Project Report Form

- Date:19 November 2022
- Team ID:PNT2022TMID39505
- Project Title:VirtualEye-Lifeguard for Swimming Pools toDetect the Active Drowning

1.INTRODUTION

1.1Project Overview

Title:***VirtualEye*** *- Life Guard For Swimming Pools To Detect Active Drowning*. Recently, there has been growing interest around the topic of drowning detection systems (DDS) in the

sport and leisure industry both across the UK and globally. Advancements in technology, coupled withthe importance of pool safety, has led to its growing prominence, with mention of DDS now in documentssuch as the latest UK standards document for health and safety in swimming pools (Healthand Safety Executive, 2018). However, the topic is a debated area for various reasons explored in thisreview. Whilst there are plenty of academic articles dedicated to the technology and design behind theseproducts in the fields of biometrics, computer science and electronic engineering, there is limitedacademic research investigating their application to real-world scenarios. Furthermore, there isuncertainty around their use alongside traditional lifeguarding; whether international testing standards(ISO standards) are robust enough; and general risks affecting the effectiveness of these products. This includes factors such as water clarity, high pool occupancy, lighting, glare and attractions such as waterslides and wave machines. These concerns alongside the lack of research and high installation costshave resulted in a reluctance by some operators to incorporate DDS into their pools. This signifies the importance of independent research into DDS.

1.2Purpose

This literature review aims to:

>> Establish and outline what is known on Drowning Detection Systems.

>> Evaluate the current literature on Drowning Detection Systems, including their use in indoorpool environments along with interaction with traditional lifeguarding.

>> Better understand where DDS are positioned in the health and safety landscape of indoorswimming pools.

The value that can be generated from these aims stem from the recognition that currently, there are no published documents drawing together all the current DDS research. The literature review aims to contribute as independent research in this field and hopes to signpost the potential future direction ofDDS research.

2.LITERATURE SURVEY

2.1 Existing problem

The methodology of this review began with establishing a search plan. This involved generating a list of key search terms. As DDS are a global concept, it was important to consider

the various synonyms and acronyms under which they are known. The sources identified were then shortlisted according to relevance and reviewed, keeping in mind the potential for bias in market based literature, or literature drawing from funded research that may compromise its partiality. The literature review draws from a range of sources including standards documents from international and national bodies, reports, academic articles, books, online articles, and news reports.

To supplement this, a group of key stakeholders involved in pool safety were consulted to signpost towards any sources that were felt to be relevant tothis review. Sources on DDS in outdoor environments are beyond the scope of this literature review andhave been excluded due to the challenging variance compared with indoor swimming environments.

2.2 References

AngelEye. (2019). AngelEye – Distributors. Retrieved from: https://www.angeleye.it/news.php?id=28 HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28 HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"& HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28 HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28 HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"& HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"& HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28 HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"& HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28 HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"& HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"& HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28 HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"& HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28 HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28 HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"& HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"& HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28 HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"& HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"newscat=10" HYPERLINK "https://www.angeleye.it/news.php?

id=28&newscat=10"newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28 HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"& HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"newscat=10" HYPERLINK "https://www.angeleye.it/news.php?id=28&newscat=10"newscat=10 Aquatics International. (2007). Traumatic Experiences – Should we make our youngest lifeguards come face to face with death? Retrieved from: https://www.aquaticsintl.com/facilities/traumaticexperiences British Standards Institution. (2018). BS EN 15288-1, Swimming pools for public use. Safety requirements for design. Retrieved from: https://shop.bsigroup.com/ProductDetail/?pid=000000000030360254 British Standards Institution 1. (2018). BS EN 15288-2, Swimming pools for public use. Safety requirements for operation.

Retrieved from: https://shop.bsigroup.com/ProductDetail/?pid=000000000030360257 Drowning Prevention. (2017).

The Need. Retrieved from: https://www.drowningprevention.com.au/

German Institute for Standardization. (2019). German national guideline DGfdB R 94.15 “Test methods for camera-based drowning detection systems under operational conditions” (German Association for Public Swimming Pools).

