International Journal of Advanced Technology and Engineering Exploration, Vol 8(84)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2021.874394

Research Article

A Mask-RCNN based object detection and captioning framework for industrial videos

Manasi Namjoshi* **and Khushboo Khurana**
Department of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur, India

Received: 09-July-2021; Revised: 16-November-2021; Accepted: 21-November-2021

Abstract

Video analysis of the surveillance videos is a tiresome and burdenous activity for a human. Automating the task of surveillance video analysis, specifically industrial videos could be very useful for productivity analysis, to assess the availability of raw materials and finished goods, fault detection, report generation, etc. To accomplish this task we have proposed a video captioning and reporting method. In video captioning, we generate summaries in understandable language that comprehend the video. These descriptions are generated by understanding the events and objects present in the video. The method presented in this paper constructs a captioned video summary, comprising of frames and their descriptions. Firstly, the frames are extracted from the video by performing uniform sampling. This reduces the task of video captioning to image captioning. Then, Mask- Region-based Convolutional Neural Network (RCNN) is utilized for detecting the objects like raw materials, products, humans, etc. from the sampled video frames. Further, a template-based sentence generation method is applied to obtain the image captions. Finally, a report is generated outlining the products present, and details relating to the production, like duration of the product being present, the number of products detected, the presence of operator at the workstation, etc. This framework can greatly help in bookkeeping, performing day-wise work-analysis, to keep track of employees working in a labor-intensive industry or factory, performing remote monitoring, etc., thereby reducing the human effort of video analysis. On the object classes for the created dataset, we have obtained an average confidence score of 0.8975, and an average accuracy of 95.62%. Moreover, as the captions are template-based the sentences generated are grammatically and meaningfully correct.

Keywords

Object detection, Mask-RCNN, Video captioning, Video analysis, Image captioning.

1.Introduction

In today's world, automation and automated systems are growing at a tremendous rate. A huge effort is applied for creating systems and programs that eliminate the need for human effort in day-to-day working. The use of robots in factories to assemble machinery [1], to perform various tasks is not new. However, for labor-intensive industries, that create employment for the masses, it is important to generate reports, find the material availability, perform productivity analysis, etc. Moreover, in any business, it is important to optimize the working of its systems for maximum profit based on minimum investment. To achieve this, productivity analysis is an essential step in any business model.

Productivity analysis uses the data collected in various processes in the business to determine areas of improvement in the system. Having a record of such analysis is helpful when trying to expand the business or find out any stages that bring the business down.

It is very common to have surveillance systems installed in the industries due to the reduced cost of these systems. The surveillance videos can be analyzed to obtain a lot of information. This paper presents a technique to perform productivity analysis and report generation by performing video captioning. Natural language sentences are framed, that describe the frames in the videos. These sentences are formed by detecting the objects present in the video frames by performing object detection.

*Author for correspondence

Lot of advancements have happened in object detection techniques [2]. Object detection has always been a topic of interest, and its applications have increased as technology and hardware improves and become more sophisticated [3]. Industries use object detection for face detection [4], and its related topic, detection of anomalies in given sample space [5]. Traffic control is an important sector, with special efforts being made for car license plate recognition [6, 7] and car model recognition [8], as well as pedestrian recognition [9] and traffic light detection [10]. Object detection in medicine is an emerging field, with detection being used to detect illnesses like breast cancer [11] and skin lesions [12].

Object detection techniques can be classified based on whether they are neural network-based or non-neural network-based. A neural network [13] is a collection of nodes that represent a biological neuron. The connections between these nodes are known as weights and use activation functions to perform tasks like modeling and self-learning algorithms. Non-neural network-based detectors include Viola-Jones object detection framework [14], and Scale-Invariant Feature Transform (SIFT) [15]. Neural network-based detection uses deep learning and includes algorithms like You Only Look Once (YOLO) [16], Region-based Convolutional Neural Network (RCNN) [17], Fast-RCNN [18], Faster-RCNN [19], and Mask-RCNN [20].

We have performed object detection to obtain different object classes in some frames of the video, and then used them for caption formation. After the objects are detected, we keep a record of all the classes of objects detected. We create captions as well as create a small report about our findings.

As a case study, we have created a dataset of images and videos from a small-scale industry. Specifically, the videos and images are obtained from a small-scale utensil-making factory. This factory produces several goods and products like tope, containers, lids, etc. The video recording of the employees working in this factory is considered as the input. The system presented in this paper aims to detect all the important products manufactured in this factory, all the machinery that is being used, the human employees interacting with the products and machinery, and some other miscellaneous items. The CCTV recordings are available in high definition, thereby, the system uses robust object detection to handle videos of a lower resolution. Moreover, accuracy is more desired in the system as opposed to the execution speed.

The main industrial analysis aspects that we deal with are described as follows**:**

- To develop a framework for automated analysis of the videos from a small-scale industry. In the case of small-scale industries and home-based factories, automating the process of analysis and bookkeeping can help reduce the need for additional labor and workers assigned for this specific job.
- Work Analysis: It is expected to compute the working hours of various operators working at different machines. Rather than depending on witness testimonies and accounts on their work schedule, it is more reliable to automate the process for finding the working hours and time spent at any given machine.
- Productivity Analysis: To find the number of pieces manufactured at each machine for various products. Automating the task of computing the productivity of a worker or a machine provides additional benefits and reduces human work.

In a small-scale business, this analysis is especially useful, as the suggestions from the analysis are easier to implement and can help change the scale of the industry. Operations like keeping track of the product, the working of the employees, the productivity of the team, all these issues can be improved after the collection of data from the industry and creating a report to discuss the day-to-day working.

The main contributions of the paper are listed below:

- Video captioning framework for Industrial Video analysis.
- Object detection utilizing Mask-RCNN network to detect the different types of products given a factory-based context.
- Creation of template sentences and template-based video frame caption generation utilizing the detected objects.
- Presentation of reports with analysis of the video components.

The remaining portions of the paper is categorized as follows. Section 2 presents the literature review. The mask-RCNN algorithm is discussed in section 3. Methodology is presented in section 4. Results of the system are presented in section 5, followed by discussions in section 6, and the final section 7 being the conclusion.

