Prevention of Side Pulling Using AWS Cloud and OpenCV

ABSTRACT

To reduce the accidents that occurs because of side pulling we have come up with a method to warn the crane handler and the people nearby. This device will have a Node Camera attached it using which photos of the surface below the hoist will be taken each second. And each time a photo is taken it will be sent to the cloud. In the cloud, using OpenCV we will detect the occurrence of side pulling if it happens. To detect this, we will be applying machine learning Algorithms in OpenCV. If a side pulling is detected a message is instantaneously sent to the crane handler from the cloud and also a buzzer beep to warn the people nearby.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 INTRODUCTION

In normal operation, overhead crane hoists are designed to lift an object at a vertical angle. A side pull is when a portion of the hoist acts horizontally, such as when the hoist lifts an object that has not been placed directly underneath it. When side pulling occurs it results in overload of the motor and takes in a lot of current. Because of this the motor's MCB trips. When you perform a side pulling action on a crane, you are pushing it past its mechanical limitations. The extreme amount of stress on the crane can cause a wire rope to snap—instantly and without warning. Needless to say, this could cause a very dangerous situation for any nearby personnel, as the wire rope could whip through the air; the load on the crane would swing vigorously and could potentially be dropped.

Side pulling is like trying to use your car as a pickup truck—it's not going to last. You will have more maintenance issues, and it will wear out quickly. If a load is lifted more than a few degrees off-centre, it multiplies the stress applied to the wire rope. This is very dangerous and can cause premature rope failure. Side pulling is an issue that starts and stops with crane operators. Crane operators need to be properly trained, and they need to retain what they've learned. For each and every lift, crane operators need to be aware of the dangers of side pulling and take extra care to ensure that the crane will lift vertically, not horizontally. Maintenance and plant operations managers need to be aware of these issues and work as a team to prevent safety incidents related to side pulling.

Overhead lifting operations are a necessary activity in workplaces around the globe. The ability to lift and move material safely is critical to the success of many businesses. Failure to follow safe lifting practices can lead to serious personal injury and cause damage to equipment and facilities.

Overhead hoists are designed to raise loads vertically.

Accordingly, the load being lifted must be centred under the hoist (Figure 1). Side pulling (Figure 2) occurs when attempting to lift any load that is not located directly under the hoist. Another form of side pulling occurs when a crane operator attempts to use the bridge or trolley drives to apply force to move an object horizontally when the load is not first fully suspended on the hoist and free of the floor or other support. Regardless of the manner in which side pull is applied, there are many unintended, damaging and potentially dangerous results that can occur. Side pulling a hoist or crane, in most cases, results in a violation of OSHA regulations, and numerous industry standards.

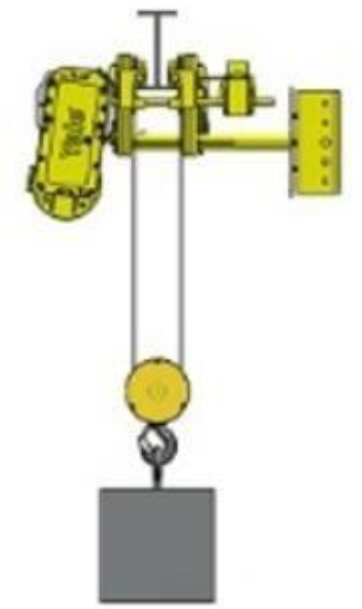


Figure 1

Figure 2

ASME B30.16, a safety standard for overhead hoists (underhung) states that:

Hoists shall not be operated unless the hoist unit is centred over the load, except when authorized by a qualified person who has determined that the components of the hoist and its mounting will not be overstressed. Should it be necessary to pick a load that is not centred under the hoist unit, precautions should be taken to control the swing of the load when it is picked clear of its support.

What are the dangers of side pulling?

As the load is lifted free of the floor or other support it will attempt to centre itself under the hoist, causing the load to rapidly swing in a horizontal arc **(Figure**

3). This pendulum effect can cause serious injury to personnel or damage to other equipment in the area.

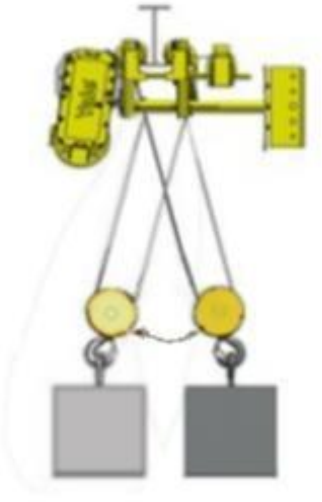


Figure 3

The wire rope or load chain can be forced out of the grooving or pockets on the hoist drum or lift wheel. This can damage the chain/rope, and may also cause damage to drums, sheaves, and other components. In the best-case scenario, this can lead to costly repairs and downtime. More importantly, it could cause the chain or wire rope to break and the load to drop, putting equipment, facilities, and personnel at serious risk.

Side pulling at an angle that is not in line with the length of the bridge or monorail **(Figure 4)** could cause the trolley hoist to tip, making the trolley inoperable. In the worst case, the trolley hoist could actually be pulled off of the beam. This side pull condition also puts stresses on the beam itself and could cause the beam to skew **(Figure 5)**.

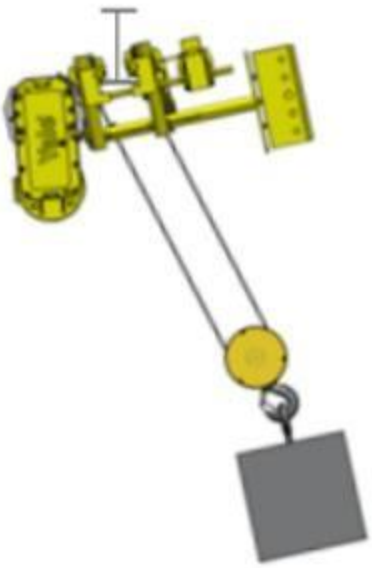


Figure 4

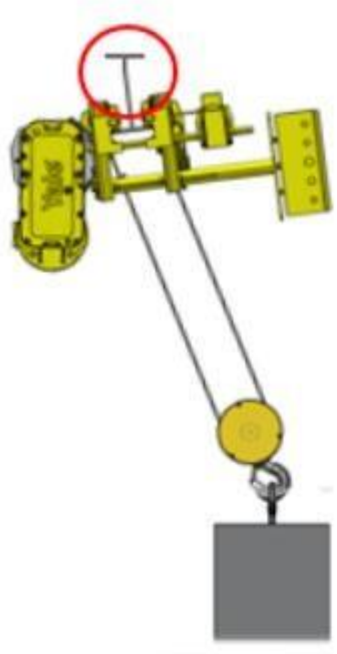


Figure 5

Side pulling is not considered “normal operation” of the hoist and therefore may void the manufacturer’s warranty. Attempting to lift a load that is located beyond the end of a bridge beam or monorail **(Figure 6)** could damage the safety stops at the end of the beam. In rare cases, this has caused the trolley hoist to fall off the end of the beam.