Haizhou Li, Haizhou Li, Kar-Ann Toh and Liyuan Li. (2012). Advanced Topics in Biometrics, World Scientific Publishing Co. Pte. Ltd., ISBN-13 978-981-4287-84-5 Health and Safety Executive. (2018). HSG179, Health and safety in swimming pools (Fourth edition).

ISO (2017) ISO_20380, First edition, Public swimming pools — Computer vision systems for the detection of drowning accidents in swimming pools — Safety requirements and test methods.

J. Smith. (2016). Recognition, vigilance and surveillance techniques. The Science of Beach Lifeguarding,Pages 183 – 192.

2.3 Problem Statement Definition

The literature found that whilst there were plenty of academic articles dedicated to the technology and design behind DDS in the fields of biometrics, computer science and electronic engineering, there was limited academic research investigating their application to real-world scenarios. This highlighted a clear gap in research, identifying the need and importance for independent, real-world scenario based research.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Empathy maps are important because they give designers an avenue into the mind of the customer and help them empathize with their experience, desires, and needs. They're also useful for taking insights you gain from user research, digging deeper, and applying them to find concrete solutions.

- scope and goals.
- Gather materials. Your purpose should dictate the medium you use to create an empathy map.
- Collect research.
- Individually generate sticky notes for each quadrant.
- Converge to cluster and synthesize.
- Polish and plan.

3.2 Ideation & Brainstorming

- Begin with the main concept. First determine the main purpose of your mind map and write it down.
- branches to the main concept. Now that you have determined the main purpose of your mind map, add branches that will outline the most basic subtopics.
- Explore topics by adding more branches.
- images and colors.

3.3 Proposed Solution

1. Problem Statement-- Such kinds of deaths account for the thied cause of unplanned death globally,withabout1.2 million case yearly.To overcome thisconflict, a meticulous system is to be impelemented along the swimming pools to save human life.
2. Solution description-- Pulse oximeters make it possible to a drop in oxygen saturation, which may indicate the loss of consciousness under water and the initial stage of drowning. The addition of a small optical sensor to commonly used pool wristbands is a convenient solution both for swimming pool visitors and staff. However, no such devices available on the market can be treated as a“gold standard”.
3. Novelty / Uniqueness-- It helps the life guard to detect thr underwater situvation where they can't easily observe.
4. Customer Satisfaction- Expected to perform rescues to prevent drowings and to provide immediate first aid and CPR.

3.4 Problem Solution fit

1. Customer segment(s).
2. Jobs to be done/problems.
3. Tiggers.
4. Emotions--Lost and insecre/confident and in control.

5. Availabels solutions--Fire fighter and trained swimmers.
6. Customer constrains--spending power,budget,no cash,network connection,avaliable devices.
7. Behaviour--Install drowning detectors,or call for emergency help.
8. problem root cause.
9. Channels of Behaviour.
10. Your solution.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

I. Installation -- Needed to be fixed under the water without creating any disturbance to the people in the swimming pool.
II. Detection -- Detecting the human bodies and counting.
III. Audio -- Ask for help or stay quiet if the person is unconscious.
IV. Support --Take swim tubes or take the help of rescue.
V. Pulse Rate Sensor -- A pulse sensor on the that measures the changes in light absorption and reflection onto the skin to measure blood flow.

4.2 Non-Functional requirements

I. Usability --To ensure the safety of each and every person present in the pool. A Lifeguard should be present all the time in the pool.
II. Security-- Lifeguards should be aware of the alert message to save the life of the swimmer.
III. Reliability --Virtual eye lifeguard triggers an immediate prior alarm if a swimmer is in peril, helping to avoid panic even in critical situations.
IV. Performance-- The alarm is triggered when the swimmer's pulse rate is decreasing.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

1. Start
2. Yolo Model Algorithm
3. Sensed data from sensor

4. Pulse rate detection
5. Normal rate(continue monitoring)
6. high or low rate(send the alert lifegurad)
7. End

5.2 Solution & Technical Architecture

1. https://lh6.googleusercontent.com/9VZqfKlE0uRHVlvXmi6D_RuSa9lH9hnJHpKr3uQpprZ_xrbknQ3dM61nYhJzMGA2UU9cWTDuhz2JEG6QQl1adbtKEUtZ7wnRHZ5Bb0ocO8OvS8cRyiIqLZx0zftk-w

