Project Report: Cyclic Repetition

Navlika Singh (B20AI025)
Mitul Agrawal (B20AI021)
Vishnu Kumar (B19BB066)
Kshitij Singh (B19ME039)

---

1) Title: **Cycle Repetition**

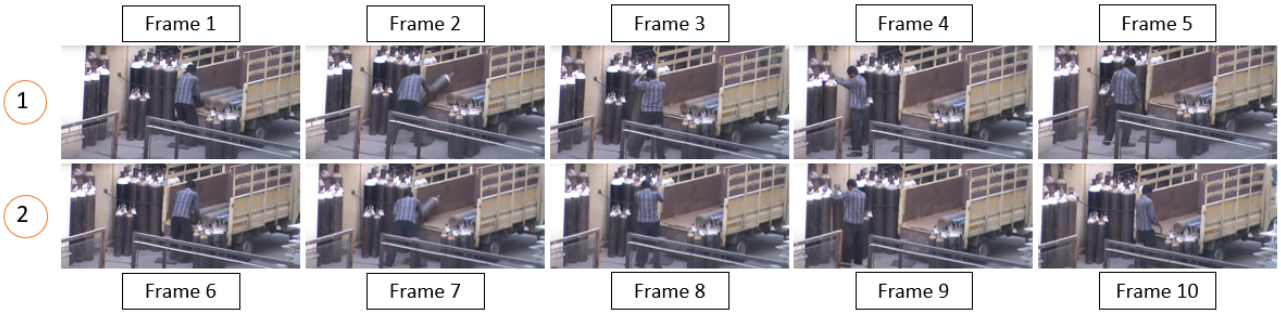
2) Abstract:

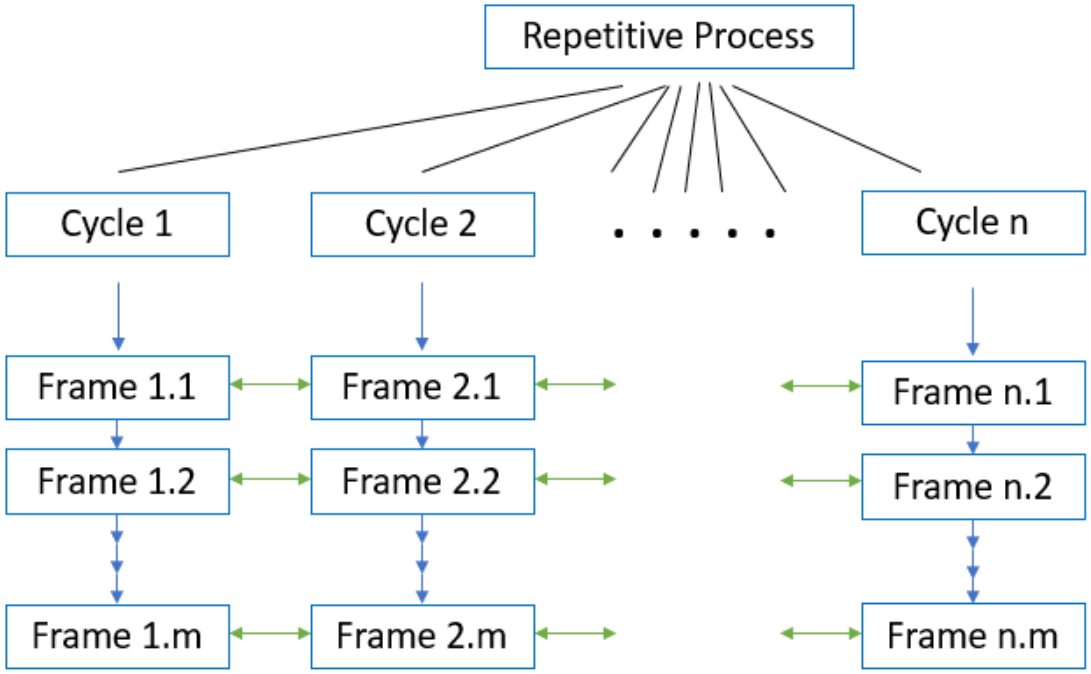
Cyclic processes ranging from industrial procedures, event patterns or day to day activities are of interest because of the variety of insights one can extract out of them. It may be that there is an underlying cause behind something that happens multiple times, or gradual changes in the scenario, or unambiguous action units, semantically meaningful segments that make up an action.
As a consequence the project focuses on classifying and estimating the period of repeating activities. The dataset gathered, consists of labeled videos with content focusing on varying repetitive activities. The team proposes two models to approach the problem, first based on centroid approach and the other on RepNet model developed by Google. We further discuss the innovations that can be implemented on the basic solution and the challenges we may face while implementing the proposed plan. The deliverables of the project are discussed towards the end of the report.