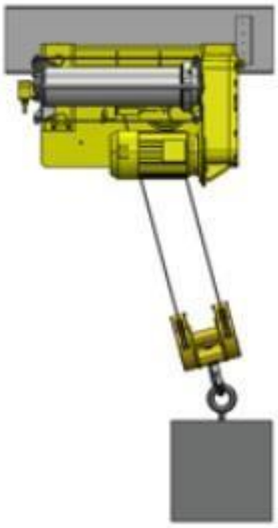


Figure 6

1.2 OBJECTIVES AND GOALS

A side pull is when part of the hoist works horizontally, as when lifting lifts an object not placed directly under it. When a side pull happens, it results in overcrowding and takes up a lot of current. As a result of this MCB car trip. When you perform the action of pulling on the side of the crane, you press it beyond its limits. Excessive pressure on the crane can cause the wire cord to explode — quickly and without warning. Needless to say, this could create a very dangerous situation for any nearby staff, as the telephone line could blow into the air; the load on the crane would be heavier and may be reduced. Pulling aside is like trying to use your car as a photography van - it won't last long. You will have many maintenance problems, and they will wear out quickly.

When the load is raised more than a few degrees from the centre, it increases the pressure applied to the wire cable. This is very dangerous and can cause cord failure prematurely. Pulling is a problem that starts and stops with crane operators. Crane operators need good training, and they need to keep what they have learned. For each lift, crane operators need to be aware of the dangers of dragging on the side and be extra careful to ensure that the crane will rise vertically, not horizontally. Plant maintenance and plant management needs to be aware of these problems and act as a team to prevent safety incidents related to side effects.

To reduce accidents that occur due to side drag we come up with a warning system that holds the crane and people nearby.

This device will have a Node Camera attached to it using which sub-hoist face images will be taken every second. And each time a photo is taken it will be sent to the cloud. In the cloud, using OpenCV we will see the availability of pull-out side effects. To find this we will be using machine learning algorithms in OpenCV.

When a pull is found the message is immediately sent to the person in charge of the crane from the cloud and buzzer beep to warn nearby people.

1.3 APPLICATIONS / FEATURES

Applications:

- Prevention of side pulling.
- Home Surveillance
- Car parking assistance
- Security Systems
- Water level monitoring

Features:

- Detects the position of the load from hoist point of view each second and sends the photo to the cloud.
- The machine learning code analyses the photo and masks the load with a circular layout.
- The masked image helps us to find the radius of the load.
- Using the radius, a requirement analysis is carried out.
- From the obtained results the present state is sent to the user.

Hardware and software requirements:

- ESP32-CAM board
- AWS Lambda
- AWS IoT Core
- Python (In AWS Lambda)
- OpenCV

2. MOTIVATION & REVIEW OF LITERATURE

2.1 MOTIVATION

One of the most necessary and frequent activities in workplaces is overhead lifting. It is an integral part of successful warehouses operations and management for many businesses. Meanwhile, it is also a technically risky operation and can lead to fatal accidents if done improperly. Overhead lifting hence needs to be done with the utmost precision following adequate safety practices. Any malpractice regarding the same will not only lead to equipment damage but also serious accidents and injuries. Hence we must ensure that all aspects of operational safety must be tended to while operating the overhead crane.

2.2 REVIEW OF LITERATURE

A. ***R. D. Castro-Zunti, J. Yépez and S. Ko, "License plate segmentation and recognition system using deep learning and OpenVINO," in IET Intelligent Transport Systems, vol. 14, no. 2, pp. 119-126, 2 2020, doi: 10.1049/ietits.2019.0481.***

In this paper, they used a LPSOCR DL system comprised of a feature extractor with DSCs and linear bottlenecks, to preserve accuracy and lessen parameter count, in conjunction with SSD architecture that itself uses DSCs.

B. ***M. Y. Arafat, A. S. M. Khairuddin and R. Paramesran, "Connected component analysis integrated edge-based technique for automatic vehicular license plate recognition framework," in IET Intelligent Transport Systems, vol. 14, no. 7, pp. 712-723, 7 2020, doi: 10.1049/iet-its.2019.0006.***

In this paper, the algorithm can also detect LPs with varying pixels ranging from 400 × 300 to high resolution 1280 × 720. Total 400 images of various conditions and 126 images from public dataset were tested and the proposed AVLPR system possesses a better detection, segmentation and recognition probability rate.

C. ***H. Li, P. Wang and C. Shen, "Toward End-to-End Car License Plate***

Detection and Recognition with Deep Neural Networks," in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 3, pp. 11261136, March 2019, doi: 10.1109/TITS.2018.2847291

In this paper they have presented a jointly trained network for simultaneous car license plate detection and recognition. With this network, car license plates can be detected and recognized all at once in a single forward pass, with both high accuracy and efficiency.

D. C. Liu and F. Chang, "Hybrid Cascade Structure for License Plate Detection in Large Visual Surveillance Scenes," in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 6, pp. 2122-2135, June 2019, doi: 10.1109/TITS.2018.2859348.

To address the problem of fast detecting small and vague license plates, they designed a hybrid cascade including three parts: the cascaded CST-pixel detector, the cascaded CC-Haarlike detector, and the cascaded ConvNet detector.

E. W. Min, X. Li, Q. Wang, Q. Zeng and Y. Liao, "New approach to vehicle license plate location based on new model YOLO-L and plate pre-identification," in IET Image 3 Processing, vol. 13, no. 7, pp. 1041-1049, 30 5 2019, doi: 10.1049/iet-ipr.2018.6449.

Here, a new approach to vehicle license plate location based on the new model YOLO-L and the plate pre-identification is proposed. K-means++ clustering for license plate region is used on the training set to select the best numbers and scales of candidate boxes and increase the accuracy of locating license plates

F. M. Y. Arafat, A. S. M. Khairuddin, U. Khairuddin and R. Paramesran, "Systematic review on vehicular licence plate recognition framework in intelligent transport systems," in IET Intelligent Transport Systems, vol. 13, no. 5, pp. 745-755, 5 2019, doi: 10.1049/iet-its.2018.5151.

In this paper VLPR framework involves Acquisition stage, Detection of LP (Texture, character, boundary information, colour, global, Miscellaneous attributes), Segmentation of LP (Vertical and horizontal, Character contour, connectivity of pixels, mathematical morphology attributes, implementing classifiers, characters prior knowledge), Recognition of LP (Pattern matching and deploying extracted attributes, deploying classifiers, ANN, statistical classifiers)

G. R. Bhargav and P. Deshpande, "Locating multiple license plates using scale, rotation, and colour-independent clustering and filtering techniques," in IET Image Processing, vol. 13, no. 12, pp. 2335-2345, 17 10 2019, doi:
10.1049/iet-ipr.2018.6237.