2.Literature review

In literature review section, we discuss the existing methods of template-based video captioning and object detection methods.

2.1Template-based video captioning

Template-based video captioning [21] is a growing field in which deep learning methods are utilized to provide live commentary on what is happening in the given video. Captions are descriptions of the events of a given piece of media, which help simplify the explanation concerning what is happening in the given media. Predefined templates can be used to ease the generation of captions, where we already assign the role that a certain type of object will play. The exiting techniques of template-based caption generation are performed in two stages. The first stage involves the extraction of visual features from the video, followed by the placement of the features in a template that is predefined and presented in an easy-to-understand language. These features are found by identifying other objects present in the image and their relation with one another. To accomplish this task, object detectors are trained in order to easily perceive different objects and events in the video in question.

The templates used for the placement may be categorized on the basis of case-frame [22, 23] or on basis of attributes [24, 25, 26]. In the first scenario, the features are placed in case-by-case sentences. In the latter technique, the features are placed according to semantic rules, like subject-verb-object rule common in English language. These methods have been show to perform well in situations where we have small domains and a respectable number of constraints.

2.2Object detection

Object detection is an expeditiously growing area of research. It is technology based on computer vision, where machine learning algorithms are used for the detection of certain defined classes of objects from an input image based on a dataset containing information of these classes.

Object detection methods can be abtracated to categorise into two broad methods, namely traditional methods and deep-learning-based methods [27]. Traditional methods rely on visual features, which are known characteristics of a class. For example, the shape of a car is more or less similar, with four wheels and a body. These wheels (circles) and body (rectangle) form the features of a car. Detection of a shape that matches these features thus becomes simpler. Different methods are used to define the features of a class by the traditional methods. Shape, colour, and texture are some of the most basic differentiating features.

Other feature detection methods include Scale-Invariant Feature Transform (SIFT) [15], Speeded Up Robust Features (SURF) [28]. Haar Feature [29] is usually used for detecting faces in given media. The Viola-Jones framework [14] uses Haar features [29] and SIFT features [15] for object detection. Feature detection is followed by the classification of objects into classes. Some of the classifiers for object class detection are Support Vector Machine (SVM) [30], Support-Vector Clustering (SVC) [31], Transductive Support Vector Machines (TSVM) [32], and more.

Deep Learning-based methods use artificial neural networks and multiple layers of computation to extract the needed features from the given input [27]. Domain-specific detectors can be classified as two-stage detectors or multiple stage detectors, and single-stage detectors. You Only Look Once (YOLO) [16] is an example of single stage detector. It utilizes a single neural network to span the complete image, dividing it into different areas, known as regions, and uses weighted probabilities to predict the bounding boxes. Two-stage detectors/multiple-stage detectors include detectors based on Convolutional Neural Networks (CNN), like RCNN [17], Fast-RCNN [18], Faster-RCNN [19], and Mask-RCNN [20]. These are region proposal-based methods, that scan the image and select important regions to focus on, which are known as the region of interest.

RCNN [17], or Regions with CNN features, adds regional proposals to simple convolutional neural networks. It uses a bottom-up approach to localize class segments in any given image. It has three stages- region proposal generation, extraction of features using CNN, and classification and localization of objects. However, RCNN only provides rough localization of proposals, and to achieve the refinement in these proposals, the speed, and accuracy of regions are lost [18].

To solve this problem, Fast-RCNN [18] implements the Spatial Pyramid Pooling Networks (SPPnet) [33] method, which computes a convolutional feature map for the image. Each object proposal is then classified using a feature vector, which is obtained from the feature map. Due to this, the training time is reduced by 100 times. SPPnet also considers feature extraction and training that is similar in working with RCNN. The input image is processed with convolution layers to obtain the maps. At each level, a Region of Interest (RoI) section gives a feature vectir with fixed length. Each feature vector gets used by FC layers, which is followed by branching of given input into two output layers. One output layer produces softmax probabilities, while the other output layer transforms the refined bounding box position coordinates into real numbers. A-Fast-RCNN [34] builds on Fast-RCNN by using adversarial networks to train for object detection in real-life scenarios as the input may be very different from the training images. In the Fast R-CNN, all network layers are trained in one stage with a multi-task loss fucntion. This not only reduces storage, it also improves accuracy and efficiency. While Fast-RCNN networks greatly reduce time when it comes to training and computation, the time it takes to compute regional proposals is still substantial [18].

Faster-RCNN [19] extends Fast-RCNN by adding the functionality of Regional Proposal Networks (RPN) to reduce the time required for proposals. Faster-RCNN uses a deep fully convolutional network (FCN) [35] for regional proposals, and a detector building on Fast-RCNN's for class identification of proposed regions. On one hand, this allows us to train the given samples in an end-to-end fashion, but on the other hand, the lack of object instances makes it difficult to work with boundary cases scenarios and shapes.

Moreover, in Faster- RCNN,pixel-to-pixel prediction poses a problem. To counter this, Mask R-CNN [20] uses a new layer known as RoIAlign, and outputs a pixel-to-pixel mask that maps to the original image. Mask-RCNN [20] adds the functionality of creating pixel-to-pixel object masks of the detected classes. This occurs in parallel to the bounding box and class label detection. This new edition has minimal overhead and is an essential tool for the generalization of detected classes of objects. By adopting this change, the method neatly increases the accuracy of the masks generated even under contraints.

In [20], experiments show that Mask-RCNN works faster and more efficiently on the COCO dataset than any Faster-RCNN and Fast-RCNN. In the survey proposed by Jiao et al. [3], various of these methods were compared on sample COCO dataset 'trainval', and the results showed that out of single-stage detectors, YOLO [16] and SSD [36] achieve high interference and test speed, whereas, for two-stage detectors, Faster-RCNN [19] show high localization and object recognition precision. Mask-RCNN [20], while behind Faster-RCNN, worked much better than any single-stage detector. When comparing Mask-RCNN and Faster-RCNN [3, 27], we find that while both have a similar runtime for COCO dataset, Faster-RCNN did not provide pixel-to-pixel detection provided by Mask-RCNN.

