19/2/26

Video Processing - 1 Lab findings

(i) Video Resolution

640 × 480 ⇒ (480 p) SD video
w × h

120 × 720 ⇒ (720p) HD video

1920 × 1080 ⇒ (1080 p) Full HD

2560 × 1440 ⇒ (1440p) 2K Quad HD

3480 × 2610 ⇒ (2610p) ultra HD.

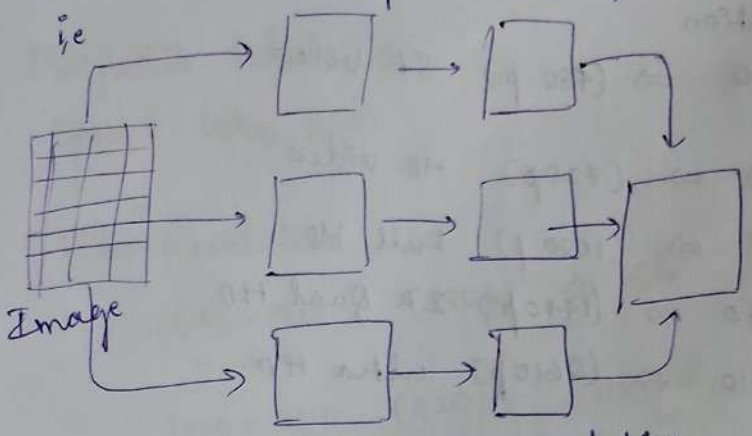
(ii) Frame Rate (Frames per second / Hz)

Steps

* Generated a short video using Chatgpt.
* extracted 15 frames from the video.
* video → series of frames → each frame is an image → each image is an arrangement of pixels.
* we use MobileNet model to try to classify our video but classification is incorrect.
* Why MobileNet doesnt work for our video data ?
  (i) A video is a sequence of frames, not just independent images, so each frame is dependent on the previous frame → this is called temporal dependency.
  (ii) MobileNet was trained on the ImageNet dataset which has static images & object centric labels. Whereas, videos have blurry motion.

frame redundancy, scene transitions etc.

(iii) MobileNet uses depth wise separable convolutions i,e



Depthwise convolution

This makes it weak for motion sensitive features.

Important findings from the experiment

(i) we use openCV to load & extract info from the video openCV processes colors as

BGR (Blue, Green, Red)

Thus, while plotting or using matplotlib we must convert BGR to RGB using openCV's attribute

CV2.COLOR_BGR2RGB

bz matplotlib expects color chanel as RGB.

(ii) Annotations

Video annotation is adding structured labels to video data so machines can understand whats happening over space & time.

(iii) Surveillance (after annotation) applications

* Reduced human monitoring
* edge device optimization
* Smart surveillance
* Traffic & smart city
* Healthcare monitoring.
* autonomous systems.

Extra:

cap = cv2.VideoCapture(Path to vid)
all
gives us the frames

cap.release() fn releases all the stored frames