In this paper, they have proposed a new method for LP detection which uses multiple filtering operations applied to clusters created using clustering techniques based on geometrical properties of LP characters. The method can detect multiple LPs in an image. Thus, the approach is less restrictive as compared with most of the previously published work

H. L. Xie, T. Ahmad, L. Jin, Y. Liu and S. Zhang, "A New CNN-Based Method for Multi-Directional Car License Plate Detection," in IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 2, pp. 507517, Feb. 2018, doi: 10.1109/TITS.2017.2784093.

In this paper Introduced a new MD-YOLO model for multi-directional car license plate detection. The proposed model can elegantly solve the problem of multi-directional car license plate detection, and can also be deployed easily in real-time circumstances, because of its reduced computational complexity compared with previous CNN based methods.

I. M. A. Khan, M. Sharif, M. Y. Javed, T. Akram, M. Yasmin and T. Saba, "License number plate recognition system using entropy-based features selection approach with SVM," in IET Image Processing, vol. 12, no. 2, pp. 200-209, 2 2018, doi: 10.1049/iet-ipr.2017.0368.

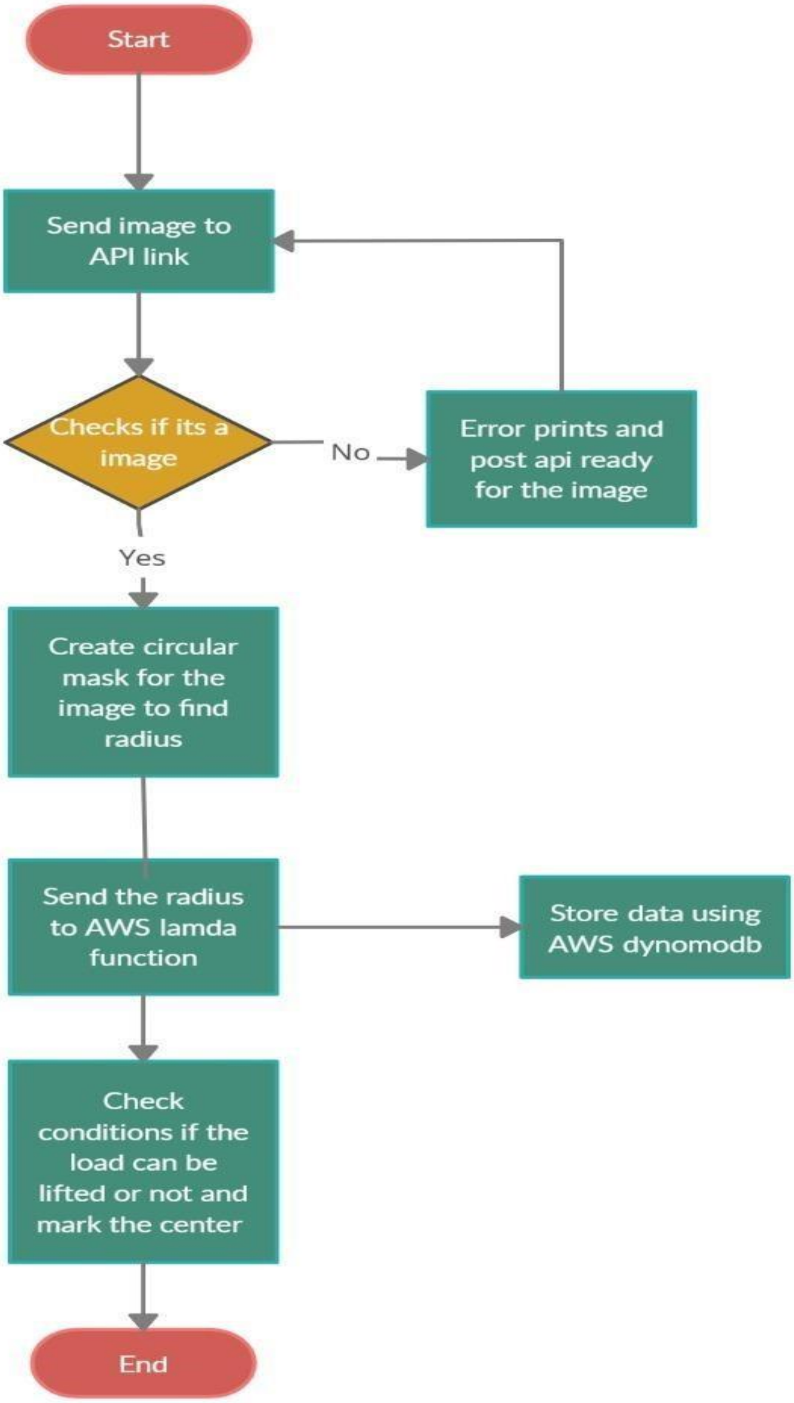
In this paper, a novel system is proposed for LPR based on the features selection. The proposed system comprises of three major steps: (i) preprocessing and segmentation of character area, (ii) features extraction of ROI, and (iii) features fusion using novel technique.

J. R. Panahi and I. Gholampour, "Accurate Detection and Recognition of Dirty Vehicle Plate Numbers for High-Speed Applications," in IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 4, pp. 767779, April 2017, doi: 10.1109/TITS.2016.2586520.

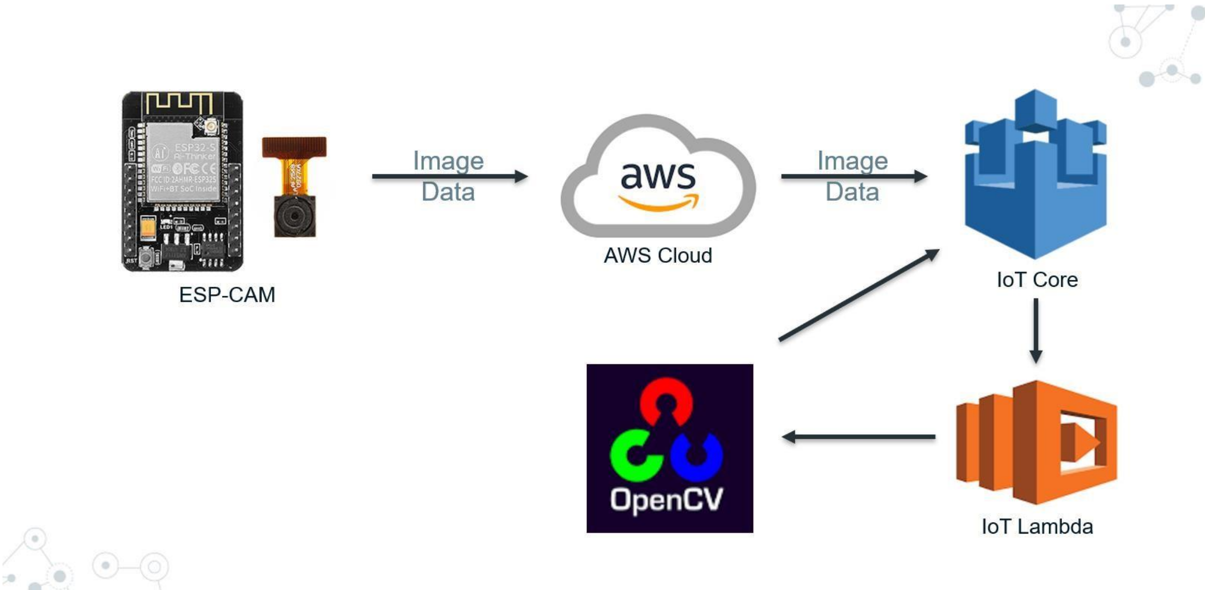
In this paper, an industrial, robust and reliable ANPR system for high speed applications is proposed. The main advantage of our system is its high detection and recognition accuracies on dirty plates. To achieve reliable evaluations, two new data sets were created and used in this paper: one for violation detection called "Crossroad Data set" and the other for vehicle counting in highways called "Highway Data set."