3.Background

3.1Mask-RCNN for object detection

As described in this paper, we have utilized the functionality presented by Mask-RCNN [20] to implement the object detection system. Mask-RCNN builds upon the workings of Faster-RCNN [19] and adds a new feature in the form of a 'mask', which is a pixel-to-pixel estimation of the shape of the object detected. Faster-RCNN is not equipped with this kind of pixel-to-pixel analysis for output. This segmentation mask is output for each Region of Interest (RoI), along with the bounding boxes of detected objects and the classification of the object in the form of class labels.

The Mask-RCNN algorithm works in two stages: Proposal and Prediction as shown in *Figure 1*. The first stage uses Region Proposal Network (RPN), used to generate proposals based on where the object is located in a region. It uses CNN algorithms along with RoIAlign, to generate RoI. Instead of outputting one bounding box, it outputs multiple possible bounding boxes.

Anchor boxes are used for the detection of multiple classes of objects. These are a set of boxes of predefined dimensions and are used to capture the scale and the aspect ratio of classes that are to be detected.

The second stage, Prediction, as shown in *Figure 1*, applies the RCNN detection twice in order to predict the object's class, to refine the bounding box, and to generate a binary mask at the pixel level of the object based on the first stage proposal. We get the result of object detection by eliminating all anchors and all boxes that fall below the confidence scores. The final output image contains detection in form of segmentation and bounding boxes and classes.

The loss of each RoI [20] is defined as Equation 1.

$$L = L_{cls} + L_{box} + L_{mask} \tag{1}$$

In Equation 1, L_{cls} refers to the classification loss and L_{box} is the bounding box loss. These are the same values defined in [18]. Mask-RCNN adds a loss function in the form of L_{mask}, which is the loss function of the binary mask generated. This L_{mask} function allows a pixel-to-pixel mask to be generated for every class detected.

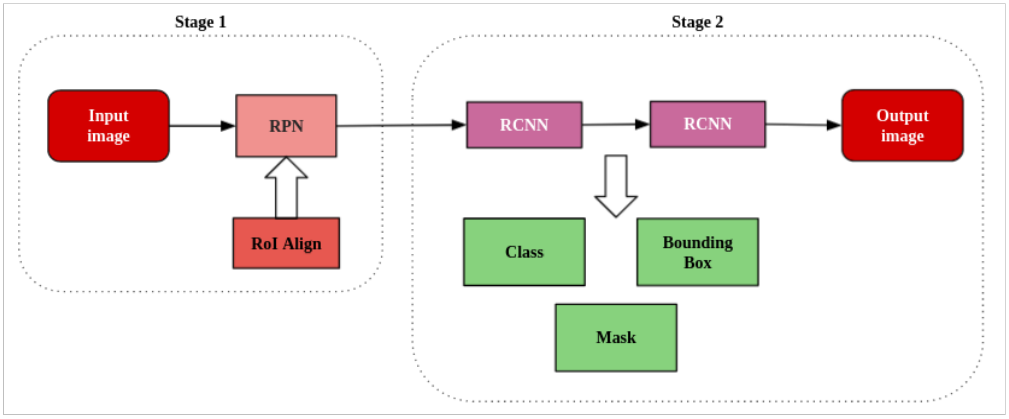


Figure 1 Mask-RCNN architecture

4.Methodology

4.1Dataset

We have compiled the images and videos for our project with the help of a small-scale aluminum utensil manufacturing industry (Pooja Productions, Nagpur). Our custom-created dataset comprises 216 training images with 13 different class selections available for various machines and tools used in the industry. Whereas the test dataset comprises of 180 images belonging to these 13 classes. This dataset of images labeled with classes is used for the object detection algorithm. The object classes are as follows:

person, tope (pot), aluminium plates, Dabba (storage box), scrap (raw material/scrap aluminium), aluminium circles, lid, storage bags, spinning machine, roll machine, press machine, circle cutting machine, furnace.

The images are captured using a normal mobile camera. The detector is trained using a set of training images. The model is trained using pre-trained weights for MS-COCO [37] and Mask-RCNN [38] as the object detection. The dataset is trained for 20 epochs with 100 steps per epoch. To train the system to identify object classes, we annotated all the images in our custom dataset by VGG Image Annotator (VIA) [39]. *Figure 2* shows a sample annotation of an image from the training set, where we have selected our RoI as classes *person* and *roll machine*.

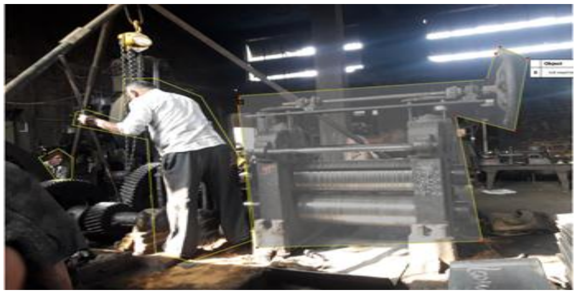


Figure 2 Sample training image with annotations

In the training function of our system, we use this annotated dataset of images as a parameter for training. Other parameters include the learning rate, total number of epochs and all the layers present in the neural network. The detections with confidence scores > 90% confidence are accepted and the corresponding class label is allotted.

The overall system of video captioning utilizes videos from the CCTV camera of the industry. The results are presented on 4 videos from the same environment.

4.2Methodology

We have presented a framework for detecting objects in the video frames and created template-based video captions. A small report is also generated based on the duration of the presence of the object. An overview of our approach is represented by *Figure 3*, and the details of this approach are in *Algorithm 1*.

The framework takes video as input and first performs uniform sampling to exact the video frames. Frames to be selected for detetction are decided according to the duration of the video. The object detection algorithm is applied to each of the sampled frames. For each frame, the class of the detected objects along with the number of occurrences of each class is saved. A few domain-specific sentence templates are pre-built and appropriate objects are filled based on the objects detected. Thereby, a template-based caption is generated for each sampled frame. Consecutively, these image-caption pairs summarize the video, creating a video summary. The saved information about the objects, count, time at which the sampled frame appears, etc. is then used for further analysis and report generation. The details about each phase in the framework are discussed next.

