DIGITAL NATURALIST – AI ENABLED TOOL FOR BIODIVERSITY RESEARCHERS

IBM NALAIYA THIRAN

PROJECT REPORT

Submitted By

YAGASH K	**(620819106097)**
SRIVINAYAGAMOORTHY S	**(620819106088)**
PANNEERSELVAM M	**(620819106059)**
SIVA S	**(620819106085)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

GNANAMANI COLLEGE OF TECHNOLOGY,

NAMKKAL-637504

ANNA UNIVERSITY::CHENNAI 600 025

NOVEMBER 2022

DIGITAL NATURALIST – AI ENABLED TOOL FOR BIODIVERSITY RESEARCHERS

IBM NALAIYA THIRAN

PROJECT REPORT

Submitted By

YAGASH K	**(620819106097)**
SRIVINAYAGAMOORTHY S	**(620819106088)**
PANNEERSELVAM M	**(620819106059)**
SIVA S	**(620819106085)**

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

GNANAMANI COLLEGE OF TECHNOLOGY,

NAMAKKAL-637504

ANNA UNIVERSITY::CHENNAI 600 025

NOVEMBER 2022

BONAFIDE CERTIFICATE

Certified that this project report titled **"VirtualEye - Life Guard for Swimming Pools to Detect Active Drowning"** is the bonafidework of **"YAGASH K (620819106097),SRIVINAYAGAMOORTHY S(620819106088), PANNEERSELVAM M (620819106059) SIVA S(620819106085)"** who carried out the project work under my supervision.

SIGNATURE

Dr. K.ANAND, B.E., M.E.,

FACULTY MENTOR

ASSISTANT PROFESSOR

Department of Electronics and

Communication Engineering,

Gnanamani college of Technology,

pachal,

Namakkal- 637 504.

-------------------------------- -----------------------------

SPOC **HEAD OF THE DEPARTMENT**

ACKNOWLEDGEMENT

At the outset, we express our heartfelt gratitude to **GOD,** who has been our strength to bring this project to light.

At this pleasing moment of having successfully completed our project, we wish to convey our sincere thanks and gratitude to our beloved **Mr.K.ANAND,** who has provided all the facilities to us.

We would like to convey our sincere thanks to our beloved Principal **Dr.T.K.KANNAN,** for forwarding us to do our project and offering adequate duration in completing our project.

We express our sincere thanks to our Head of the Department **Dr.R.PRABU,** Department of Electronics and Communication Engineering for fostering the excellent academic climate in the Department.

We express our pronounced sense of thanks with deepest respect and gratitude to our faculty mentor **Dr.K.ANAND,** Assistant Professor Department of Electronics and Communication Engineering and to our faculty evaluator **Mrs.SWATHI,** Industry Mentor for his valuable and precious guidance and for having amicablerelation.

With deep sense of gratitude, we extend our earnest and sincere thanks to our SPOC **Mrs.T.THENMOZHI,** Assistant Professor Department of Electronics and Communication Engineering for her guidance and encouragement during this project. all the faculty members of our department, friends and students who helped us directly and indirectly in all aspects of the project work to get completed successfully.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATION

NCC	National Cadet Corps
CNN	Convolutional Neural Network
RCNN	Region based Convolutional Neural Network
LBP	Local Binary Patterns
SVM	Support Vector Machine
ODI	Omnidirectional Image
MNIST	Modified National Institute of Standards and Technology database
ZCA	Zero-phase Component analysis
OTP	One time Password
UPI	Unified Payments Interface
API	Application Programming Interface
HMAC	Hash-Based Message Authentication Code
DFD	Data Flow diagram
UI	User Interface
HTTP	Hyper Text Markup Language
CSS	Cascading Style sheets
JS	JavaScript
WSGI	Web Server Gateway Interface
IAM	Identity and Access Management
OWASP	Open web application security project
ML	Machine Learning
NFT	Non-functional testing
SHA	Secure Hash Algorithm

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

Drowning incidents are potentially severe but thankfully rare for most lifeguards. Due to the infrequency of drowning incidents, the visual search for such occurrences is challenging. The difficulties involved in detecting infrequent drowning targets are reflected in other areas of real-world visual search with uncommon targetitems, such as airport security screenings. For example, Wolfe et al. found low-prevalence targets were missed more frequently than high-prevalence targets (occurring on 50% of trials),with error rates of 30% and 7%, respectively. In regards to lifeguarding, visual search has been defined as observing part of an aquatic environment (beaches, pools, open water), and processing and assessing the events happeningwithin that locationWhile this definition suggests that the surveillance ofthe water is a fundamental and critical role of the lifeguard, there is relatively little focus on training in these areas This is reflected in the UK National Pool Lifeguard Qualification (NPLQ) training manual where only 6 out of 214 pages are dedicated to the education of scanning and observation behaviours With this limited focus on visual training, lifeguards may be underprepared for detecting struggling swimmers in a timely manner.