3. DESIGN METHODOLOGY

3.1 FLOW DIAGRAM

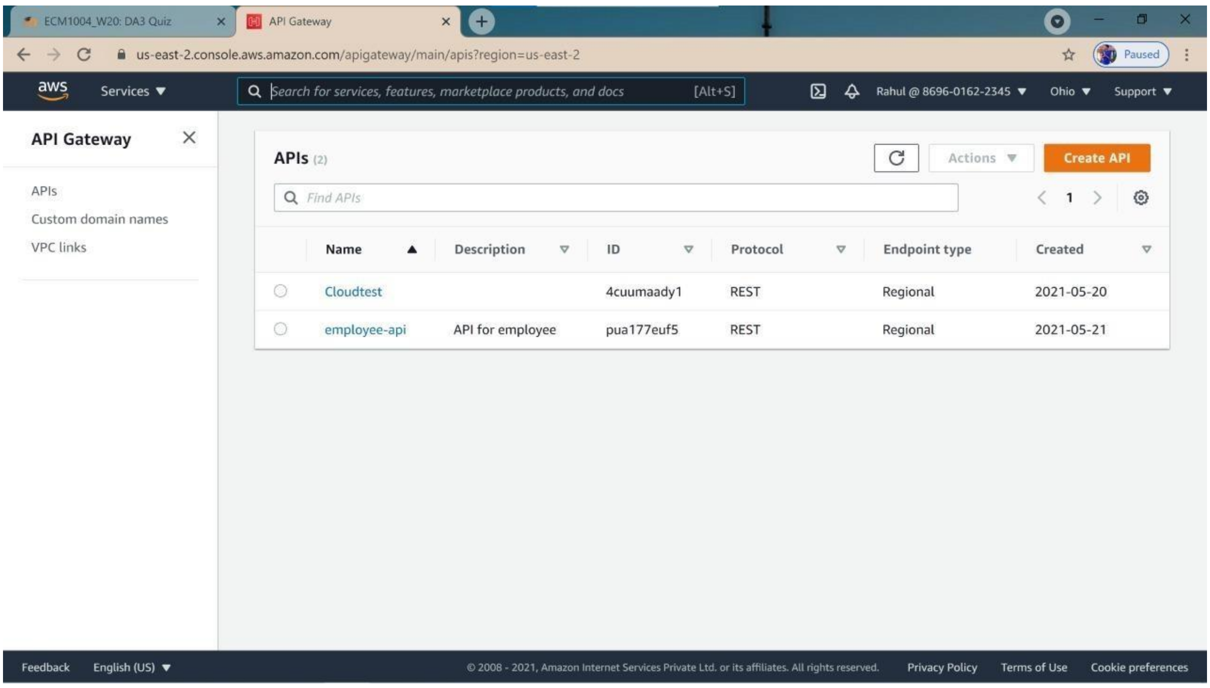


3.2 BLOCK DIAGRAM

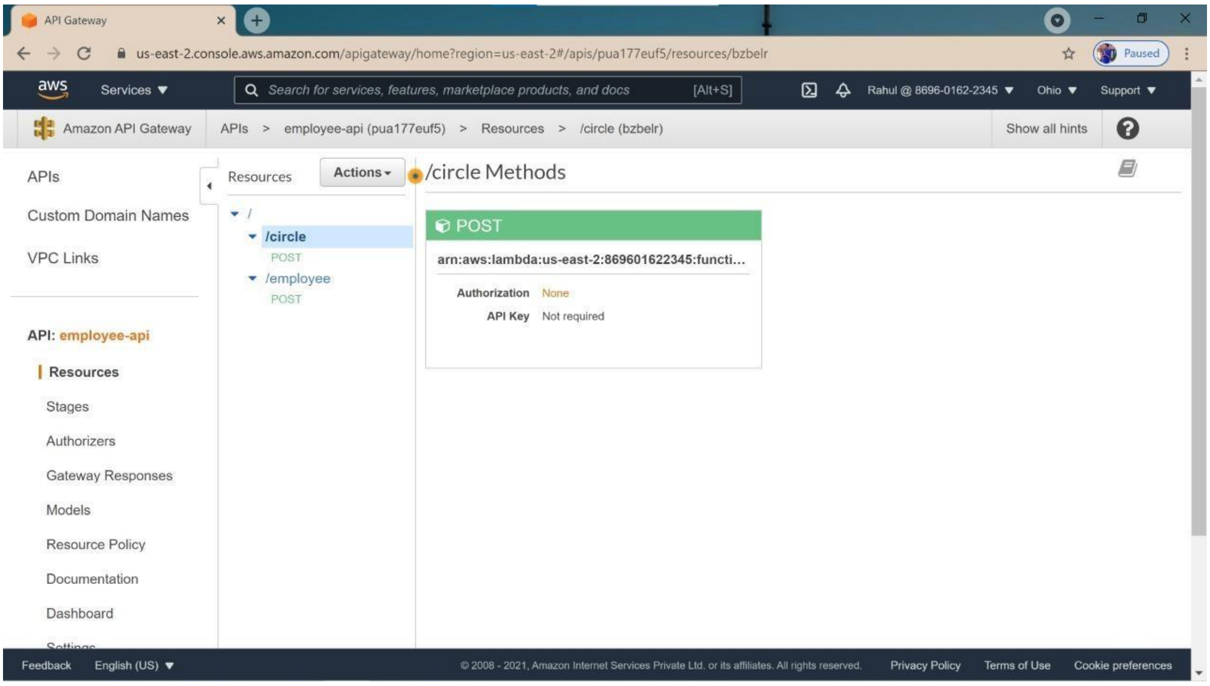


3.3 IMPLEMENTATION

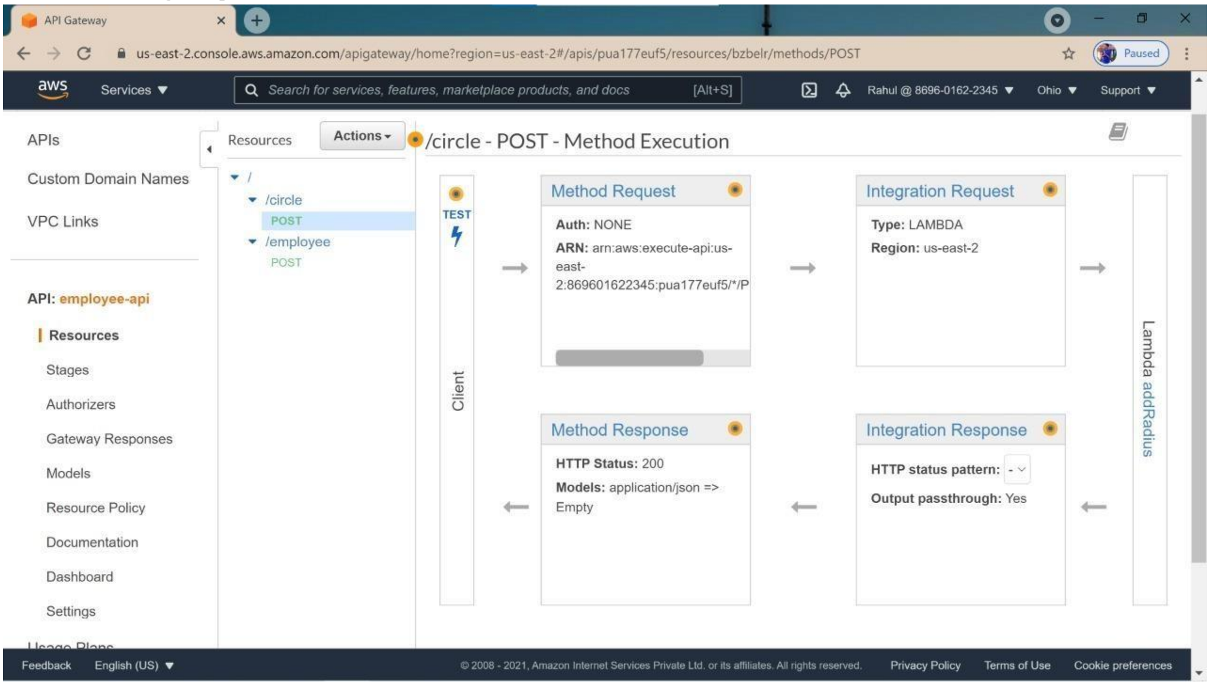
1.Creating an API gateway



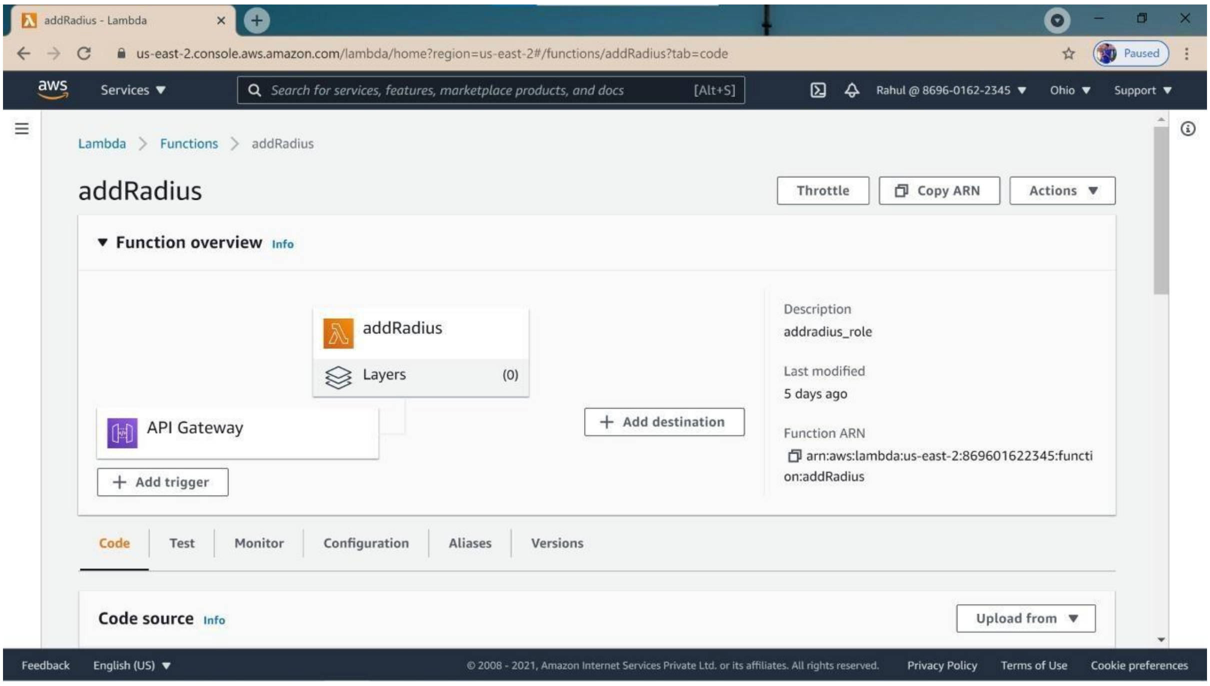
2. Creating resources and methods



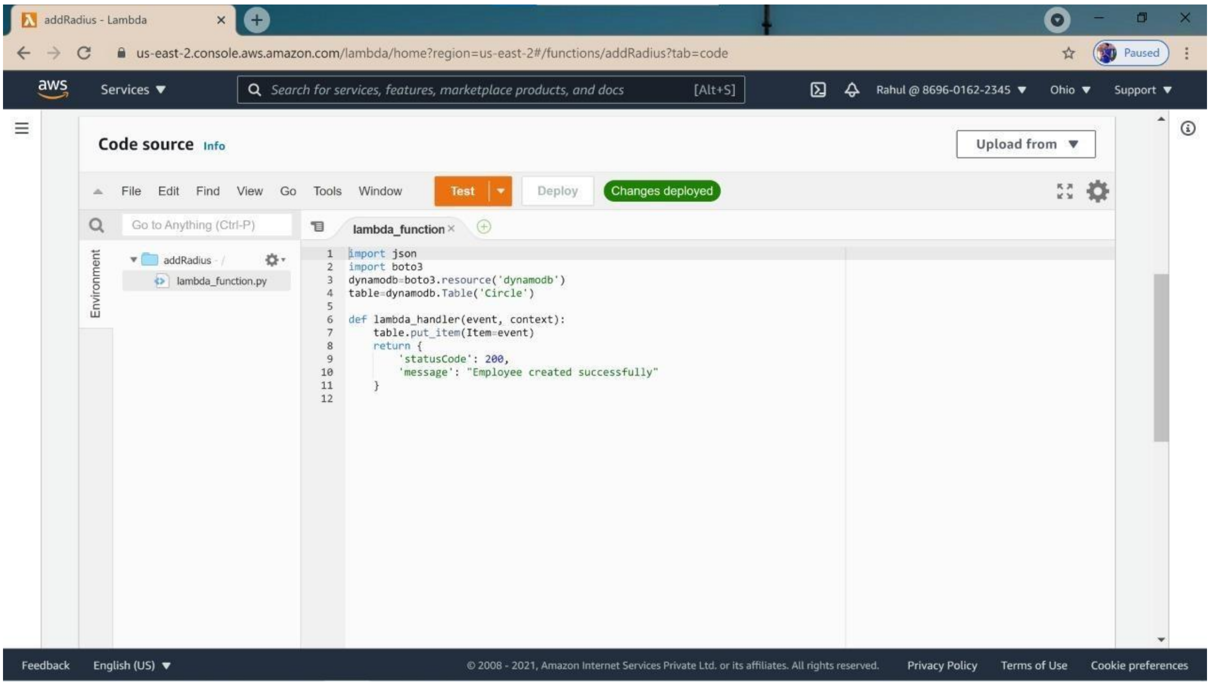
3. Attaching a function to POST method



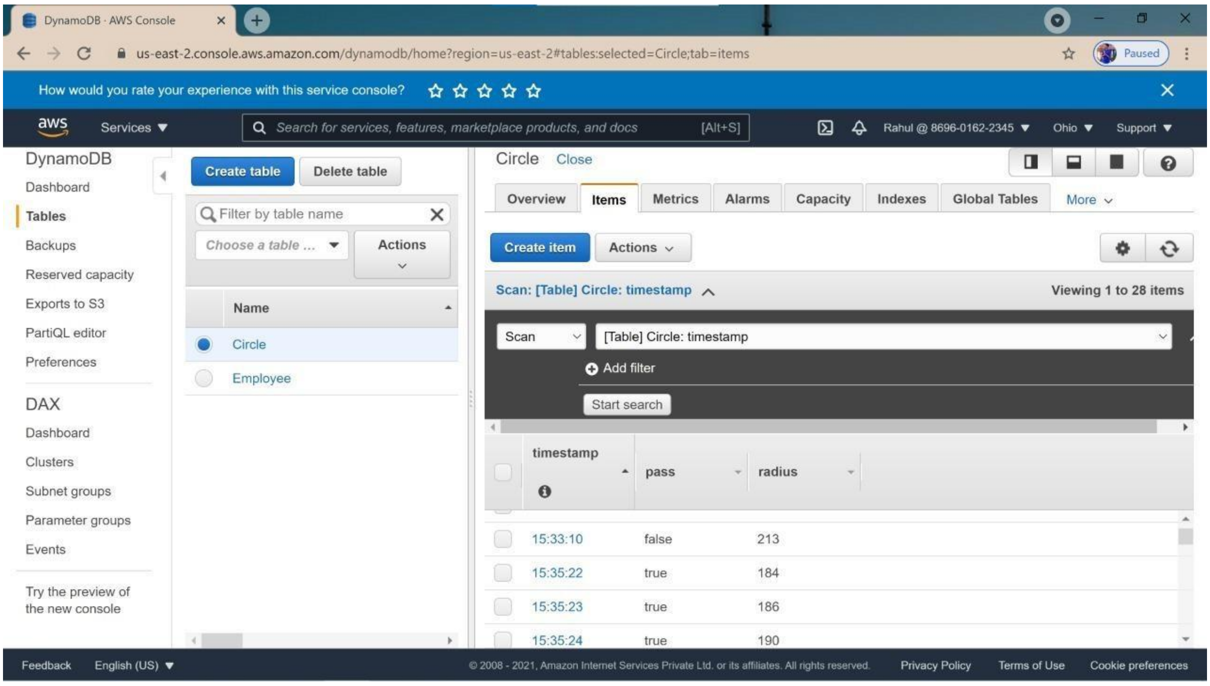
4. Editing the lambda permissions



5. Lambda code



6. Creating dynamodb table and Getting results



4. SOFTWARE/CLOUD PLATFORMS USED

4.1 AMAZON WEB SERVICES

Amazon web service is a platform that offers flexible, reliable, scalable, easyto-use and cost-effective cloud computing solutions. AWS is a comprehensive, easy to use computing platform offered Amazon. The platform is developed with a combination of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offering. Amazon Web Services offers a wide range of different business purpose global cloud-based products. The products include storage, databases, analytics, networking, mobile, development tools, enterprise applications, with a pay-as-you-go pricing model.

Here, are Cloud Compute Services offered by Amazon:

EC2(Elastic Compute Cloud) - EC2 is a virtual machine in the cloud on which you have OS level control. You can run this cloud server whenever you want. LightSail — This cloud computing tool automatically deploys and manages the computer, storage, and networking capabilities required to