Video frame sampling and extraction

The input is taken in video format (.mp4/.avi). The duration of the video, along with the frame rate of the video is calculated using OpenCV tools [40]. Then, based on the length of the video, the frames are sampled as follows:

- If the video is less than 60 seconds, one frame is extracted per second.
- If the video is less than 60 minutes, three frames are extracted per minute.
- If the video is more than 60 minutes, one frame is extracted per minute.

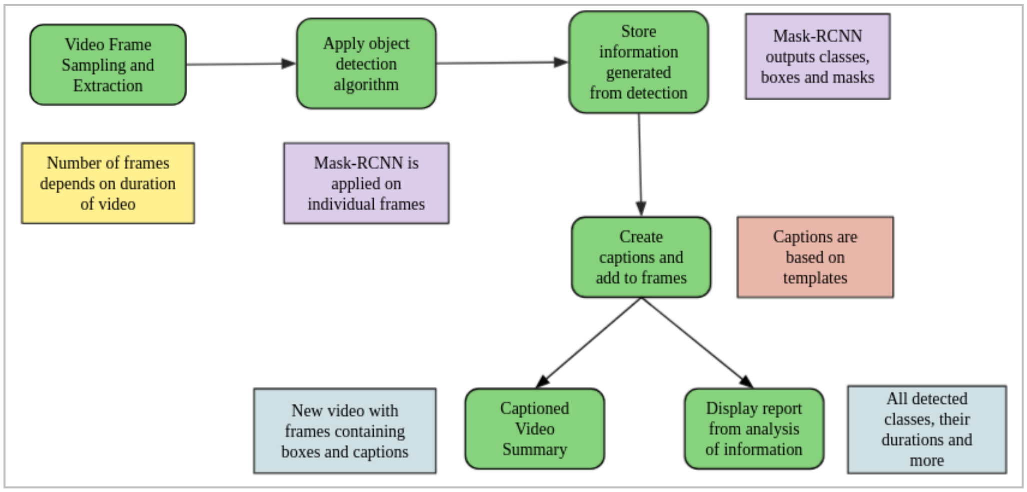

Video Frame Sampling and Extraction
Apply object detection algorithm
Store information generated from detection
Mask-RCNN outputs classes, boxes and masks
Number of frames depends on duration of video
Mask-RCNN is applied on individual frames
Create captions and add to frames
Captions are based on templates
New video with frames containing boxes and captions
Captioned Video Summary
Display report from analysis of information
All detected classes, their durations and more

Figure 3 Overview of proposed framework

```
Algorithm 1
Input: Video $U$
Output: Video $V$ and Report $R$
Step 1:Video Frame Sampling
Obtain video frames from an input video
Let $L$ be the length of $U$
$I = \{I_1, I_2, . .. . , I_N\}$ frames are obtained
$N = f(L)$, where D refers to the total duration of input video

If ($L< 60$) sec then $N = L$
If ($L> 60$ and $L< 3600$) then $N = 3 * L$
If ($L> 3600$) then $N = L / 60$

Step 2: Object detection
Apply Mask-RCNN on each $I_i$ to obtain $O_i$ for each $I_i$
where i = 0,1,. . . , $N$
$O_i$ = Mask-RCNN($I_i$)
The steps are as follows:
2.1 Find RoI for each $I_i$
2.2 Run RCNN algorithms to get output $O_i$
Output $O_i= \{C, B, M\}$
where $C$ is classification, $B$ are the bounding boxes,
$M$ is the mask
Save these frames with
Detection $I' = \{I'_1, I'_2, . .. . , I'_N\}$ and information.

Step 3:Template based Video Captioning
Generate the caption $C_i$ for each $I'_i$ based on classes $Z$
and Templates $T$, where $i=0,1,. . . , N$
$Z = \{human, objects, machine\}$
Step 4: Video generation
For i in $I'$
    $C'_i$ added to $I'_i$
Video $V = I'$ consolidated

Step 5: Report Generation
Using saved information, report $R$ generated.
```

Object detection

Object detection is applied to the sampled frames obtained. We use Mask-RCNN for this purpose. The Mask-RCNN algorithm detects object classes, creates bounding boxes, and binary masks for the detected objects. The obtained classifications are stored in a new file. It contains information about all detected classes for each image. Object detection using Mask-RCNN proceeds as given below:

1. Anchor sorting and filtering

Region Proposal Network (RPN) is utilized to obtain the regions. We also get anchor values for our image. Lower valued anchors are removed.

2. Refinement of Bounding Boxes

The bounding boxes are finalized after all the refinements in this step.

3. Mask Generation

The masks are generated, updated to fir the original dimensions and then arranged in respect to the image in their respective positions.

4. Activations of layers

This is used to detect any unwanted activations, like background or noise.

All these steps are then combined to give the final detected image.

Template-based video captioning

After the objects and their classes are detected. Natural language description is generated for each video frame. The process of generating Natural language descriptions is referred to as captioning. Captioning is the process where ‘captions’, or texts based on the visual media visible on the screen, are generated for each frame of the given visual media to aid a better understanding of the happenings on the screen. In machine learning and object detection, captioning is an extra feature that can be added to the system that remarks upon the classes being depicted in the given image. We can add context to the detected objects, like describing what an object is doing, the connection between different objects, etc. In our approach, the captions are added to the frame. We use sample-defined templates for our captions. These templates take into consideration what the detected objects are. We have defined categories of classes (Z) for our specific application as follows.

human: person
objects: tope, aluminium plates, Dabba, scrap, aluminium circles, lid, storage
machine: spinning machine, roll machine, press machine, circle cutting machine,
furnace

We use these categories to select the most suitable template from our list. Some of the sample templates (T) include

Person is working on {machine}
{human | objects} is/are present
{machine}is present
{objects}is/are being manufactured

Captioned video summary

All the frames with captions and bounding boxes are then consolidated into a new video using the OpenCV tool [40] and saved to the system.

Analysis and report generation

Based on the classes detected, the video created and the original video, we provide information like the duration of all detected objects, in which frame they are present, etc.