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

7. CODING & SOLUTIONING

7.1 Feature 1

```
input: "data"
input_shape {
  dim: 1
  dim: 3
  dim: 300
```

```
  dim: 300
}

layer {
  name: "data_bn"
  type: "BatchNorm"
  bottom: "data"
  top: "data_bn"
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
}
layer {
  name: "data_scale"
  type: "Scale"
  bottom: "data_bn"
  top: "data_bn"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 1.0
  }
  scale_param {
    bias_term: true
  }
}
layer {
  name: "conv1_h"
  type: "Convolution"
  bottom: "data_bn"
  top: "conv1_h"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
```

```
    decay_mult: 1.0
  }
  convolution_param {
    num_output: 32
    pad: 3
    kernel_size: 7
    stride: 2
    weight_filler {
      type: "msra"
      variance_norm: FAN_OUT
    }
    bias_filler {
      type: "constant"
      value: 0.0
    }
  }
}
layer {
  name: "conv1_bn_h"
  type: "BatchNorm"
  bottom: "conv1_h"
  top: "conv1_h"
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
}
layer {
  name: "conv1_scale_h"
  type: "Scale"
  bottom: "conv1_h"
  top: "conv1_h"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 1.0
  }
  scale_param {
```

```
    bias_term: true
  }
}
layer {
  name: "conv1_relu"
  type: "ReLU"
  bottom: "conv1_h"
  top: "conv1_h"
}
layer {
  name: "conv1_pool"
  type: "Pooling"
  bottom: "conv1_h"
  top: "conv1_pool"
  pooling_param {
    kernel_size: 3
    stride: 2
  }
}
layer {
  name: "layer_64_1_conv1_h"
  type: "Convolution"
  bottom: "conv1_pool"
  top: "layer_64_1_conv1_h"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  convolution_param {
    num_output: 32
    bias_term: false
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
      value: 0.0
    }
  }
}
layer {
  name: "layer_64_1_bn2_h"
  type: "BatchNorm"
```

```
  bottom: "layer_64_1_conv1_h"
  top: "layer_64_1_conv1_h"
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
}
layer {
  name: "layer_64_1_scale2_h"
  type: "Scale"
  bottom: "layer_64_1_conv1_h"
  top: "layer_64_1_conv1_h"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 1.0
  }
  scale_param {
    bias_term: true
  }
}
layer {
  name: "layer_64_1_relu2"
  type: "ReLU"
  bottom: "layer_64_1_conv1_h"
  top: "layer_64_1_conv1_h"
}
layer {
  name: "layer_64_1_conv2_h"
  type: "Convolution"
  bottom: "layer_64_1_conv1_h"
  top: "layer_64_1_conv2_h"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  convolution_param {
    num_output: 32
```

```
    bias_term: false
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
      value: 0.0
    }
  }
}
layer {
  name: "layer_64_1_sum"
  type: "Eltwise"
  bottom: "layer_64_1_conv2_h"
  bottom: "conv1_pool"
  top: "layer_64_1_sum"
}
layer {
  name: "layer_128_1_bn1_h"
  type: "BatchNorm"
  bottom: "layer_64_1_sum"
  top: "layer_128_1_bn1_h"
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
}
layer {
  name: "layer_128_1_scale1_h"
  type: "Scale"
  bottom: "layer_128_1_bn1_h"
  top: "layer_128_1_bn1_h"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
```

```
    decay_mult: 1.0
  }
  scale_param {
    bias_term: true
  }
}
layer {
  name: "layer_128_1_relu1"
  type: "ReLU"
  bottom: "layer_128_1_bn1_h"
  top: "layer_128_1_bn1_h"
}
layer {
  name: "layer_128_1_conv1_h"
  type: "Convolution"
  bottom: "layer_128_1_bn1_h"
  top: "layer_128_1_conv1_h"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  convolution_param {
    num_output: 128
    bias_term: false
    pad: 1
    kernel_size: 3
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
      value: 0.0
    }
  }
}
layer {
  name: "layer_128_1_bn2"
  type: "BatchNorm"
  bottom: "layer_128_1_conv1_h"
  top: "layer_128_1_conv1_h"
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
```

```
  }
  param {
    lr_mult: 0.0
  }
}