3) Project details:

   a) Description and Motivation:

      The project aims to make a computer vision based model to count the number of times a repetitive activity was performed and to estimate the time period of a single event cycle.
      We want to make a generalized model to count continuous repetitive tasks in the manufacturing line or any other industrial operational process.





The First Industrial Revolution began with the use of steam power. Industry 2.0 began with the use of electricity. Use of Computer Chips and Automated Systems marked the beginning of Industry 3.0.
Industry 4.0 is the era of smart machines and production facilities which can trigger actions without human intervention.

Till now, this process of counting has been done manually which takes human resources. By automating this, time is saved, the process becomes more organized and it is a step towards Industry 4.0.

b) Literature Review for related work:

Cyclic or periodic processes are an inevitable part of our lives. May it be industrial, involving manufacturing merchandise or processing it for shipping, or daily life activities like swinging on a swing or exercising, or may it be more universal like earth rotating on its axis or revolving around the sun. Hence, it is only natural to study such processes and try to draw meaningful insights from them.

'Detection and Recognition of Periodic, Non-rigid Motion' paper published by Ramprasad Polana and Randal C. Nelson in June, 1997. The highlights of the paper are discussed as follows:
Centroid Method -

- Centroid of frame : (xt, yt) = Summation[(i, j)/N] where i,j are pixels in a frame.
- (xt, yt) = (x0, y0)+(u,v)∗t, where (u,v) is the local velocity of the object.
- Just centroid wont work when multiple moving objects are present. Solution : Restrict centroid computation to the area that is most likely to have the object of interest.
- From the position estimates of the past K flow frames, we can get an estimate of the velocity of the object.
- p(t + 1) = (xt + u, yt + v)
- S'(t + 1) = {(x + u, y + v) : (x, y) ∈ S(t)}. [S(t) : Set of pixels to consider for centroid computation].
- (x(t+1), y(t+1)) = w∗p(t + 1) + (1−w)∗c(t + 1) [w is between 0 & 1].

Google developed a model for 'Video Understanding Using Temporal Cycle-Consistency Learning' (article posted on August 8, 2019). This was focused on applying machine learning to understand each frame of a video, that is assessing the interdependency of frames. They proposed a potential solution using a self-supervised learning method called Temporal Cycle Consistency Learning (TCC). This uses correspondences between examples of similar sequential processes to learn representations for fine-grained temporal understanding of video.

Google further developed on this idea, and released an improved model called RepNet, focusing on Counting Repetitions in Videos (article posted on June 22, 2020). The architecture of the model consists of three parts: a frame encoder, an intermediate representation, called a temporal self-similarity matrix, and a period predictor. The frame encoder uses the ResNet architecture as a per-frame model to generate embeddings of each frame of the video, after which the TSM returns a matrix for subsequent modules to analyze for counting repetitions.

Another method is to use div, grad and curl for repetition counting. It is a physics based approach in which a force field is generated for all motions in a given video. Then displacement is calculated for all the vectors. And we get a term which can be differentiated, with this term we can calculate div, grad and curl for the moving object. These derivatives can give a sense of 3 types of motion and prolonged such motions give 9 types of motion which can be extended to 18 types when moving from 2D to 3D viewing angle. Then a continuous frequency wavelet spectra is used which will plot the motions on a time scale. And we check for the graph which strongly represents motion, this will predict the actual nature of motion and with the time scale we can predict the count of reparations.