Next, we discuss the experimentation and the results obtained.

5. Results

5.1Analysis of various object detection algorithms

To access the best object detection algorithm, we experimented with MS-COCO dataset [37]. This dataset consists of total 164,000 images, with 118,000 training images, 5000 images for validation, and 41,000 images for testing. The dataset is obtained from the MS-COCO repository [37] and implemented on standard codes available for each technique. For MS-COCO dataset the average precision (mAP) values obtained using different object detection algorithms are presented in *Table 1*. The latest deep learning-based algorithms: Fast-RCNN [18], Faster-RCNN [19], Mask-RCNN [20], YOLO [16], and SSD512 [36] are compared. *Table 1* shows us that Mask-RCNN achieved the best performance. However, YOLO [16] algorithm has a better computation time. Faster-RCNN [19] also has a similar precision score as Mask-RCNN [20], however, Mask-RCNN [20] offers better precision as well as an additional result of pixel mask. Thereby, we utilized the Mask-RCNN [20] for object detection.

Table 1 Comparison of various object detection algorithms

Algorithm	**Average precision(mAP)**
Fast-RCNN [9]	19.7
Faster-RCNN [2]	21.9
Mask-RCNN [1]	29.8
YOLO [8]	21.6
SSD512 [16]	28.8

5.2Implementation and results

Training the model for this project is done using NVIDIA DGX-2 Server with 128 GB GPU memory, 256 GB RAM. Whereas, all the experimentations (inferencing) are performed on a system with Intel Processor having 4 GB Ram, with NVIDIA graphics on Ubuntu OS. The model was trained for 20 epochs with 100 steps per epoch.

For object detection, first the Mask-RCNN [13, 20] based model is trained. The training module uses TensorFlow, Keras, and Python3 [38]. For all object instances, a bounding box and mask is generated using Feature Pyramid Network (FPN) [41] along with a ResNet101 [42] backbone.

First, the video frames are obtained as per the method discussed in the previous section. After obtaining the video frames I = {I_1, I_2, ... , I_N}, regional proposals are found and refined for each Ii by anchor sorting and filtering.

Refinement of regional proposals for a sample frame containing aluminium circles is presented in *Figure 4*. *Figure 4(a)* shows the multiple probable regional proposals for our image in question. These are shown as the dotted lines. Anchors are used to further refine the regional proposals, as demonstrated in *Figure 4(b)*. Finally, the final regional proposals after all the refinements are shown in *Figure 4(c).*

Further, after the bounding boxes are refined, the pixel-to-pixel mask is generated for each detection. These generated and scaled masks are placed on the image in their respective positions. (Refer to *Figure 5*).

Furthermore, layer activation is performed to detect any unwanted activations, like background or noise. *Figure 6* presents the layer activations.

Finally, the bounding boxes are created and labels are assigned to each detection. This final result is then stored along with the number of objects detected, and their labels for each frame. The objects with bounding boxes are obtained as shown in *Figure 7*. It is evident that the aluminium circles are detected with good confidence scores.

After the objects are detected for each frame, template-based captions are generated. These captions describe the objects in the frame.

This is done by storing the details of the detected objects in a file, which is utilized for template-based video caption generation. This caption is then displayed on the frame and all frames are used to form a summary video. The summary video contains frames with captions. *Figure 8(a)* shows the text file obtained for video video_1.mp4.The file contains the detected objects in each of the sampled frames of this video. For example, frame 1 has the presence of two objects- person and tope. Similarly, frame 9 has two persons, aluminium plates and tope. *Figure 8 (b)* presents the detected object and caption for the first frame and the generated report is shown in *Figure 8* (c). The presence of an object is presented in the report. The duration for which the operator (person) is available in the video can assist in the analysis of the working hours of the operator. The presence of aluminium plates confirms the availability of raw material. Such reports in the real-time system can be used for alarm or alert generation in case of raw material is about to exhaust or if the operator is absent for a long duration.

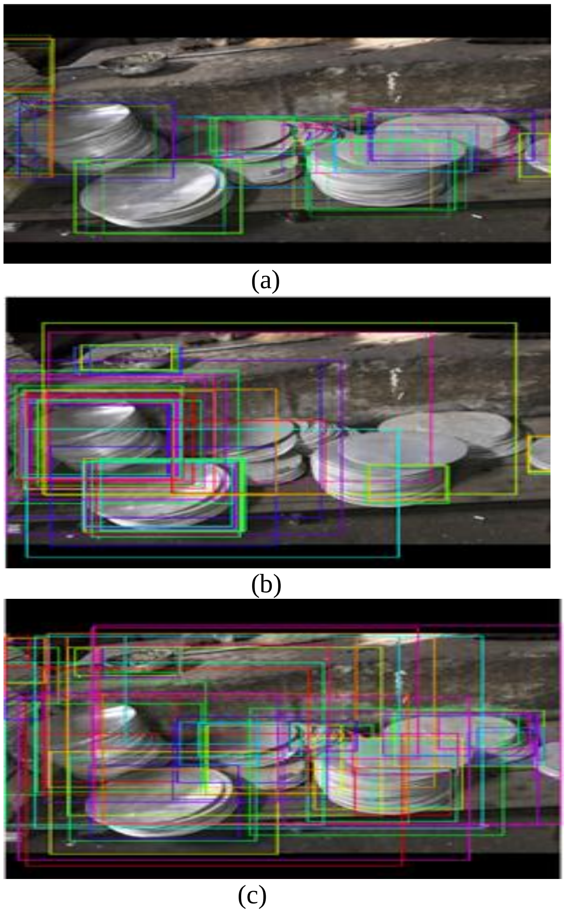

(a)
(b)
(c)

Figure 4 (a) Regional proposals (b) Multiple regional proposals on further refinement (c) Final refined regional proposals