1.2 PROJECT PURPOSE

Video surveillance can be used as a tool for monitoring and security. Observing public and private sites has increasingly become a very sensitive issue. The visual monitoring capabilities can be employed in many different locations to help people live more safely. Video- based surveillance systems are designed and installed in places such as railway stations, airports, and even dangerous environments. Image processing, pattern recognition and machine-vision based methods are efficient ways for real-time intelligent monitoring of the objects or events of interest. The existing surveillance systems deliver valued information in monitoring of large areas. Applying intelligence in video surveillance systems allows real-time monitoring of places, people and their activities. The tracking approach can change with varying targets and can change from a single camera to multiple camera configurations. Tracking methods in video surveillance use different parameters such as objects' motion, position, path of movement and velocity , biometrics such as skin color or clothes color and many more. The tracking must be robust and overcome occlusion and noise which are common problems in monitoring. One important environment that the need for monitoring systems is crucially sensed is the swimming pool. Each year many people including children are drowned or very close to drowning in the deeps of the swimming pools, and the life guards are not trained well enough to handle these problems . This raises the need for having a system that will automatically detect the drowning person and alarm the lifeguards of such danger. Real-time detection of a drowning person in swimming pools is a challenging task that requires an accurate system.

CHAPTER 2

LITERATURE SURVEY

2.1 PROBLEM STATEMENT:

Video surveillance can be used as a tool for monitoring and security. Observing public and private sites has increasingly become a very sensitive issue. The visual monitoring capabilities can be employed in many different locations to help people live more safely. Video- based surveillance systems are designed and installed in places such as railway stations, airports, and even dangerous environments. Image processing, pattern recognition and machine-vision based methods are efficient ways for real-time intelligent monitoring of the objects or events of interest. The existing surveillance systems deliver valued information in monitoring of large areas. Applying intelligence in video surveillance systems allows real-time monitoring of places, people and their activities. The tracking approach can change with varying targets and can change from a single camera to multiple camera configurations. Tracking methods in video surveillance use different parameters such as objects' motion, position, path of movement and velocity , biometrics such as skin color or clothes color and many more. The tracking must be robust and overcome occlusion and noise which are common problems in monitoring. One important environment that the need for monitoring systems is crucially sensed is the swimming pool. Each year many people including children are drowned or very close to drowning in the deeps of the swimming pools, and the life guards are not trained well enough to handle these problems . This raises the need for having a system that will automatically detect the drowning person and alarm the lifeguards of such danger. Real-time detection of a drowning person in swimming pools is a challenging task that requires an accurate system. The challenge is due to the presence of water ripples, shadows and splashes and therefore detection needs to have high accuracy.

Why is the visual search task of a lifeguard so difficult?

Many factors have a negative impact on successful target detection in basic studies of visual search, including crowding,target- distractor similarity and attentional set.
For instance, crowding is typically defined as an effect that limits perception of objects' features when surrounded by neighbouring distractors. The ability to recognize and respond tocrowded targets is dramatically reduced during visual search. The negative impacts of crowding overlap considerably with the related concept of visual clutter. As the number of items in a search area increases, the space between items becomes smaller and this limits the searcher's attention to smaller areas. This phenomenon of crowding has obvious relevance to lifeguarding, for example, with increased numbers of swimmers, physical space within the zone of supervision will become visually cluttered, causing delayed reaction times in visual searches . This problem of visual clutter is also noted in other research studies, both in the laboratory and in applied settings. For example, found that individuals were better at detecting targets in rural scenes with limited clutter, compared to urban city scenes with high rates of visual clutter. Ho et al.,found similar effects in young and old people in their visual searches of roads, with more clutter in the search area having a detrimental effect on searches of road signs. Similarly passive drowning can be mistaken for intended submergence or floating face down in the water.The inclusion of extra target behaviours alongside those of drowning and distress also add to the complexity of lifeguard visual search: not only must they keep alert for drowning targets but they must also be attentive to risk-taking behaviours, rule breaking, and the quality of the water. Research into attentional set suggests that the greater the number of target features thatmay define a target, the less efficient visual search is Recent research argues that this is because differentfeatures in the search set need to be searched for sequentially . A related problem is termination of search due to the detection of a task relevant (but non-drowning) target: if a lifeguard identifies swimmers engaging in risk taking behaviours, they would need to interrupt their scan of the pool to intervene and stop any potentially dangerous actions thus possibly missing a drowning target.. However, unlike the static images used in surveillance based visual search tasks(such as airportsecurity and radiology), lifeguards are faced with

the challenge of dynamic scenes. Lifeguardsare required to observe swimmers moving around a pool. The scene they observe constantly changes. This creates difficulties in using memory as a swimmer that has already been checkedmay later begin to drown or move into an area that has already been scrutinised. What may bemore relevant to the searches of lifeguards is the theory behind Multiple Object Tracking. This theory suggests that searchers are able to track a small number of multiple moving objects around a screen by pre-attentively tagging them. In recent research ithas been shown that expert sportsmen, such as basketball players who need to be able to followthe ball and other players in a game, have substantial superior visual skill in complex neutral dynamic tasks after training in three dimensional multiple object tracking. It was also found that these expert sportsmen have a greater capacity for learning these skills compared to amateur and non-athletes. Regular surveillance of swimmers may help to improve lifeguards' search skills in tracking multiple objects at a time, resulting in an increased ability to detect drowning swimmers in the search zone.

Procedure:

In order to recruit lifeguards, the experimenter arranged testing sessions at various pools and leisure centres around Nottingham and Leicester, with a quiet office or side-room acting as the laboratory. Control participants were tested under similar conditions. Participants were given written instructions and asked to fill in a consent form and demographic questionnaire. Prior to the study, participants were made aware that they would be searching for any potentially drowning victims from a lifeguard's perspective, and that the study may contain a drowning. They were told to press the space bar on the laptop upon identifying a drowning target that would require lifeguard assistance or intervention, and were also told that this would terminate the clip (preventing detection of a subsequent drowning target should their first response have been premature). Participants were then given a practice trial followed by a final opportunity to ask any remaining questions before the trials began. Once the test had ended participants were fully debriefed and thanked for their time and participation. This research was conducted with approval obtained from the University ethics committee and run in accordance with British Psychological Society guidelines.

