Feasibility Study on a Video Stream

Purpose: Can computer vision detect hangups on a video ?

Test program by Rich Budek 01/19/2021

BACKGROUND

A feasibility study on a still image extracted from a movie was done in a previous document (study). It showed that indeed from an image, a proper computer vision program could locate an area on an image and evaluate it for left over production material to determine if the product had "hung up" in the tooling.

This app or program builds on that experience and expands it. Now, the app takes a MPEG movie file taken on a standard, nothing fancy Android cellphone with a 1280 x 720 picture size and walks thru the file evaluating for defects. There are many options that can be toggled on or off.

The app begins by looking for motion in a pre-defined location on the image. If motion is detected from the previous frame then no evaluation is done. Once the image is stable, it means that the machine is in the dwell position and sitting still. This frame is then extracted and evaluation is done on it.

The same evaluation function that was used to evaluate still images is used on this extracted frame. However, because the camera was not held in a stationary position, the view point switches from frame to frame. So, a wide enough tolerance was picked for the search box was chosen so that it would work on all of the frames. This was the only manual adjustment done.

After evaluation, if the option to store the still images is turned on, then every frame that has been evaluated is stored in a a unique file name. This basically allows the user to step thru every station and verify that the app worked corrrectly.

Evaluating the motion could be done real-time while the machine is in motion, or have the motion be saved in a MPEG movie file and processed at a later time. This report shows that running under a VM on a i5 processor can accomplish the task. The average time between frames is very near to the required 33ms, a 30fps requires 1/30th of a second between frames. Even when storing the images to the drive, the time is about 1.5 frames. It is felt this could be optimized, but skipping every other frame would still allow the collection of every defect.

title	Mini Twisted Bones
descrip	production run 2020
data file	Top_Only.mp4
size	150 KB
date	07/20/2020
pixel size	1280 x 720
operator	Rich Budek
search box	
UL	11 , 505

LR	735 , 580
motion box	
UL	30 , 307
LR	817 , 504

RESULTS:

These are the timing stats for the movie.

FRAMES:	
frames actual =	1,043
frames estim =	1,043
w/ stations total =	36
w/ stations bad =	6
w/ stations good =	30
TIMES:	
time total =	40.420 secs
time count frames =	0.000 secs
time eval =	40.420 secs
time per frame =	38.8(ms)

Note: For continuous streaming evaluation, on a 30fps stream, the evaluation time needs to be **33.33ms or less** in order to not skip frames. If the time is between 33.33ms and 66.66ms then it means the evaluation will skip every other frame, which is still very much acceptable. The reason is that the image stays stable from 5-7 frames and that is enough time to grab it and analyze it.

BAD IMAGES:

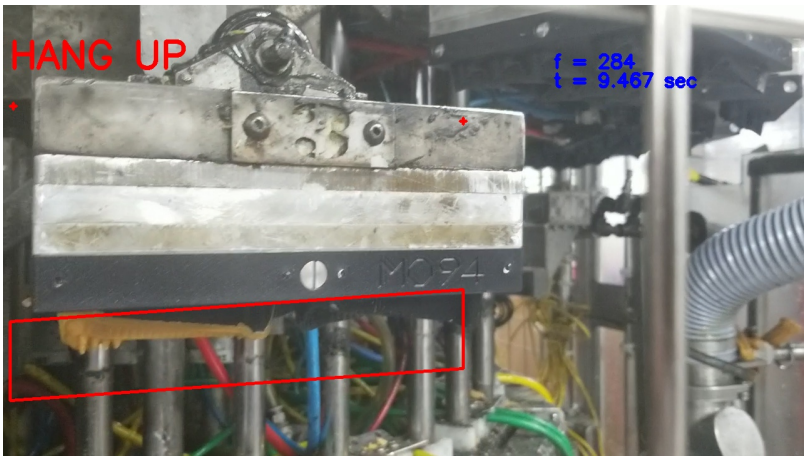
These are the images were a defect was found. Again, they are after motion has stopped.

	image	frame	time	defects	file name
1	10	284	9.467	10238	Still_10.jpg
2	12	342	11.400	14388	Still_12.jpg
3	13	370	12.333	10346	Still_13.jpg
4	17	489	16.300	13263	Still_17.jpg
5	18	515	17.167	10783	Still_18.jpg
6	33	982	32.733	15810	Still_33.jpg