Video doing motions

With motion displacement takes place

With displacements we can calculate vectors of a point

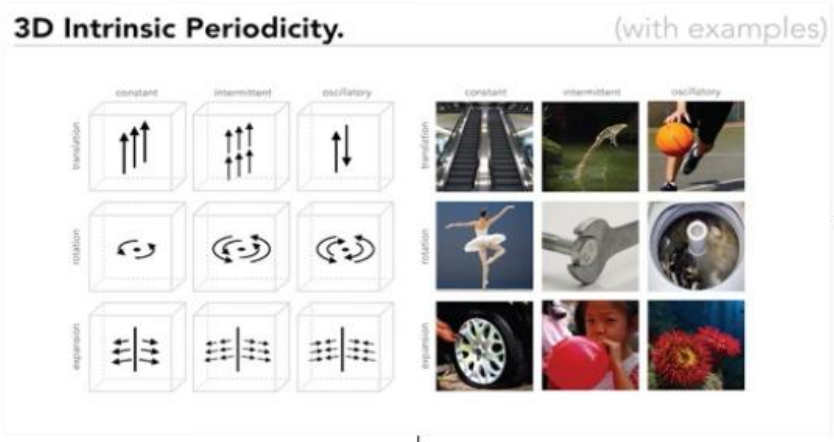
Similarly combining different vectors will give a force field.

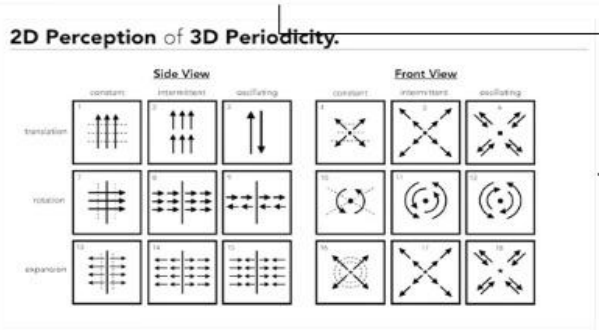
Gradient is ΔF

Divergence is Δ.F

Curl is Δ*F

These three differentiations give different motions.
eg., Translations, Rotation, Oscillatory, etc.





Algo

Motion segmentation to detect motion

Pass the motion through these functions

Use continous frequency wavelet spectra to plot the motions on time scale

Check the graph which represent strongest response

This can predict what is the nature of the motion and also its direction.

With the time scale count the repetations.

c) Technical plan:

i) Basic Solution Hypothesis:

Dataset

The dataset will consist of videos. The content of the video will consist of varying periodic activities. The video must also be labeled for classification purposes. To make the dataset robust the videos captures may follow the following criteria:

- Videos in a variety of lightning conditions, camera angles and of objects in different orientations.
- The proportion of the target item should be higher in the data set.

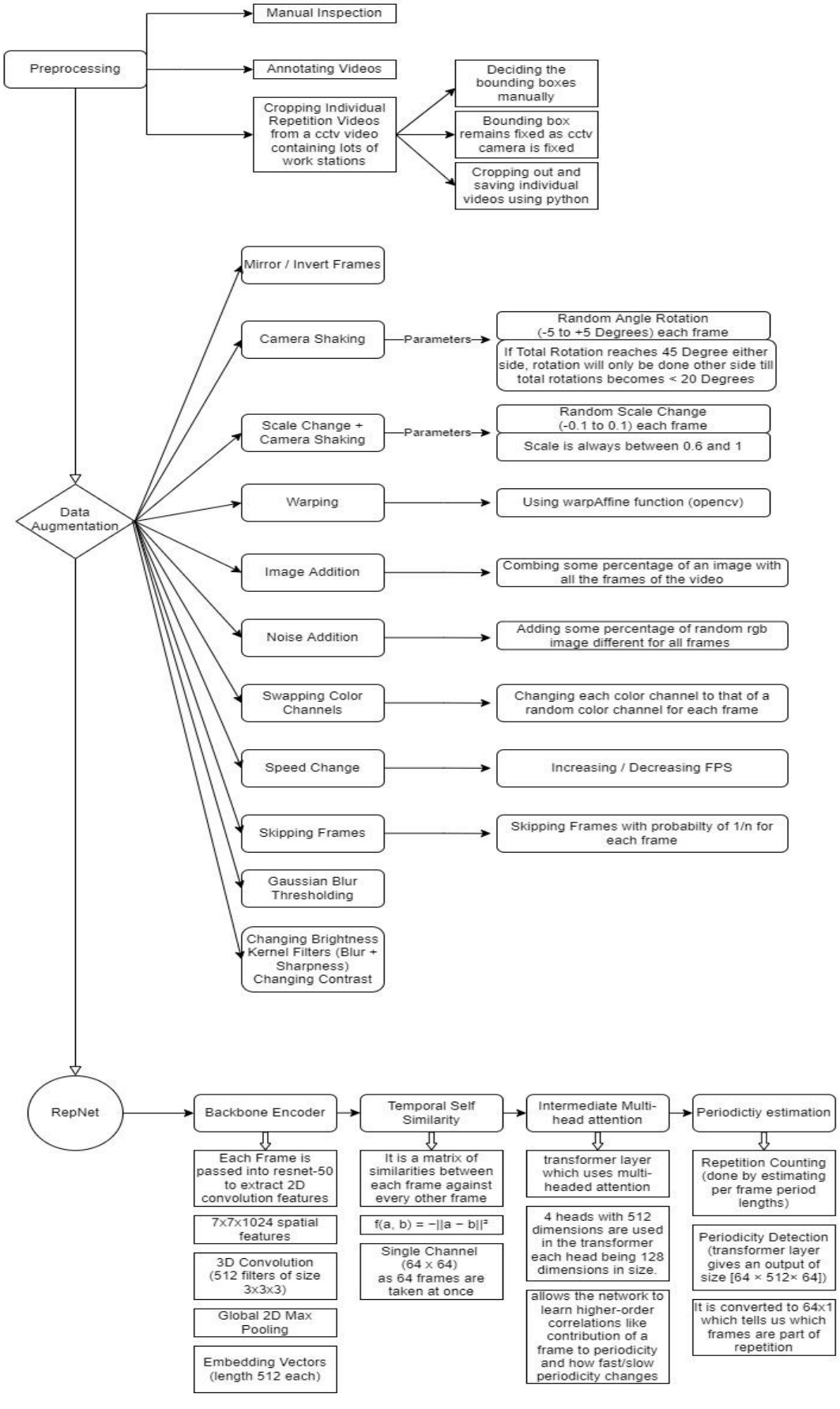
If the dataset is not labeled for classification purposes, label it using one of the many open source tools available for labeling images.