Discussion:

The results of the current study have found the predicted advantage for lifeguards in spotting and responding to drowning targets in a swimming pool situation. They identified both active and passive drowning targets more frequently and more quickly than control participants, which suggest that experience and/or training have positively influenced the visual search and target processing skills of this specialist group. Lifeguards also appear to have a higher threshold for responding to a drowning target. This may reflect their greater sensitivity to visual cues that discriminate between drowning and normal swimming. Additionally, lifeguards may be more aware of the dangers of committing to a potentially drowning target. Once a response is initiated in a pool situation (e.g. entering the water to rescue the drowning swimmer) the lifeguard is limited in their ability to spot secondary drowning targets. Thus lifeguards may need greater evidence before responding, though this did not negatively impact on their time to respond when they chose to do so. lifeguards should still be able to respond to fully submerged targets, even those who are prone at the bottom of a pool. Brener and Oostman (2002) demonstrated the difficulty of spotting submerged targets when they timed lifeguard responses to unexpected manikins that were allowed to sink in pools. Fourteen percent of lifeguards failed to spot the submerged manikin with three minutes, with 90% of them failing to spot the manikin within the industry standard 10 seconds. While a surface-based training tool may increase the detection of drowning targets prior to complete submergence, if this is not 100% reliable, then it may result in those few submerged targets who slip through the net of vigilance being even less easy to spot due to emphasis in training being on rescuing victims at the surface of the water, and always being given a warning before practicing deep water rescues. Nonetheless, the current study has demonstrated a valid testing paradigm that can be extended to include the above suggestions. The method holds promise as a form of assessment, and could lead to the development of more useful training techniques, while simultaneously providing greater insight into visual search skills in complex, real world scenes.

Social Impact:

Designed for whom has to guarantee every day the safety in public and intensive-use pools, AngelEye LifeGuard detects potential drownings and promptly notifies you. It features the latest artificial intelligence technology and adapts to the needs of the user. It’s the ultimate drowning detection system for those who demand the ultimate in safety.

Scalability of the Solution:

Scalability of the Solution A robust human tracking and semantic event detection within the context of video surveillance system capable of automatically detecting drowning incidents in a swimming pool.

Solution description:

a real time drowning detection method based on HSV color space analysis is presented which uses prior knowledge of the video sequences to set the best values for the color channels. Our method uses a HSV thresholding mechanism along with Contour detection to detect the region of interest in each frame of video sequences. The presented software can detect drowning person in indoor swimming pools and sends an alarm to the lifeguard rescues if the previously detected person is missing for a specific amount of time.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

In empathy map canvas, we gathered views of different people and it helps us to understand the user behaviour. It helps to gain a deeper insight into our customers. The four empathy map quadrants look at what the user says, thinks, feels, and does and it helps us to understand the user better.