run your applications. Elastic Beanstalk — The tool offers automated deployment and provisioning of resources like a highly scalable production website. EKS (Elastic Container Service for Kubernetes) — The tool allows you to Kubernetes on Amazon cloud environment without installation. AWS Lambda — This AWS service allows you to run functions in the cloud. The tool is a big cost saver for you as you to pay only when your functions execute.

Following are the pros of using AWS services:

AWS allows organizations to use the already familiar programming models, operating systems, databases, and architectures. It is a cost-effective service that allows you to pay only for what you use, without any up-front or long-term commitments. You will not require to spend money on running and maintaining data centres. Offers fast deployments. You can easily add or remove capacity. You are allowed cloud access quickly with limitless capacity. Total Cost of Ownership is very low compared to any private/dedicated servers. Offers Centralized Billing and management. Offers Hybrid Capabilities Allows you to deploy your application in multiple regions around the world with just a few clicks.

Disadvantages of AWS:

If you need more immediate or intensive assistance, you'll have to opt for paid support packages. Amazon Web Services may have some common cloud computing issues when you move to a cloud. For example, downtime, limited control, and backup protection. AWS sets default limits on resources which differ from region to region. These resources consist of images, volumes, and snapshots. Hardware-level changes happen to your application which may not offer the best performance and usage of your applications.