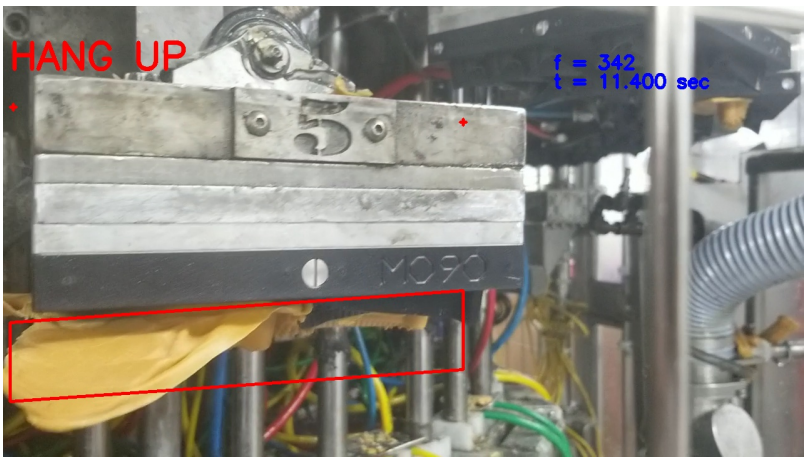
STILL IMAGES WITH DEFECTS FOUND

These are all of the images where defects (product hanging) were found.

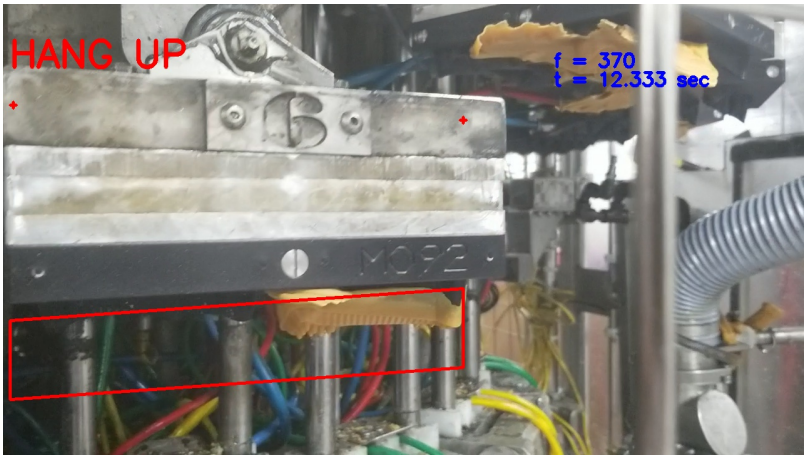
Frame = 284 time = 9.467 secs :



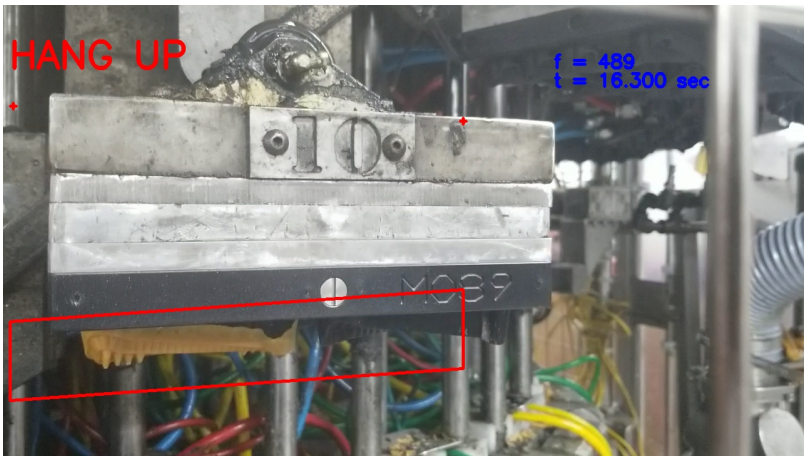
Frame = 342 time = 11.400 secs :



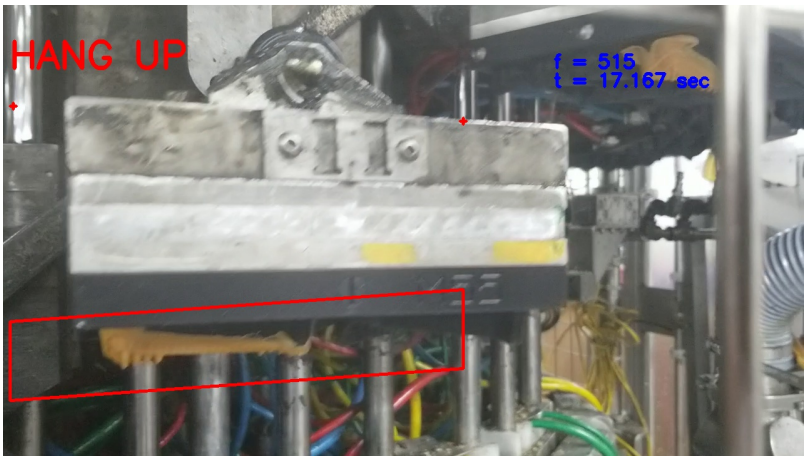
Frame = 370 time = 12.333 secs :



Frame = 489 time = 16.300 secs :



Frame = 515 time = 17.167 secs :



Frame = 982 time = 32.733 secs :