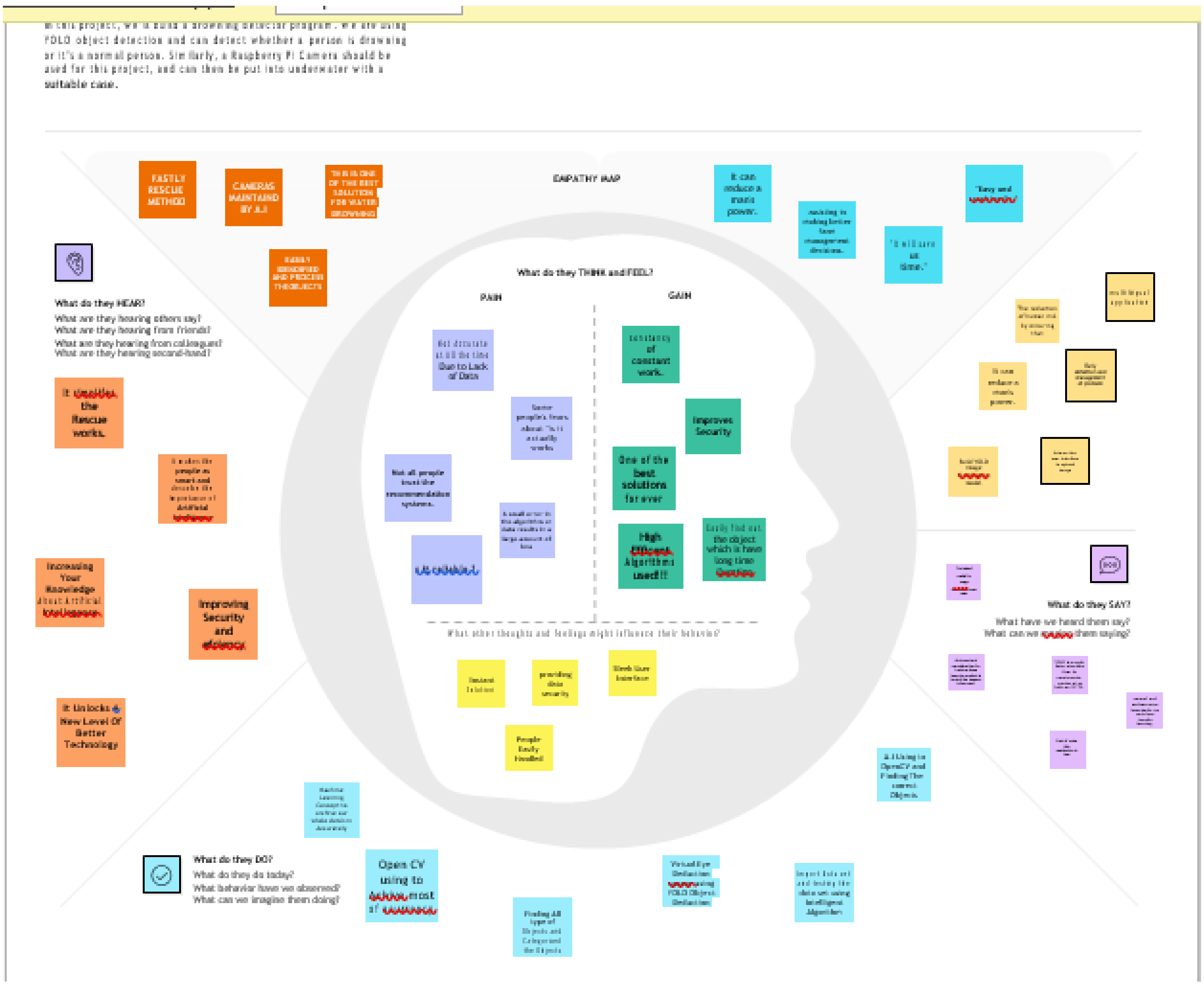


Figure 3.1. Empathy map

3.2 IDEATION AND BRAINSTROMING

Ideation helps us to gather the idea of our individual teammates and grouping them based on the possibilities and similarities. It also helps us to prioritize our ideas based on importance and feasibility.

TECHNICAL ARCHITECTURE:

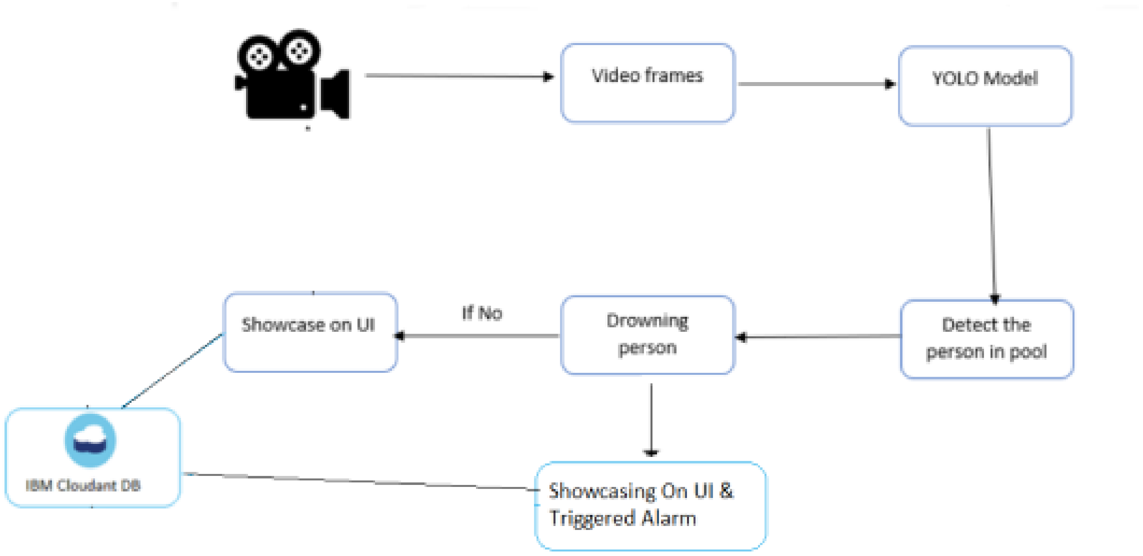


Figure 1: Architecture and data flow of the VirtualEye Life Guard for Swimming Pools to Detect Active Drownin