For our needs we also require a top down view of the worker stitching the cloth. This view will help in proper monitoring of the stitching person, this view will also help as other people who can pass from nearby will not hinder our view.

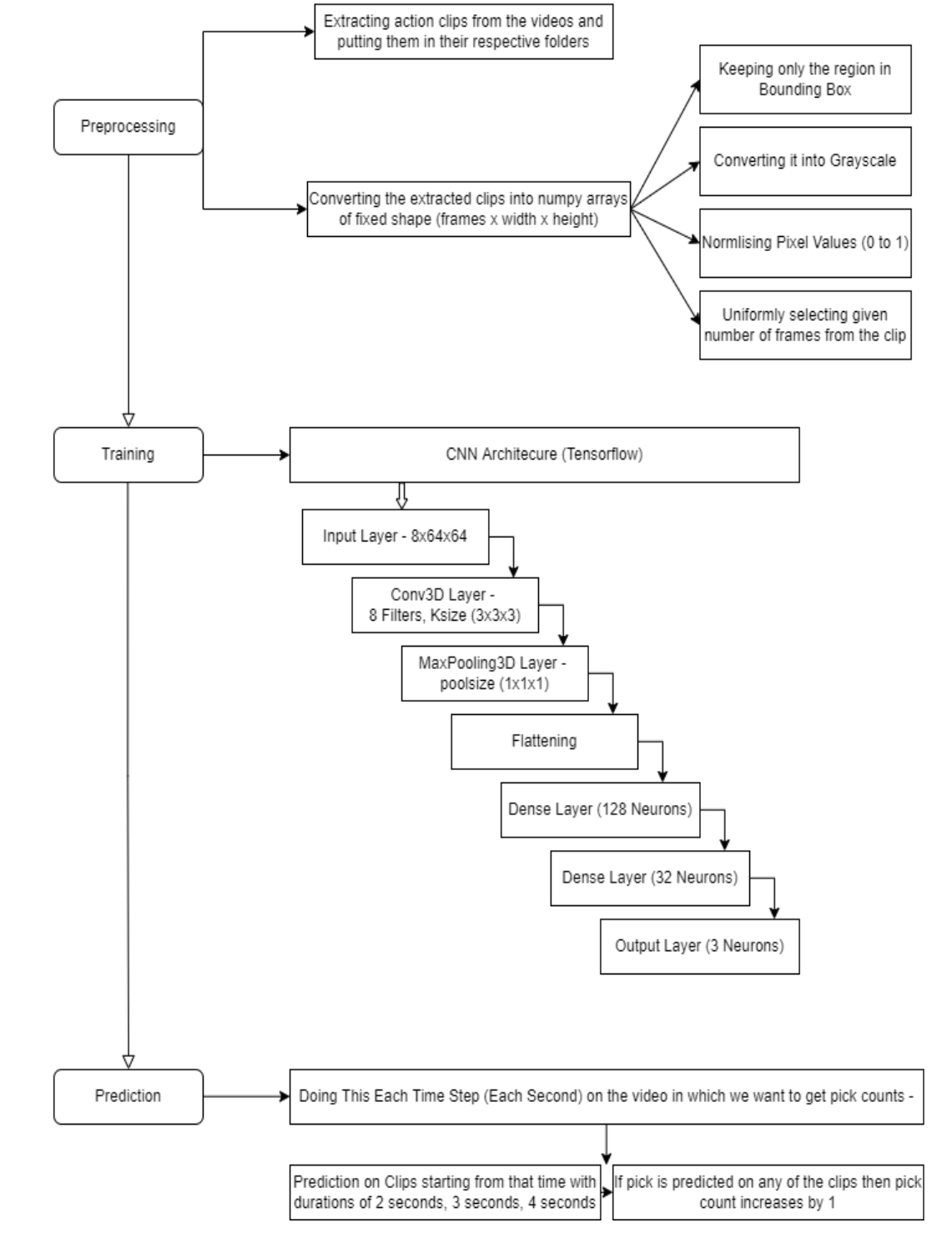
Data Augmentation

Performing data-augmentation techniques to increase the size of the data set. All the data augmentation techniques and other details have been written in the flowchart below:

Repetition Counting



Repetition Counting [3D CNN]



Approach 1: Transfer Learning

Model

The model deployed is RepNet. RepNet is developed by Google, for counting repetitions in Videos. The architecture of the model consists of three parts: a frame encoder, an intermediate representation, called a temporal self-similarity matrix, and a period predictor.

Frame Encoder: It uses the ResNet architecture as a per-frame model to generate embeddings of each frame of the video.

Temporal self-similarity matrix (TSM): This matrix is calculated by comparing the frame's embedding with every other frame in the video, returning a matrix that is easy for subsequent modules to analyze for counting repetitions.

Period Predictor: For each frame, the Transformers are used to predict the periodicity, that is whether or not a frame is part of the periodic process and the respective period of repetition, directly from the sequence of similarities in the TSM. Then it obtains the per frame count by dividing the number of frames captured in a periodic segment by the period length. Summing this gives the number of repetitions in the video.

Approach 2: Average pixel intensity spikes

This refers to taking the average pixel intensity values of a spatial region in the video and plotting it with respect to time till the end of the video to observe the change of a region's average pixel value (from 0-255) with time.