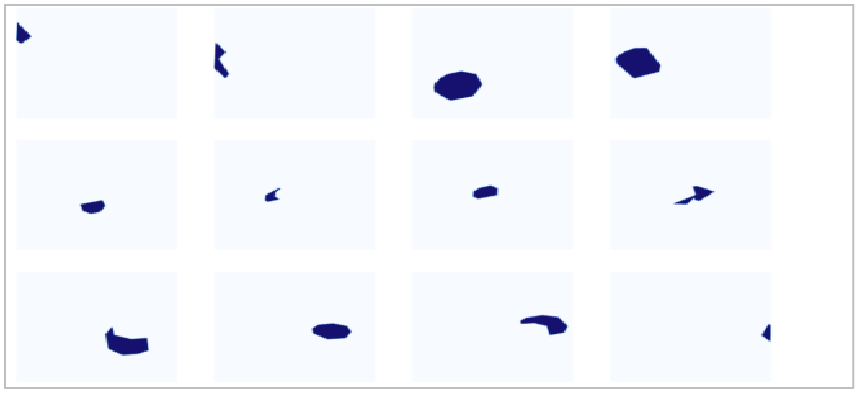


Figure 5 Generated masks

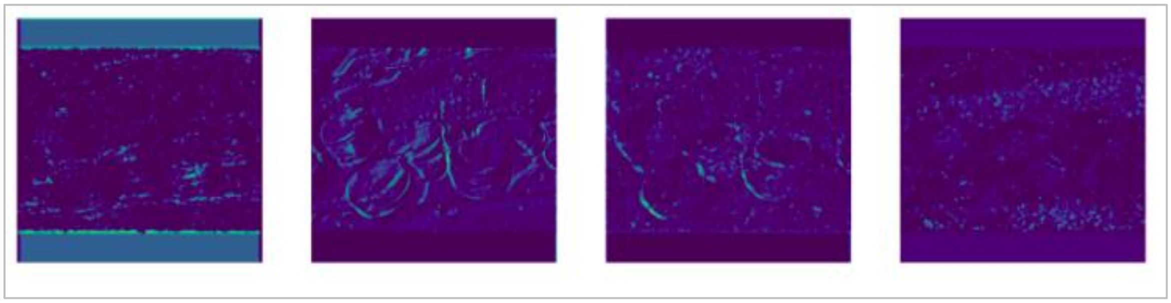


Figure 6 Different activation layers

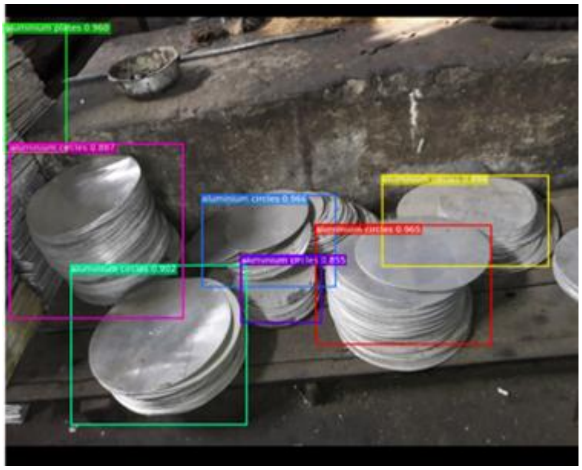
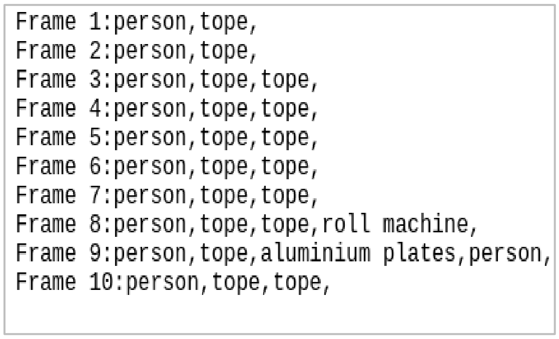


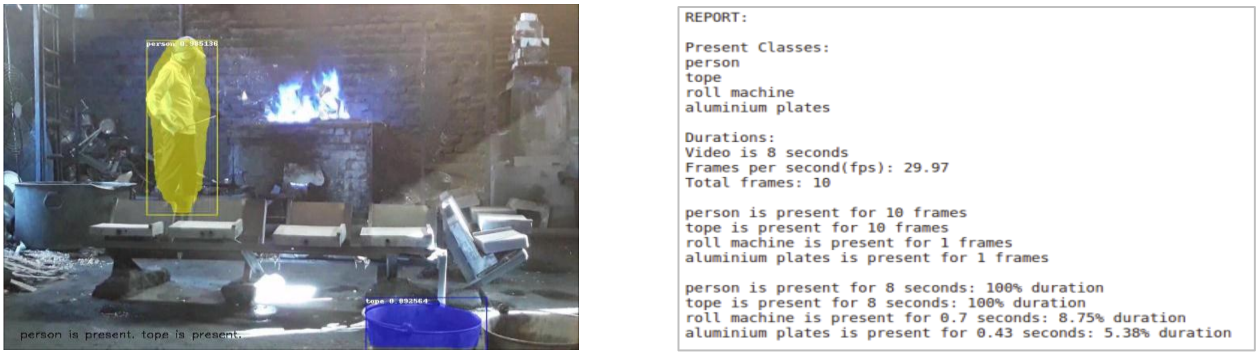
Figure 7 Final detected boxes

Table 2 shows the confidence scores of classes detected in the video input video_1.mp4. For each object that is detected, the confidence scores given by the Mask-RCNN algorithm are presented. As observed all the detected objects have a confidence score above 0.8. Further, we present the accuracy for the detections. *Table 3* presents the accuracy of the given detections. This is computed by comparing the actual number of objects to be detected in the video and the average number of objects detected by our algorithm. The average is calculated by the formula: the total objects detected divided by frame generated. Thereby, the accuracy is reported in terms of percentage. We have evaluated the time required for execution and presented it in *Table 4*. The details of computation and total time required to analyze videos of different durations are tabulated. We compare the total duration of each video, the number of frames that it generates, and the computation time required by our algorithm.



```
Frame 1:person,tope,
Frame 2:person,tope,
Frame 3:person,tope,tope,
Frame 4:person,tope,tope,
Frame 5:person,tope,tope,
Frame 6:person,tope,tope,
Frame 7:person,tope,tope,
Frame 8:person,tope,tope,roll machine,
Frame 9:person,tope,aluminium plates,person,
Frame 10:person,tope,tope,
```