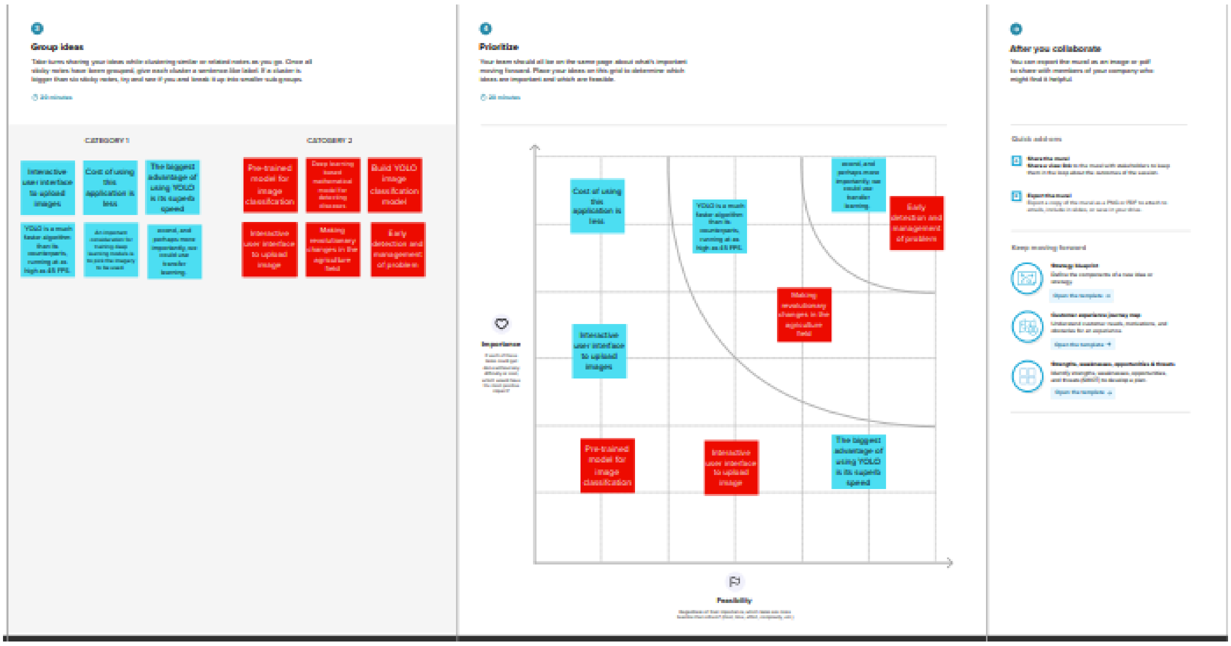
STEP 1: TEAM GATHERING, COLLABORATION AND SELECTING THE PROBLEM STATEMENT

Figure 3.2.2. Ideation and brainstorming

STEP 2: BRAINSTROM

Figure 3.2.3. Brainstorm, Idea Listing and Grouping

STEP 4: IDEA PRIORITIZATION



Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Drowning incidents are potentially severe but thankfully rare for most lifeguards. Due to the infrequency of drowning incidents, the visual search for such occurrences is challenging (Lanagan-Leitzel, Skow & Moore, 2015). The difficulties involved in detecting infrequent drowning targets are reflected in other areas of real-world visual search with uncommon targetitems, such as airport security screenings (Wolfe, Horowitz & Kenner, 2005; Biggs & Mitroff,2015).
2.	Idea / Solution description	a real time drowning detection method based on HSV color space analysis is presented which uses prior knowledge of the video sequences to set the best values for the color channels. Our method uses a HSV thresholding mechanism along with Contour detection to detect the region of interest in each frame of video sequences. The presented software can detect drowning person in indoor swimming pools and sends an alarm to the lifeguard rescues if the previously detected person is missing for a specific amount of time.
3.	Novelty / Uniqueness	The presented algorithm for this system is tested on several video sequences recorded in swimming pools in real conditions and the results are of high accuracy with a high capability of tracking individuals in real time.
4.	Social Impact / Customer Satisfaction	Designed for whom has to guarantee every day the safety in public and intensive-use pools, AngelEye LifeGuard detects potential drownings and promptly notifies you. It features the latest artificial intelligence

3.4 PROBLEM SOLUTION FIT

In problem solution fit, a compatibility test is done between our problem and its proposed solution and based on the outcome the below listed parameters are described.

Table 3.4.1. Customer Segments and constraints, Available solutions

Project Title: VirtualEye - Life Guard For Swimming Pools To Detect Active Drowning

Project Design Phase-I - Solution Fit **Team ID:** PNT2022TMID31103

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

- People and kinds Security Guard
- Beach and Parks
- Homes , Hotels and Malls
- Where there is a swimming pool, there is also needed

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

- This is basically a web application , Which is Supported in almost all devices.
- The easy graphical representation make a clear understanding for all people.
- The Results for their problem will be in minute .

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have?

In this project, we'll build a drowning detector program. We are using YOLO object detection and can detect whether a person is drowning or it's a normal person.

Explore AS, differen

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for

- IT'S EFFCIENT METHOD OF EVERY HUMAN CARE WATER RESOURCE MANAGEMENT
- EVERY RESCUSE DEPARTMENT EASILY IDENDIFIED WHO FACED ISSUES DURING SWIMMING BY OUR HIGH EFFIENT METHODE
- FASTEST EFFECTIVE METHODE TO IDENDIFIED AND SOLVED THE ISSUES

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back

- kind of super-vision model is not very reliable for the ability of lifeguardsto deal with emergencies is weak
- the rescue speed of drowned swimmers is also very slow.
- it is hardfor lifeguards to maintain a high concentration of attentionfor a long time.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done? usage and benefits; indirectly associated: customers spend free

- Its saves many of people's life and makes their process faster.
- By using our product , they able to saves a lot of
- money spend for a expert.
- It improves their field growth with our product.
- It ensures the causes previously and provide solutions before the drowning happens.