Here we observed that after every stitching cycle was complete, the workers on all the 3 stations (left, center and right table worker) would throw their completed cloth on a table on the other side. We thought that if we can identify when these throws were made, we can count the number of times each worker throws a cloth on the other side which can help us with our initial objective of counting. So for this task we made a rectangular bounding box on all the 3 regions of the video where this throwing was happening (corresponding to the 3 tables in the front camera view). Next we calculated the average pixel intensity values of the regions. Since the cloth being stitched was all white, the average pixel intensity value whenever this cloth was being thrown was very high (closer to 225 compared to normally) as the bounding box while the cloth was being thrown now had almost all the pixels as white or near white due to the cloth occupying the entire bounding box compared to normally when the bounding box contained the background which was more close to darker shades of gray courtesy of the metallic tables. The resulting plot

was a graph of pixel intensity vs time showing clearly where the peaks were. We observed that there was a clear peak after every time interval t which was not fixed due to human inconsistencies (worker) but can be worked with if we set a threshold as the gap between 2 peaks was sufficient. But the problem with this approach was that when we tried to specify the bounding box in the region where this throwing action is happening at each table, it also encompassed part of the region where a lot of the part of the cloth being stitched also came in because the cloth needed to be flipped many times during the entire stitching cycle, and this method hence could not differentiate between when the cloth was being flipped and when it was being thrown as peaks were obtained for both flipping and throwing. Many times though, the peak obtained for throwing was higher than that obtained for flipping, but it was inconsistent and not reliable enough to help the program decide when there was throwing and when flipping which resulted in wrong numbers.

Approach 3: Action Classifier

Action classifier refers to an AI model that can tell what is the physical action that is happening in a video or in particular frames of videos. It can also predict multiple actions happening in the same video.