(a)



```
REPORT:

Present Classes:
person
tope
roll machine
aluminium plates

Durations:
Video is 8 seconds
Frames per second(fps): 29.97
Total frames: 10

person is present for 10 frames
tope is present for 10 frames
roll machine is present for 1 frames
aluminium plates is present for 1 frames

person is present for 8 seconds: 100% duration
tope is present for 8 seconds: 100% duration
roll machine is present for 0.7 seconds: 8.75% duration
aluminium plates is present for 0.43 seconds: 5.38% duration
```

(b) (c)

Figure 8 Results on video video_1.mp4 (a) Text File with information (b) Detected objects in one frame with captions (c) Sample report generated

Table 2 Confidence scores for detected classes in video_1.mp4

Frame No.	Class detected	Confidence scores
1	person	0.985136
	tope	0.892564
2	person	0.983757
	tope	0.926952
3	person	0.981102
	tope	0.890004
	tope	0.879402
4	person	0.991032
	tope	0.911131
	tope	0.900311
5	person	0.958930
	tope	0.888139
	tope	0.890142
6	person	0.990810
	tope	0.933614
	tope	0.895523
7	person	0.995183
	tope	0.819889
	tope	0.799352
8	person	0.986291
	tope	0.708169
	tope	0.892716
	roll machine	0.726274
9	person	0.987159
	tope	0.901975
	aluminium plates	0.881253
	person	0.788502
10	person	0.985724
	tope	0.842362
	tope	0.713862

Table 3 Accuracy of each frame with respect to detected objects

Video Name	Actual number of objects present in video	Average number of objects detected from video	Accuracy (%)
video_1.mp4	3	3	100
video_2.mp4	7	6.3	90
video_3.mp4	2	1.9	95
video_4.mp4	4	4.1	97.5

Table 4 Duration of video with computation time

Video Name	Input video duration(s)	frames	Computation Time(s)
video_1.mp4	8	10	110
video_2.mp4	20	24	240
video_3.mp4	30	36	370
video_4.mp4	40	50	425

6. Discussion

Based on the report generated from the video input show, we can obtain the duration of the appearance of each detected object. We can use this information to analyze the working of the factory, the products being manufactured, the working of the employees and much more. The method is scalable and can accommodate the addition of new objects to the dataset. We have applied the method to domain-specific videos. However, it can be applied to any domain by the creation of a dataset, as we have already shown in preceding section.

We also experimented with various object detection algorithms on a large dataset- MS COCO to confirm the use of Mask-RCNN as the object detector. As it performs the best amongst the different methods, it gives good accuracy and confidence score for the images from the industrial domain. Moreover, as we have used template-based captioning, we can add more sentence templates to accommodate scalability. The advantage of template-based captioning is that these methods are good for domain-specific video captioning and generate sentences that are grammatically correct as opposed to other methods.

Although the results are encouraging, there is a scope of improvement as the input video time increases. As observed in *Table 2*, as the time of the input video increases, the computation time also increases. For longer videos, the computation time goes up in minutes. Thereby, when the CCTV video of the entire day needs to be analyzed, the computation time is expected to be high. This is accepted if an offline analysis is required. However, for real-time analysis, more sophisticated methods need to be adapted. A complete list of abbreviations is shown in *Appendix I.*

7. Conclusion and future work

In this paper, we have used Mask-RCNN for object detection from a video in a factory-based situation. First, the video frames are selected from our original input video, and then we assign different classes to detected objects in each of the frames. These objects are utilized during template-based caption generation. After obtaining the output, we have created a new video and generated a report. The output video includes frames with captions, which are selected from templates. The report contains information like the duration of the video, how many frames were selected, the duration of each detected class instance, and more.

The presented method gives us the adaptability and scalability that different situations may require. It can be used for productivity analysis of any small-scale industry or manufacturing unit, as we can train and create our custom dataset based on the context of the industry in question. The generated reports can be used to perform question-answering on the video data. The method can be further improved by increasing the dataset. In the future, we will work on longer-duration videos and improve the technique to handle longer-duration videos.

Acknowledgment

The authors are grateful to the owners and employees of Pooja Productions, Nagpur for their support in collecting the required data for this paper. We would also like to thank the AI and ML Centre of Excellence, Shri Ramdeobaba College of Engineering and Management for providing access to the NVIDIA server.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Gasparetto A, Scalera L. A brief history of industrial robotics in the 20th century. Advances in Historical Studies. 2019; 8(1):24-35.

[2] Chandan G, Jain A, Jain H. Real time object detection and tracking using Deep learning and openCV. In international conference on inventive research in computing applications 2018 (pp. 1305-8). IEEE.

[3] Jiao L, Zhang F, Liu F, Yang S, Li L, Feng Z, et al. A survey of deep learning-based object detection. IEEE Access. 2019; 7:128837-68.

[4] Minaee S, Luo P, Lin Z, Bowyer K. Going deeper into face detection: a survey. arXiv preprint arXiv:2103.14983. 2021.

[5] Ganokratanaa T, Aramvith S, Sebe N. Unsupervised anomaly detection and localization based on deep spatiotemporal translation network. IEEE Access. 2020; 8:50312-29.

[6] Elihos A, Balci B, Alkan B, Artan Y. Deep learning based segmentation free license plate recognition using roadway surveillance camera images. arXiv preprint arXiv:1912.02441. 2019.

[7] Yang X, Wang X. Recognizing license plates in real-time. arXiv preprint arXiv:1906.04376. 2019.

[8] Song H, Liang H, Li H, Dai Z, Yun X. Vision-based vehicle detection and counting system using deep learning in highway scenes. European Transport Research Review. 2019; 11(1):1-6.

[9] Cao J, Pang Y, Xie J, Khan FS, Shao L. From handcrafted to deep features for pedestrian detection: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2021.

[10] Janahiraman TV, Subuhan MS. Traffic light detection using tensorflow object detection framework. In international conference on system engineering and technology (ICSET) 2019 (pp. 108-13). IEEE.