Focus on J&P, tap into BE, understand RC

Table 3.4.2. Jobs to be done,
Root cause, Behaviour

Focus on J&P, tap into BE … stand RC			
	3. TRIGGERS TR What triggers customers to act? PEOPLE AND MAINLY KIDS FACED DROWNING ISSUES DURING SWIMMING	**10. YOUR SOLUTION** SL If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour. • Let's understand how this project is used to detect a drowning person. • So, the initial step is to identify the person first and draw a blue rectangle box around them. • After drawing the rectangular box, this project store the center position of the person from the created rectangular box.	**8. CHANNELS of BE** **8.1 ONLINE** What kind of actions do cus **8.2 OFFLINE** What kind of actions do cust them for customer developm • Their Data an • Its Highly imp most of the d
	4. EMOTIONS: BEFORE / AFTER EM How do customers feel when they face a problem or a job and afterwards? MORE PEOPLE'S FACED DROWNING ISSUES, SOME CASES CASUED SEVIER PROBLEM IN SWIMMERS AND KIDS		

PROJECTS ALL SUBMISSIONS AND RESOURCES:

TEAM ID: PNT2022TMID31103

TEAM MEMBERS:

YAGASH K	**(620819106097)**
SRIVINAYAGAMOORTHY S	**(620819106088)**
PANNEERSELVAM M	**(620819106059)**
SIVA S	**(620819106085)**

PROJECT PLANNING PHASE

PROJECT MILESTONE

Date	25 October 2022
Team ID	PNT2022TMID31103
Project Name	Virtual Eye - Life Guard for Swimming Pools to Detect Active Drowning
Maximum Marks	4 Marks

S.NO	MILESTONE	DESCRIPTION	DURATION
1	Prerequisites	Prerequisites are all the needs at the requirement level needed for the execution of the different phases of a project.	1 WEEK
2	Create & Configure IBM cloud services	IBM Cloud provides solutions that enable higher levels of compliance, security, and management, with proven architecture patterns and methods for rapid delivery for running mission-critical workloads.	1 WEEK
3	Develop the python script	A Python script is a set of commands included in a file that is intended to be run similarly to a program. The concept is that the file will be run or performed from the command line or from within a Python interactive shell to perform a particular activity. Of course, the file includes methods and imports different modules.	3 WEEKS
4	Develop web application	A web application (or web app) is application software that runs in a web browser, unlike software programs that run locally and natively on the operating system (OS) of the device.	1 WEEK

5	Ideation phase	Ideation is the process where you generate ideas and solutions through sessions such as Sketching, Prototyping, Brainstorming, Brain writing, Worst Possible Idea, and a wealth of other ideation techniques.	1 WEEK
6	Project design phases	Project design is an early phase of a project where the project's key features, structure, criteria for success, and major deliverables are planned out. The aim is to develop one or more designs that can be used to achieve the desired project goals.	1 WEEK
7	Project planning phase	In the Planning Phase, the Project Manager works with the project team to create the technical design, task list, resource plan, communications plan, budget, and initial schedule for the project, and establishes the roles and responsibilities of the project team and its stakeholders.	1 WEEK
8	Project development phase	Project development is the process of planning and allocating resources to fully develop a project or product from concept to go-live.	4 WEEKS

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	28 October 2022
Team ID	PNT2022TMID31103
Project Name	Virtual Eye - Life Guard for Swimming Pools to Detect Active Drowning
Maximum Marks	8 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a lifeguard , I can register for the application by entering my email, password, and confirming mypassword.	2	High	K.YAGASH
Sprint 1	User conformation	USN-2	As a lifeguard, I will receive the conformation mail once I have registered for the application	2	High	M.PANNEERSELVAM
Sprint-1	Login	USN-3	As a lifeguard , I can log into the application by entering email& password	2	High	S.SRIVINAYAGAMOORTHY
Sprint-2	Cloudant DB	USN-1	Create DB	2	High	S.SIVA

Sprint-3	Coding (Accessing datasets)	USN-1	Coding is a set of instructions used to manipulate information so that a certain input results in a particular output.	2	High	K.YAGASH, M.PANNEERSELVAM, S.SRIVINAYAGAMOORTHY S.SRIVINAYAGAMOORTHY, S.SIVA
Sprint-4	Application building	USN-1	As a Lifeguard , It will show the current Information of the swimming pool	1	High	K.YAGASH, M.PANNEERSELVAM, S.SRIVINAYAGAMOORTHY S.SRIVINAYAGAMOORTHY, S.SIVA

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	4 Days	28 Oct 2022	27 Oct 2022	20	07 Oct 2022
Sprint-2	20	5 Days	29 Oct 2022	08 Nov 2022	20	08 Nov 2022
Sprint-3	20	8 Days	02 Nov 2022	12 Nov 2022	20	10 Nov 2022
Sprint-4	20	9 Days	10 Nov 2022	15 Nov 2022	20	12 Nov 2022

Delivery plan sprint-1

Live Location Tracking:

GPS is installed on gadget to track its current location can be tracked on android app and via SMS request sent from parent phone to safety gadget. Outputs of live location tracking

2) Panic Alert Systems:

Panic alert system on gadget is triggered during panic situation, automatic call and SMS are triggered to parental phone. The alert is also updated to the cloud for purpose of app monitoring. Fig. 4. Outputs of panic alert system.

3) Stay Connected Feature:

Stay connected feature is used to trigger call and pre-defined SMS anytime from gadget to parental phone by just pressing a button and also parent can make SMS and call to the gadget anytime.

4) Health Monitoring System:

Health monitoring system is implemented using heart beat sensor, temperature sensor which is updated to the cloud and also can be monitored via app. The current value of sensors can be obtained using SMS request sent to gadget from parent phone. Outputs of health monitoring system.

5) Gadget Plugged or Unplugged Monitoring:

Gadget plug or unplugged is monitored using contact switch installed on smart gadget, as soon as the device is unplugged, an alert is provided to parent phone via SMS and it is also updated to cloud for app monitoring.

