Remove IT!

Shivan Bhatt

I. INTRODUCTION

With the advent of object removing technology almost a decade earlier that can remove any specified object from an image and fill its place with appropriate background, the next natural step would be to advance this to remove any unwanted moving object in the foreground that clutters the subject of the video, the background. This project takes in a sequence of images with the same frame of reference and produce frames with only the extracted background without knowing the ground-truth background modern machine learning methods.

One obvious method to achieve the desired result would be to photoshop single frames and remove the weed object from each one of them and then stitch together the frames. The problem with this naïve approach is naïve too. Its simply not feasible to do this when there are large number of frames present and is extremely exhaustive and work heavy. Even after the painstaking work of editing every single frame the result may not be what was desired. Since the frames were retouched independently, individual frames would appear fine but the assembly of them would contain clear visual artifacts and not look realistic.

Another method that utilizes other frames also is Median Stack filter (MSF). It stacks all images on top of each other and takes the median value of the pixels of each image for the output image at respective locations. This is a simple but effective method that works in many cases but requires fairly large number of similar images and does not work well if the object is moving slow or there are some parts of the background that are never revealed.

We design and implement unsupervised learning methods and an advanced machine learning algorithm to achieve the mentioned task and compare with the MSF method.

II. UNSUPERVISED METHODS

K-means clustering technique can be applied to group together pixels from the image sequence that belong to the background. Using K-means we try three approaches,

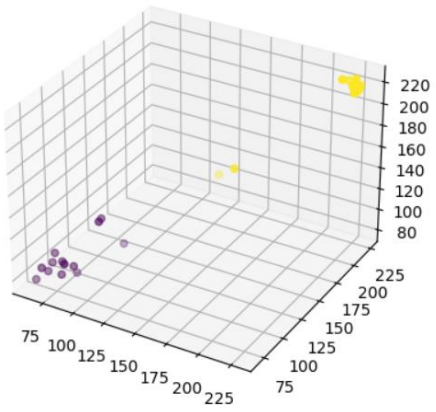

Fig. 1: K-Means clustering applied to a set of RGB values of same pixel position for 40 images

(i) Most Popular Cluster (MPC) selection, (ii)Lowest Variance Cluster (LVC) selection, (iii)LVC selection with inpainting. In the array of input images each pixel has certain RGB values and through K-means we differentiate RGB values for the background and the moving foreground object. Fig 2. shows the result of K-means applied to imageset.

(i) *Most Popular Cluster Selection*

Using K-means with 2 centroids we group together the RGB values of each pixel into clusters. We then choose the densest cluster, take it as our background and then create the output image using that cluster. This depends on the frequency of RGB values.

(ii) *Lowest Variance Cluster Selection*

This model selects the cluster with the lowest variance. It uses the intuition that RGB values of a pixel that constitutes the moving object will have a high variance. Both MPC and LVC when applied would produce similar clusters but might choose different.

(*iii*) *LVC Selection with Inpainting*

Another method would be to apply inpainting to the LVC output. This utilizes a mask that highlights the problem areas that needs further attention and then uses inpainters like Navier-Stokes inpainter or Telea to fill the RGB values at the problem spots.

III. PIPELINE ALGORITHM

The unsupervised methods work well but fails with slow moving objects where the background might not be visible in any frame and also they don't regard the context and only go by the pixels at each particular position. This can be overcome by employing a combination of unsupervised and supervised learning algorithms that considers the context of the images as well. The pipeline has 3 principal components: segmentation to identify and extract a foreground mask, filtering out the extracted mask and inpainting to fill the missing pixels.

(i) *Segmentation*

The purpose of segmentation to extract out masks of the moving foreground object from every frame. This introduces a notion of context as it marks the foreground object in the frame. Many segmentation algorithms are available in open domain, here, we use a pretrained tool FgSegNet_v2 which is essentially a CNN architecture to create our masks. This is a state-of-the-art segmentation tool that has a encoder-decoder network and uses a binary cross entropy loss and few

training examples to accurately segmentation masks.

Fig. 2: Left: the mask of the always hidden background. Right: Output of the filter that lacks the hidden background.

(ii) *Filtering*

Filtering is done to create a background for the output image from the extracted frames. Using pixels that are not present in the mask and only in the background, we can apply previous unsupervised techniques to create the final background. This works better than solely unsupervised approach as by creating and using masks we eliminated the foreground object's pixels from participating in the stacking process. But this technique may leave voids in the image if certain part of the background was never exposed in any frame. Fig 2 shows the part of the background never exposed and the output of the filtering process. To fill the gaps, inpainting is employed.

(iii) *Inpainting*

Inpainting is the process used to fill in and recover missing parts of an image. We use it to inpaint the background pixels that were never exposed and were covered by the moving object in each frame. There are many different kinds of inpainters available, here we tried a non-machine learning technique called Navier-Stokes inpainter and also a Deep Learning based inpainter available in open source.

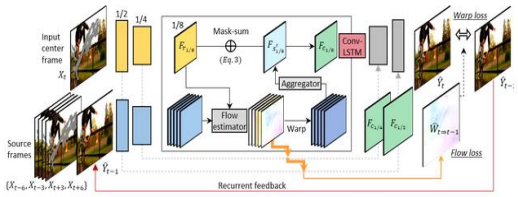

Fig. 3: Network overview of the Deep-Video Inpainter.