[11] Shen L, Margolies LR, Rothstein JH, Fluder E, Mcbride R, Sieh W. Deep learning to improve breast cancer detection on screening mammography. Scientific Reports. 2019; 9(1):1-12.

[12] Ali AR, Li J, O'shea SJ. Towards the automatic detection of skin lesion shape asymmetry, color variegation and diameter in dermoscopic images. Plos One. 2020; 15(6):1-21.

[13] Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences. 1982; 79(8):2554-8.

[14] Viola P, Jones M. Robust real-time object detection. International Journal of Computer Vision. 2001.

[15] Lowe DG. Object recognition from local scale-invariant features. In proceedings of the seventh international conference on computer vision 1999 (pp. 1150-7). IEEE.

[16] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In proceedings of the conference on computer vision and pattern recognition 2016 (pp. 779-88). IEEE.

[17] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In proceedings of the conference on computer vision and pattern recognition 2014 (pp. 580-7). IEEE.

[18] Girshick R. Fast R-CNN. In proceedings of the international conference on computer vision 2015 (pp. 1440-8). IEEE.

[19] Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems. 2015; 28:91-9.

[20] He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. In proceedings of the international conference on computer vision 2017 (pp. 2961-9). IEEE.

[21] Amirian S, Rasheed K, Taha TR, Arabnia HR. Automatic image and video caption generation with deep learning: a concise review and algorithmic overlap. IEEE Access. 2020; 8:218386-400.

[22] Kojima A, Tamura T, Fukunaga K. Natural language description of human activities from video images based on concept hierarchy of actions. International Journal of Computer Vision. 2002; 50(2):171-84.

[23] Hakeem A, Sheikh Y, Shah M. CASE^ E: a hierarchical event representation for the analysis of videos. In AAAI 2004 (pp. 263-8).

[24] Khan MU, Gotoh Y. Describing video contents in natural language. In proceedings of the workshop on innovative hybrid approaches to the processing of textual data 2012 (pp. 27-35).

[25] Das P, Xu C, Doell RF, Corso JJ. A thousand frames in just a few words: lingual description of videos through latent topics and sparse object stitching. In proceedings of the conference on computer vision and pattern recognition 2013 (pp. 2634-41). IEEE.

[26] Khan MU, Al HN, Gotoh Y. A framework for creating natural language descriptions of video streams. Information Sciences. 2015; 303:61-82.

[27] Liu L, Ouyang W, Wang X, Fieguth P, Chen J, Liu X, et al. Deep learning for generic object detection: a survey. International Journal of Computer Vision. 2020; 128(2):261-318.

[28] Bay H, Tuytelaars T, Van GL. Surf: speeded up robust features. In European conference on computer vision 2006 (pp. 404-17). Springer, Berlin, Heidelberg.

[29] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In proceedings of the computer society conference on computer vision and pattern recognition. CVPR 2001. IEEE.

[30] Cortes C, Vapnik V. Support-vector networks. Machine Learning. 1995; 20(3):273-97.

[31] Ben-hur A, Horn D, Siegelmann HT, Vapnik V. Support vector clustering. Journal of Machine Learning Research. 2001:125-37.

[32] Wang J, Shen X, Pan W. On transductive support vector machines. Contemporary Mathematics. 2007; 443:7-20.

[33] He K, Zhang X, Ren S, Sun J. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015; 37(9):1904-16.

[34] Wang X, Shrivastava A, Gupta A. A-fast-RCNN: hard positive generation via adversary for object detection. In proceedings of the conference on computer vision and pattern recognition 2017 (pp. 2606-15). IEEE.

[35] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In proceedings of the conference on computer vision and pattern recognition 2015 (pp. 3431-40). IEEE.

[36] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot multibox detector. In European conference on computer vision 2016 (pp. 21-37). Springer, Cham.

[37] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: common objects in context. In European conference on computer vision 2014 (pp. 740-55). Springer, Cham.

[38] https://github.com/matterport/Mask_RCNN. Accessed 10 January 2021.

[39] https://www.robots.ox.ac.uk/~vgg/software/via. Accessed 11 February 2021.

[40] Bradski G. The openCV library. Dr. Dobb's Journal: Software Tools for the Professional Programmer. 2000; 25(11):120-3.

[41] Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object

detection. In proceedings of the conference on computer vision and pattern recognition 2017 (pp. 2117-25). IEEE.

[42] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In proceedings of the conference on computer vision and pattern recognition 2016 (pp. 770-8).

Manasi Namjoshi completed her Bachelor of Engineering in Computer Science and Technology in 2021 from Shri Ramdeobaba College of Engineering and Management, Nagpur, India. She is currently pursuing her Master of Science in Computer Science (Artificial Intelligence) from the University of Southern California, California, USA. Her research interests include Machine Learning, Artificial Intelligence and Deep Learning.
Email: manasinamjoshi19@gmail.com

Khushboo Khurana is an Assistant Professor at the Department of Computer Science and Engineering at, Shri Ramdeobaba College of Engineering and Management, Nagpur. She is also currently pursuing her Ph.D. from Visvesvaraya National Institute of Technology (VNIT), Nagpur. She holds a Bachelors in Computer Science and Engineering and a Masters in Technology from RCOEM, for which she also received a gold medal. Her topics of interest include Deep Learning and Video and Image Processing.
Email: khuranakp@rknec.edu

Appendix I

S. No.	Abbreviation	Description
1	CNN	Convolutional Neural Networks
2	FCN	Fully convolutional network
3	FPN	Feature Pyramid Network
4	mAP	Average Precision
5	RCNN	Region-based Convolutional Neural Network
6	RoI	Region of Interest
7	RPN	Regional Proposal Networks
8	SIFT	Scale-invariant feature transform
9	SPPnet	Spatial pyramid pooling networks
10	SURF	Speeded Up Robust Features
11	SVC	Support-vector clustering
12	SVM	Support Vector Machine
13	TSVM	Transductive Support Vector Machines
14	VIA	VGG Image Annotator
15	YOLO	You Only Look Once