We observed that after each stitching cycle was getting completed, the workers at each of the 3 statons visible in the front view camera angle (the left, center and right table worker) threw the stitched up cloth to the table across them on the other side in the collecting area. So we thought that if we can identify this throwing action in the videos by the workers, then we can also count the number of times they throw the completed cloth pieces on the other side which will help us with our initial objective of counting. The problem with this approach was that many times it was not able to correctly differentiate between throwing of the completed cloth and the worker flipping the cloth in the stitching process, as both of these processes were very similar looking.

So we decided to focus on another area in the video where the workers were picking the clothes for stitching on the main cloth. We observed that for every piece of main cloth, the workers picked 2 clothes from the bundle kept in front of them. If we can detect properly how many times the worker is picking the cloth from the bundle, we can get the number of times a complete stitching process is happening, which will be half of the number of times the picking process has happened. We decided to train an action classifier that can detect 3 types of action: picking, normal stitching, and person (this class will detect if a person is walking across the region in which the cloth is being stitched). For all the 3 stations in the front view, we obtained 3-6 second clips of all these 3 classes and then

used our standard preprocessing and augmentation steps that we have mentioned before to obtain a larger and cleaner dataset which we then trained the model on. The model so obtained was able to detect some of the picking actions happening but oftentimes confused between person and picking.

ii) Proposed Innovations over the Basic Solution:

Some of the proposed innovations on the basic model are as follows:

- Remove motionless background (set rgb=0,0,0)
- Think of rgb values as respective masses and find the center of mass coordinates of r,g,b. Also the average mass can be found out for r,g,b. And this center of mass and average mass should approximately perform repetition too.
- Further Extending the Idea : Center,Average gives (2x3)+(3) = 9 values. Now the difference of these values between previous and next frame can be taken (local velocity) giving us 27 values. More such values can be taken.

iii) Benefit to the user agency:

There are multiple ways in which the user agency can benefit from this:

- Repeating processes are of interest to researchers for the variety of insights that can be obtained from them. It may be that there is an underlying cause behind something that happens multiple times, or there may be gradual changes in a scene that may be useful for understanding.
- Analyzing these repeating processes may provide us with unambiguous but semantically meaningful segments that make up an action.
- These units may be indicative of more complex activity and may allow us to analyze more such actions automatically at a finer time-scale without having a person annotate these units.
- Perceptual systems that aim to observe and understand our world for an extended period of time will benefit from a system that understands general repetitions.
- In the field of heavy machinery maintenance, it can be used to count the number of cycles that a particular machine part goes through and analyze thus if it is working perfectly or not.
- In manufacturing lines, it can be used to count the number of times all the substeps of a large manufacturing assembly occur and then use it to find if there is any wastage of

parts/resources and whether any process can be optimized by comparing multiple different repeating approaches to solving the same problem and finding which one leads to more repetitions in the given time frame.

d) Experimental Plan:

i) Examples and Use Cases:

Counting reparations of one's exercise set.
Counting the No. of unloading or loading items in any factory.
Counting biological repetitive processes say heart beats.

ii) Benchmarking the Technical Plan with Industry Use Cases:

We will have to achieve a high accuracy.
We will need to observe if there is a change in speed of any repetitions. If some counts are longer or shorter.
We will need to also think about longer reparations, say if something takes more than a day or for very shorter reparations such as something happening in a second or maybe a fraction of seconds.

iii) Challenges faced:

Different activities take different time to repeat themselves. And in some activities the time for one repetition may vary from another repetition. Different activities also makes the problem tougher as if in a person doing any exercise video vs a bouncing ball video. We will also have to maximize our accuracy with respect to real world deployment. The action classifier method still has some difficulties to differentiate between some of the processes happening. There were a lot of interferences by workers passing in front of the camera and through the region where stitching was happening for which we had to make several workarounds to help avoid these interferences. The workers being humans most often did not finish the entire cycle of cloth stitching in the same time duration which was also a very challenging aspect.

e) References:

- https://openaccess.thecvf.com/content_CVPR_2020/papers/Dwibedi_Counting_Out_Time_Class_Agnostic_Video_Repetition_Counting_in_the_CVPR_2020_paper.pdf

- https://www.youtube.com/watch?v=qSArFEIoSbo
- https://link.springer.com/article/10.1007/s11263-019-01194-0
- https://link.springer.com/article/10.1023/A:1007975200487
- https://ai.googleblog.com/2020/06/repnet-counting-repetitions-in-videos.html
- https://ai.googleblog.com/2019/08/video-understanding-using-temporal.html