4.2 CODING AND ANALYSIS

Masking Code: import cv2

import numpy as np

```
def nothing(x):
    pass

cap = cv2.VideoCapture(0) cv2.namedWindow("Trackbars")

cv2.createTrackbar("L - H", "Trackbars", 0, 179, nothing)
cv2.createTrackbar("L - S", "Trackbars", 0, 255, nothing)
cv2.createTrackbar("L - V", "Trackbars", 0, 255, nothing)
cv2.createTrackbar("U - H", "Trackbars", 179, 179, nothing)
cv2.createTrackbar("U - S", "Trackbars", 255, 255, nothing)
cv2.createTrackbar("U - V", "Trackbars", 255, 255, nothing)

while True:
    _, frame = cap.read()
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    l_h = cv2.getTrackbarPos("L - H", "Trackbars")
l_s = cv2.getTrackbarPos("L - S", "Trackbars")     l_v
= cv2.getTrackbarPos("L - V", "Trackbars")     u_h =
cv2.getTrackbarPos("U - H", "Trackbars")     u_s =
cv2.getTrackbarPos("U - S", "Trackbars")     u_v =
cv2.getTrackbarPos("U - V", "Trackbars")
```

```
    lower_blue = np.array([l_h, l_s, l_v])
upper_blue = np.array([u_h, u_s, u_v])     mask =
cv2.inRange(hsv, lower_blue, upper_blue)

    result = cv2.bitwise_and(frame, frame, mask=mask)

    cv2.imshow("frame", frame)
cv2.imshow("mask", mask)     cv2.imshow("result",
result)

    key   =   cv2.waitKey(1)
if key == 27:
        break

cap.release() cv2.destroyAllWindows()
```