6) Boundary monitoring system:

This is used to track the safety gadget using the binding gadget by implementing signal strength concept as soon as the safety gadget moves far away from the BLE listener gadget then an alert is provided to itself. Listener device and broad cast device .

7. Overview of safety gadget Figure 7 shows the circuit connection with sensors. The temperature sensor, pulse sensor, BLE module, GSM module and GPS module are shown. 6. Limitation The system is dependent on communication signal/network

Delivery plan sprint-2

Live Location Tracking:

GPS is installed on gadget to track its current location can be tracked on android app and via SMS request sent from parent phone to safety gadget. Outputs of live location tracking

2) Panic Alert Systems:

Panic alert system on gadget is triggered during panic situation, automatic call and SMS are triggered to parental phone. The alert is also updated to the cloud for purpose of app monitoring. Fig. 4. Outputs of panic alert system.

3) Stay Connected Feature:

Stay connected feature is used to trigger call and pre-defined SMS anytime from gadget to parental phone by just pressing a button and also parent can make SMS and call to the gadget anytime.

4) Health Monitoring System:

Health monitoring system is implemented using heart beat sensor, temperature sensor which is updated to the cloud and also can be monitored via app. The current value of sensors can be obtained using SMS request sent to gadget from parent phone. Outputs of health monitoring system.

5) Gadget Plugged or Unplugged Monitoring:

Gadget plug or unplugged is monitored using contact switch installed on smart gadget, as soon as the device is unplugged, an alert is provided to parent phone via SMS and it is also updated to cloud for app monitoring.

6) Boundary monitoring system:

This is used to track the safety gadget using the binding gadget by implementing signal strength concept as soon as the safety gadget moves far away from the BLE listener gadget then an alert is provided to itself. Listener device and broad cast device .

7. Overview of safety gadget Figure 7 shows the circuit connection with sensors. The temperature sensor, pulse sensor, BLE module, GSM module and GPS module are shown. 6. Limitation The system is dependent on communication signal/network

```
[net]
# Testing
# batch=1
# subdivisions=1
# Training
batch=64
subdivisions=16
width=608
height=608
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.01
burn_in=1000
max_batches = 500200
policy=steps
steps=400000,450000
scales=.1,.1

[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
pad=1
activation=leaky

# Downsample

[convolutional]
batch_normalize=1
filters=64
size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=32
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
activation=leaky
```

```
[shortcut]
from=-3
activation=linear

# Downsample

[convolutional]
batch_normalize=1
filters=128
size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=64
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=64
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

# Downsample

[convolutional]
batch_normalize=1
```

```
filters=256
size=3
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
```

```
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear


[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
```

```
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

# Downsample

[convolutional]
batch_normalize=1
filters=512
size=3
stride=2
```

```
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear


[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear


[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
```

```
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear


[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear


[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
```

```
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear


[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

# Downsample

[convolutional]
batch_normalize=1
filters=1024
size=3
```

```
stride=2
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=1024
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=1024
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=1024
size=3
stride=1
pad=1
```

```
activation=leaky

[shortcut]
from=-3
activation=linear

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
filters=1024
size=3
stride=1
pad=1
activation=leaky

[shortcut]
from=-3
activation=linear

######################

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=1024
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=1024
```

```
activation=leaky

[convolutional]
batch_normalize=1
filters=512
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=1024
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=255
activation=linear


[yolo]
mask = 6,7,8
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,
156,198,  373,326
classes=80
num=9
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1


[route]
layers = -4

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[upsample]
stride=2

[route]
layers = -1, 61



[convolutional]
```

```
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=512
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=512
activation=leaky

[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=512
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=255
activation=linear


[yolo]
mask = 3,4,5
```

```
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,
156,198,  373,326
classes=80
num=9
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1



[route]
layers = -4

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[upsample]
stride=2

[route]
layers = -1, 36



[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=256
activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
```

```
pad=1
filters=256
activation=leaky

[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky

[convolutional]
batch_normalize=1
size=3
stride=1
pad=1
filters=256
activation=leaky

[convolutional]
size=1
stride=1
pad=1
filters=255
activation=linear


[yolo]
mask = 0,1,2
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,
156,198,  373,326
classes=80
num=9
jitter=.3
ignore_thresh = .7
truth_thresh = 1
random=1
```

```
#import necessary packages

import cv2

import os

import numpy as np

from .utils import download_file


initialize = True

net = None

dest_dir = os.path.expanduser('~') + os.path.sep + '.cvlib' + os.path.sep + 'object_detection' +
os.path.sep + 'yolo' + os.path.sep + 'yolov3'

classes = None

#colors are BGR instead of RGB in python

COLORS = [0,0,255], [255,0,0]


def populate_class_labels():


   #we are using a pre existent classifier which is more reliable and more efficient than one

   #we could make using only a laptop

   #The classifier should be downloaded automatically when you run this script

   class_file_name = 'yolov3_classes.txt'

   class_file_abs_path = dest_dir + os.path.sep + class_file_name

   url = 'https://github.com/Nico31415/Drowning-Detector/raw/master/yolov3.txt'

   if not os.path.exists(class_file_abs_path):

      download_file(url=url, file_name=class_file_name, dest_dir=dest_dir)

   f = open(class_file_abs_path, 'r')

   classes = [line.strip() for line in f.readlines()]


   return classes




def get_output_layers(net):
```

```
    #the number of output layers in a neural network is the number of possible
    #things the network can detect, such as a person, a dog, a tie, a phone...
    layer_names = net.getLayerNames()

    output_layers = [layer_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]

    return output_layers


def draw_bbox(img, bbox, labels, confidence, Drowning, write_conf=False):

    global COLORS
    global classes

    if classes is None:
        classes = populate_class_labels()

    for i, label in enumerate(labels):

        #if the person is drowning, the box will be drawn red instead of blue
        if label == 'person' and Drowning:
            color = COLORS[0]
            label = 'DROWNING'
        else:
            color = COLORS[1]

        if write_conf:
            label += ' ' + str(format(confidence[i] * 100, '.2f')) + '%'
```

```
        #you only need to points (the opposite corners) to draw a rectangle. These points
        #are stored in the variable bbox
        cv2.rectangle(img, (bbox[i][0],bbox[i][1]), (bbox[i][2],bbox[i][3]), color, 2)

        cv2.putText(img, label, (bbox[i][0],bbox[i][1]-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

    return img

def detect_common_objects(image, confidence=0.5, nms_thresh=0.3):

    Height, Width = image.shape[:2]
    scale = 0.00392

    global classes
    global dest_dir

    #all the weights and the neural network algorithm are already preconfigured
    #as we are using YOLO

    #this part of the script just downloads the YOLO files
    config_file_name = 'yolov3.cfg'
    config_file_abs_path = dest_dir + os.path.sep + config_file_name

    weights_file_name = 'yolov3.weights'
    weights_file_abs_path = dest_dir + os.path.sep + weights_file_name

    url = 'https://github.com/Nico31415/Drowning-Detector/raw/master/yolov3.cfg'

    if not os.path.exists(config_file_abs_path):
        download_file(url=url, file_name=config_file_name, dest_dir=dest_dir)
```

```
url = 'https://pjreddie.com/media/files/yolov3.weights'

if not os.path.exists(weights_file_abs_path):
    download_file(url=url, file_name=weights_file_name, dest_dir=dest_dir)




global initialize
global net

if initialize:
    classes = populate_class_labels()
    net = cv2.dnn.readNet(weights_file_abs_path, config_file_abs_path)
    initialize = False

blob = cv2.dnn.blobFromImage(image, scale, (416,416), (0,0,0), True, crop=False)

net.setInput(blob)

outs = net.forward(get_output_layers(net))

class_ids = []
confidences = []
boxes = []

for out in outs:
    for detection in out:
        scores = detection[5:]
        class_id = np.argmax(scores)
        max_conf = scores[class_id]
        if max_conf > confidence:
```

```
            center_x = int(detection[0] * Width)
            center_y = int(detection[1] * Height)
            w = int(detection[2] * Width)
            h = int(detection[3] * Height)
            x = center_x - w / 2
            y = center_y - h / 2
            class_ids.append(class_id)
            confidences.append(float(max_conf))
            boxes.append([x, y, w, h])


indices = cv2.dnn.NMSBoxes(boxes, confidences, confidence, nms_thresh)

bbox = []
label = []
conf = []

for i in indices:
    i = i[0]
    box = boxes[i]
    x = box[0]
    y = box[1]
    w = box[2]
    h = box[3]
    bbox.append([round(x), round(y), round(x+w), round(y+h)])
    label.append(str(classes[class_ids[i]]))
    conf.append(confidences[i])

return bbox, label, conf
```