Navier-Stokes Inpainting: This uses the navier-stokes equation for incompressible fluids to fill in the missing parts. The procedure first travels along the edges of the region selected to be generated and then continues joining points with same intensity while matching the gradient vectors of the region boundary. Then the rest of the region is filled by ensuring the minimum possible variance in the region.

Deep-Video-Inpainting: This is a deep network architecture built upon an image-based encoder-decoder model that fills spatio-temporal holes with plausible content. It collects and refines information from neighboring frames and synthesize still-unknown regions. Through recurrent feedback and a temporal memory module the output is made to be consistent by making use of features from temporally near and far neighboring frames. Its trained in the DAVIS dataset and presented at the IEEE conference on CVPR, 2019. Fig. 3 shows the network overview for the inpainter.

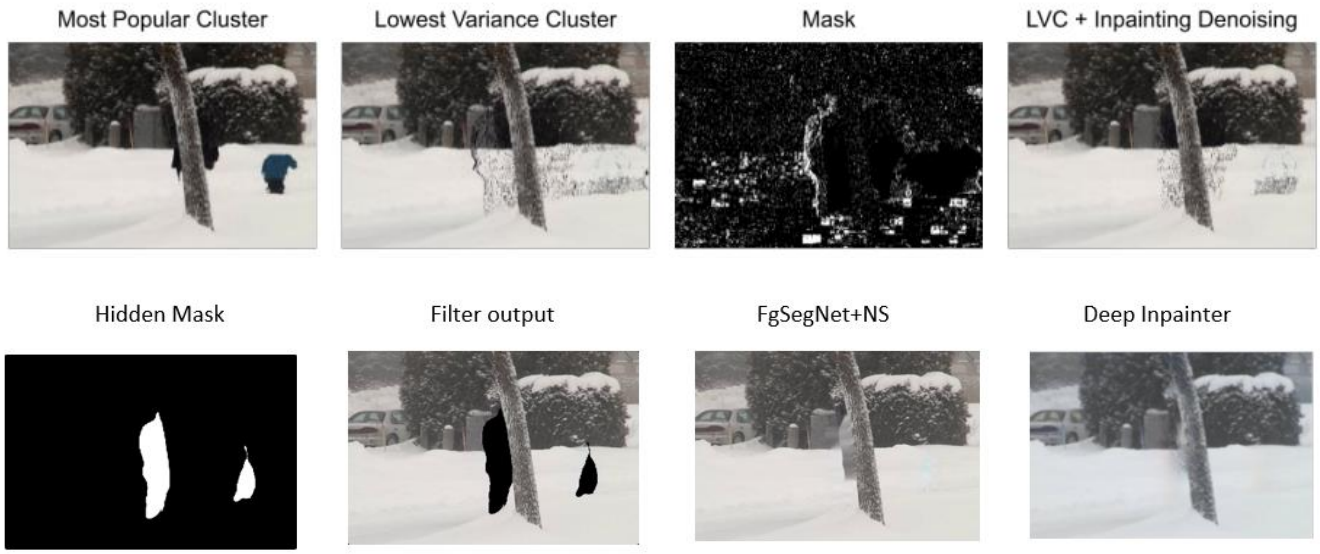

Fig. 4: Upper: results of the unsupervised approach. Lower: results of the pipeline approach. Deep Inpainter yields the best result of all.

IV. RESULTS AND ANALYSIS

The approaches were evaluated on the snow skating image sequence present in the CDNET 2014 dataset. For the Unsupervised approach, Fig. 4: shows the results for MPC, LVC, and LVC with inpainting as well for the pipeline. MPC shows results similar to MSF as both techniques use output metric determined by RGB value frequency at each pixel. The K-means algorithms yield improved results but still somewhat unsatisfactory and has significant noise. LVC with NS inpainting produces slightly better results but still has noticeable artifacts present.

The pipeline algorithm yields drastically improved results which largely depend on the accurate mask extraction from the FgSegNet_v2 which then uses MSF filter and inpainting to complete the output. The Navier-Stokes inpainting still has some visual artifacts which can be seen in the output, but by using the Deep video inpainter the artifacts are more or less negligible. The FgSegNet and Deep inpainter combination produces the most appealing results with the least artifacts and shows significant improvement upon the MSF method which shows that use of machine learning in this domain is a promising approach and further research will most definitely lead to fairly accurate object removal form videos.

V. FUTURE WORK

Object removal from still images is something that has already been achieved and there are various tools for the same both utilizing both ML and non ML based procedures, but removal from consecutive image sequence or videos is something that is still being researched and vast work is going on to accomplish this with the least amount of visual artifacts using deep learning algorithms. One such similar project is Adobe Cloak that was first revealed in 2017 and presented very promising results. It uses a deep learning based autoencoder architecture and hasn't been released to

common public yet. In the project, we demonstrated completely removing an object from a frame using other temporally related neighboring frames in a way that preserves their relation and can be stitched back without any prominent artifacts. Future work on this particular project would be to effectively employ the pipeline algorithm and output a video file from which the moving object has been completely removed. We can tweak the pipeline by using different inpainters and segmentation filters and altering the components order. Some recent work can be found on github which uses this principle with a different flavor.