Code for detection and sending the

message to AWS import cv2 import

```
numpy as np from datetime import
datetime

def nothing(x):
    pass

cap = cv2.VideoCapture(0) cv2.namedWindow("Trackbars")
```

```
cv2.createTrackbar("L - H", "Trackbars", 0, 179, nothing)
cv2.createTrackbar("L - S", "Trackbars", 0, 255, nothing)
cv2.createTrackbar("L - V", "Trackbars", 0, 255, nothing)
cv2.createTrackbar("U - H", "Trackbars", 179, 179, nothing)
cv2.createTrackbar("U - S", "Trackbars", 255, 255, nothing)
cv2.createTrackbar("U - V", "Trackbars", 255, 255, nothing)

while       True:
try:
        _, frame = cap.read()
        hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

        l_h = cv2.getTrackbarPos("L - H", "Trackbars")
        l_s = cv2.getTrackbarPos("L - S", "Trackbars")
l_v = cv2.getTrackbarPos("L - V", "Trackbars")
u_h = cv2.getTrackbarPos("U - H", "Trackbars")
u_s = cv2.getTrackbarPos("U - S", "Trackbars")
u_v = cv2.getTrackbarPos("U - V", "Trackbars")

        lower_blue = np.array([l_h, l_s, l_v])
upper_blue = np.array([u_h, u_s, u_v])        mask =
cv2.inRange(hsv, lower_blue, upper_blue)
```

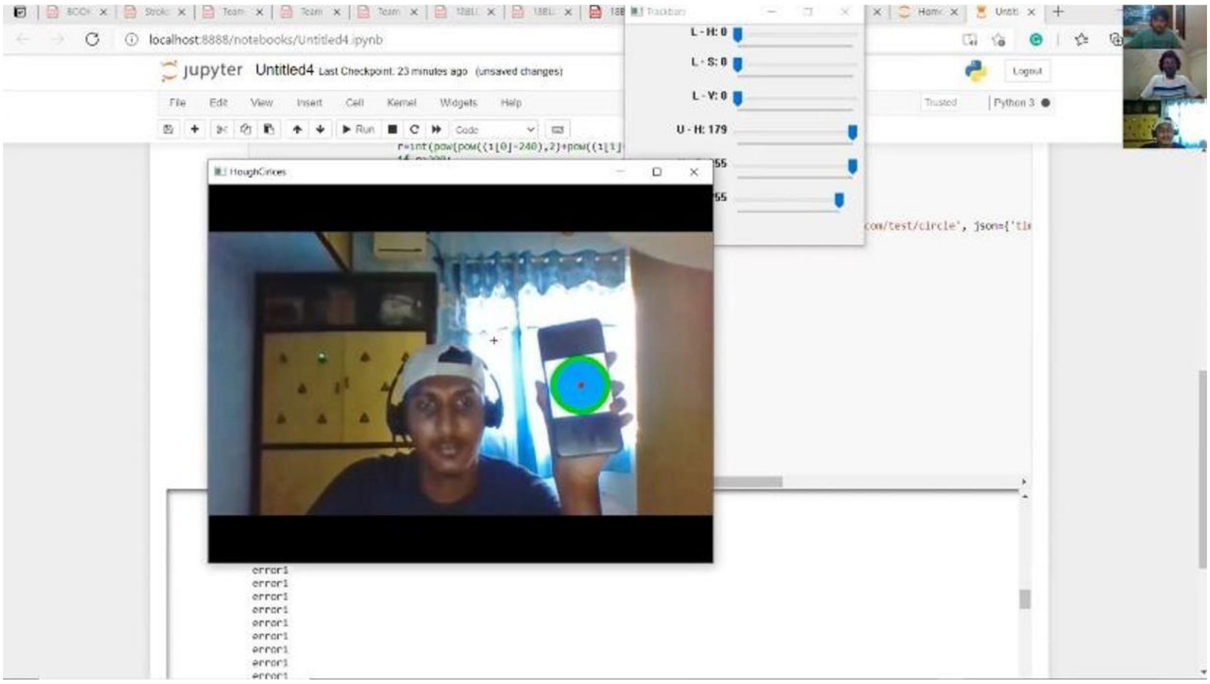
```
result = cv2.bitwise_and(frame, frame, mask=mask)
gray_img=cv2.cvtColor(result,cv2.COLOR_BGR2GRAY)
img= cv2.medianBlur(gray_img,5)
cimg = cv2.cvtColor(img,cv2.COLOR_GRAY2BGR)
circles=
cv2.HoughCircles(img,cv2.HOUGH_GRADIENT,1,120,param1=100,param2=
30,minRadius=20,maxRadius=35)
circles= np.uint16(np.around(circles))
for i in circles[0,:]:
cv2.circle(frame,(i[0],i[1]),i[2],(0,255,0),6)
r=int(pow(pow((i[0]-240),2)+pow((i[1]-320),2),0.5))
if r>200: p='false' else:
p='true'
now = datetime.now()
response = requests.post('https://pua177euf5.execute-api.us-east-
2.amazonaws.com/test/circle', json={'timestamp':
now.strftime("%H:%M:%S"),'radius': r,'pass':p })
print("Status code: ", response.status_code)
print("Printing Entire Post Request")
print(response.json())
cv2.circle(frame,(i[0],i[1]),2,(0,0,255),3)
cv2.imshow("HoughCirlces",frame) except
AttributeError as error:
print("error1")
```

```
        cv2.imshow("HoughCirlces",frame)
except TypeError as error:
        print("error1")
        cv2.imshow("HoughCirlces",frame)
key = cv2.waitKey(1)     if key == 27:
        break



cap.release() cv2.destroyAllWindows()
```

4.3 RESULTS AND ANALYSIS



We have successfully detected the object and analysed its position details.

5. CONCLUSION & FUTURE WORK

5.1 CONCLUSION & FUTURE WORK

Moving heavy materials has always been a risky proposition, even with the help of overhead cranes. The side-loading caused can prove to be hazardous for everyone involved with the operation. Proper precautions have to be taken in order to reduce any amount of side pull and avoid fatal personal and equipment damage. This project ensures that.

Implementation using ESP32/STM32 for 24/7 working with the crane to produce reliable output. Future plans include converting it to real time video processing instead of image processing. It will make it expensive but it will be more reliable for companies who are ready to pay.

5.2 COST ANALYSIS

S.No.	Application	Cost	Used	total
1	API Gateway	Free for 1 million API calls 1$ for next each Million till 300 million 0.9$ after that	Less than 1000 calls	0
2	AWS Lambda	Free for 1 million Calls 0.2$ for every 1 million calls	Less than 1000 calls	0
3	DynamoDB	Based on capacity of usage	Less than 400 Kb of data	0
Total				0$

REFERENCES

1. R. D. Castro-Zunti, J. Yépez and S. Ko, "License plate segmentation and recognition system using deep learning and OpenVINO," in IET Intelligent

Transport Systems, vol. 14, no. 2, pp. 119-126, 2 2020, doi: 10.1049/iet-its.2019.0481.