REFERENCES:

https://www.linkedin.com/company/aqua-teik, Journal of Computational Information Systems 9: 21 (2013) 8619{8627 Available at

http://www.Jofcis.com , International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 3, Special Issue 2, March 2015

Avramidis, S., Butterly, R., & Llewellyn, D. (2009). Drowning incident rescuer characteristics: Encoding the first component of the 4w model. International Journal of Aquatic research, 3, 66-82. Berbaum,

K. S., Franken, E. A., Jr., Caldwell, R. T., & Schartz, K. M. (2010). Satisfaction of search in traditional radiographic imaging. In E. Samei, & E. Krupinski (Eds.),

The handbook of medical image perception and techniques (pp. 107–138). Cambridge: Cambridge University Press. Biggs, A. T., Cain, M. S., Clark, K., Darling, E. F., & Mitroff, S. R. (2013).

Assessing visual search performance differences between Transportation Security Administration Officers and nonprofessional visual searchers.

Visual Cognition, 21(3), 330-352. Biggs,

T., & Mitroff, S. R. (2015). Improving the Efficacy of Security Screening Tasks: A Review of Visual Search Challenges and Ways to Mitigate Their Adverse Effects.

Applied Cognitive Psychology, 29(1), 142-148. Blackwell, A. (2016). National Pool Lifeguard Qualification: The lifeguard 8th edition. RLSS UK: Warwickshire.

Blackwell, A., Dillon, T., McCready, C., & Talbot, J. (2012).

National Pool Lifeguard Qualification: The Lifeguard. RLSS UK: Warwickshire. Brener, J., & Oostman, M. (2002). Lifeguards watch, but they don’t always see! World Waterpark Magazine, 14–16. DeMers, J., & Giles, M. C. (2011).

Scanning techniques. In United States lifeguard standards: An evidence based review and report by the United States lifeguard standards coalition. Retrieved from http://c.ymcdn.com/sites/www.usla.org/resource/resmgr/lifeguard_library/lifeguard_st andards_final_re.pdf.