gensub\Duel.of.Arrows.2024.2160p.WEB-DL.DDP2.0.H265-ParkHD.mp4" -vf "crop=1920:280:0:830, drawtext=fontfile='D\\Gensub\\arial.ttf': text='%{e}': x=20: y=40: fontsize=40: fontcolor=white: box=1: boxcolor=black@1.0" -c:a copy -map 0:v -map 0:a "D:\Gensub\XtractHDsub\video-cropped-timestamps.mp4"**

Note: I had to add the arial.tff font file to correct some missing paths in my PC environment. You may not need it.

## **Step 3: Extract frames from video**

**ffmpeg -i "D:\Gensub\XtractHDsub\video-cropped-timestamps.mp4" -vf "fps=1" "D:\Gensub\XtractHDsub\frames\frame\_%04d.png"**

ffmpeg -i "input\_video.mp4" -vf "fps=1" "output\_folder/frame\_%04d.png"

## **Step 4: OCR with GG Vision**

After testing Tesseract against GG Vision OCR, I decided that Tessaract cannot do a good job. Google Vision OCR is far superior. But since there are so many frames, it would be wasteful to do OCR on empty frames. Therefore I wrote a script to use East Text <https://pyimagesearch.com/2018/08/20/opencv-text-detection-east-text-detector/> to preprocess. If a text is detected, then do the GG OCR.

This step creates an Excel file for easy edit.

Run : python createxls.py

## **Step 5: combine SRT lines and create SRT file**

This step combines the Excel SRT lines from the 1 sec per frame images. Creates a new Excel file just in case, and generates an SRT file. The created SRT file will also have a 2 sec minimum duration unless it conflicts with the next start time. From here use Subtitle Edit to translate and fixes problems.