2. M. Y. Arafat, A. S. M. Khairuddin and R. Paramesran, "Connected component analysis integrated edge based technique for automatic vehicular license plate recognition framework," in IET Intelligent Transport Systems, vol. 14, no. 7, pp. 712-723, 7 2020, doi: 10.1049/ietits.2019.0006.
3. H. Li, P. Wang and C. Shen, "Toward End-to-End Car License Plate Detection and Recognition With Deep Neural Networks," in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 3, pp. 1126-1136, March 2019, doi: 10.1109/TITS.2018.2847291.
4. C. Liu and F. Chang, "Hybrid Cascade Structure for License Plate Detection in Large Visual Surveillance Scenes," in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 6, pp. 2122-2135, June 2019, doi: 10.1109/TITS.2018.2859348.
5. W. Min, X. Li, Q. Wang, Q. Zeng and Y. Liao, "New approach to vehicle license plate location based on new model YOLO-L and plate preidentification," in IET Image Processing, vol. 13, no. 7, pp. 1041-1049, 30 5 2019, doi: 10.1049/iet-ipr.2018.6449.
6. M. Y. Arafat, A. S. M. Khairuddin, U. Khairuddin and R. Paramesran, "Systematic review on vehicular licence plate recognition framework in intelligent transport systems," in IET Intelligent Transport Systems, vol. 13, no. 5, pp. 745-755, 5 2019, doi: 10.1049/iet-its.2018.5151.
7. R. Bhargav and P. Deshpande, "Locating multiple license plates using scale, rotation, and colour-independent clustering and filtering techniques," in IET Image Processing, vol. 13, no. 12, pp. 2335-2345, 17 10 2019, doi: 10.1049/iet-ipr.2018.6237.
8. L. Xie, T. Ahmad, L. Jin, Y. Liu and S. Zhang, "A New CNN-Based Method for Multi-Directional Car License Plate Detection," in IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 2, pp. 507-517, Feb. 2018, doi: 10.1109/TITS.2017.2784093.
9. M. A. Khan, M. Sharif, M. Y. Javed, T. Akram, M. Yasmin and T. Saba, "License number plate recognition system using entropy-based features

selection approach with SVM," in IET Image Processing, vol. 12, no. 2, pp. 200-209, 2 2018, doi: 10.1049/iet-ipr.2017.0368.

10. Y. Yang, D. Li and Z. Duan, "Chinese vehicle license plate recognition using kernel-based extreme learning machine with deep convolutional features," in IET Intelligent Transport Systems, vol. 12, no. 3, pp. 213219, 4 2018, doi: 10.1049/iet-its.2017.0136.

11. R. Panahi and I. Gholampour, "Accurate Detection and Recognition of Dirty Vehicle Plate Numbers for High-Speed Applications," in IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 4, pp. 767-779, April 2017, doi: 10.1109/TITS.2016.2586520.

12. M. Wafy and A. M. M. Madbouly, "Efficient method for vehicle license plate identification based on learning a morphological feature," in IET Intelligent Transport Systems, vol. 10, no. 6, pp. 389-395, 8 2016, doi: 10.1049/iet-its.2015.0064.

13. M. R. Asif, Q. Chun, S. Hussain and M. S. Fareed, "Multiple licence plate detection for Chinese vehicles in dense traffic scenarios," in IET Intelligent Transport Systems, vol. 10, no. 8, pp. 535-544, 10 2016, doi: 10.1049/iet-its.2016.0008.

14. A. H. Ashtari, M. J. Nordin and M. Fathy, "An Iranian License Plate Recognition System Based on Color Features," in IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 4, pp. 1690-1705, Aug. 2014, doi: 10.1109/TITS.2014.2304515.

15. S. Du, M. Ibrahim, M. Shehata and W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 2, pp. 311-325, Feb. 2013, doi: 10.1109/TCSVT.2012.2203741.

16. F. M. Oliveira-Neto, L. D. Han and M. K. Jeong, "An Online SelfLearning Algorithm for License Plate Matching," in IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 4, pp. 1806-1816, Dec. 2013, doi: 10.1109/TITS.2013.2270107.

17. R. G. J. Wijnhoven and P. H. N. de With, "Identity verification using computer vision for automatic garage door opening," in IEEE Transactions on Consumer Electronics, vol. 57, no. 2, pp. 906-914, May 2011, doi: 10.1109/TCE.2011.5955239.

18. A. Conci, J. Carvalho and T. Rauber, "A complete system for vehicle plate localization, segmentation and recognition in real life scene.," in IEEE Latin America Transactions, vol. 7, no. 5, pp. 497-506, Sept. 2009, doi: 10.1109/TLA.2009.5361185.

19. C. Henry, S. Y. Ahn and S. Lee, "Multinational License Plate Recognition Using Generalized Character Sequence Detection," in IEEE Access, vol. 8, pp. 35185-35199, 2020, doi: 10.1109/ACCESS.2020.2974973.

20. W. Weihong and T. Jiaoyang, "Research on License Plate Recognition Algorithms Based on Deep Learning in Complex Environment," in IEEE Access, vol. 8, pp. 91661-91675, 2020, doi: 10.1109/ACCESS.2020.2994287.

21. I. V. Pustokhina et al., "Automatic Vehicle License Plate Recognition Using Optimal K-Means With Convolutional Neural Network for Intelligent Transportation Systems," in IEEE Access, vol. 8, pp. 9290792917, 2020, doi: 10.1109/ACCESS.2020.2993008.

22. M. C. Rademeyer, A. Barnard and M. J. Booysen, "Optoelectronic and Environmental Factors Affecting the Accuracy of Crowd-Sourced Vehicle-Mounted License Plate Recognition," in IEEE Open Journal of Intelligent Transportation Systems, vol. 1, pp. 15-28, 2020, doi: 10.1109/OJITS.2020.2991402.

23. B. B. Yousif, M. M. Ata, N. Fawzy and M. Obaya, "Toward an Optimized Neutrosophic k-Means With Genetic Algorithm for Automatic Vehicle License Plate Recognition (ONKM-AVLPR)," in *IEEE Access*, vol. 8, pp. 49285-49312, 2020, doi: 10.1109/ACCESS.2020.2979185.

24. W. Wang, J. Yang, M. Chen and P. Wang, "A Light CNN for End-to-End Car License Plates Detection and Recognition," in IEEE Access, vol. 7, pp. 173875-173883, 2019, doi: 10.1109/ACCESS.2019.2956357. H. Seibel, S. Goldenstein and A. Rocha, "Eyes on the Target: SuperResolution and License-Plate Recognition in Low-Quality Surveillance Videos," in *IEEE Access*, vol. 5, pp. 20020-20035, 2017, doi: 10.1109/ACCESS.2017.2737418.