layer {
  name: "layer_128_1_scale2"
  type: "Scale"
  bottom: "layer_128_1_conv1_h"
  top: "layer_128_1_conv1_h"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 1.0
  }
  scale_param {
    bias_term: true
  }
}
layer {
  name: "layer_128_1_relu2"
  type: "ReLU"
  bottom: "layer_128_1_conv1_h"
  top: "layer_128_1_conv1_h"
}
layer {
  name: "layer_128_1_conv2"
  type: "Convolution"
  bottom: "layer_128_1_conv1_h"
  top: "layer_128_1_conv2"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  convolution_param {
    num_output: 128
    bias_term: false
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
```

```
    bias_filler {
      type: "constant"
      value: 0.0
    }
  }
}
layer {
  name: "layer_128_1_conv_expand_h"
  type: "Convolution"
  bottom: "layer_128_1_bn1_h"
  top: "layer_128_1_conv_expand_h"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  convolution_param {
    num_output: 128
    bias_term: false
    pad: 0
    kernel_size: 1
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
      value: 0.0
    }
  }
}
layer {
  name: "layer_128_1_sum"
  type: "Eltwise"
  bottom: "layer_128_1_conv2"
  bottom: "layer_128_1_conv_expand_h"
  top: "layer_128_1_sum"
}
layer {
  name: "layer_256_1_bn1"
  type: "BatchNorm"
  bottom: "layer_128_1_sum"
  top: "layer_256_1_bn1"
  param {
    lr_mult: 0.0
  }
  param {
```

```
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
}
layer {
  name: "layer_256_1_scale1"
  type: "Scale"
  bottom: "layer_256_1_bn1"
  top: "layer_256_1_bn1"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 1.0
  }
  scale_param {
    bias_term: true
  }
}
layer {
  name: "layer_256_1_relu1"
  type: "ReLU"
  bottom: "layer_256_1_bn1"
  top: "layer_256_1_bn1"
}
layer {
  name: "layer_256_1_conv1"
  type: "Convolution"
  bottom: "layer_256_1_bn1"
  top: "layer_256_1_conv1"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  convolution_param {
    num_output: 256
    bias_term: false
    pad: 1
    kernel_size: 3
    stride: 2
    weight_filler {
      type: "msra"
```

```
    }
    bias_filler {
      type: "constant"
      value: 0.0
    }
  }
}
layer {
  name: "layer_256_1_bn2"
  type: "BatchNorm"
  bottom: "layer_256_1_conv1"
  top: "layer_256_1_conv1"
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
}
layer {
  name: "layer_256_1_scale2"
  type: "Scale"
  bottom: "layer_256_1_conv1"
  top: "layer_256_1_conv1"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 1.0
  }
  scale_param {
    bias_term: true
  }
}
layer {
  name: "layer_256_1_relu2"
  type: "ReLU"
  bottom: "layer_256_1_conv1"
  top: "layer_256_1_conv1"
}
layer {
```

```
  name: "layer_256_1_conv2"
  type: "Convolution"
  bottom: "layer_256_1_conv1"
  top: "layer_256_1_conv2"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  convolution_param {
    num_output: 256
    bias_term: false
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
      value: 0.0
    }
  }
}
layer {
  name: "layer_256_1_conv_expand"
  type: "Convolution"
  bottom: "layer_256_1_bn1"
  top: "layer_256_1_conv_expand"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  convolution_param {
    num_output: 256
    bias_term: false
    pad: 0
    kernel_size: 1
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
      value: 0.0
    }
  }
```

```
}
layer {
  name: "layer_256_1_sum"
  type: "Eltwise"
  bottom: "layer_256_1_conv2"
  bottom: "layer_256_1_conv_expand"
  top: "layer_256_1_sum"
}
layer {
  name: "layer_512_1_bn1"
  type: "BatchNorm"
  bottom: "layer_256_1_sum"
  top: "layer_512_1_bn1"
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
}
layer {
  name: "layer_512_1_scale1"
  type: "Scale"
  bottom: "layer_512_1_bn1"
  top: "layer_512_1_bn1"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 1.0
  }
  scale_param {
    bias_term: true
  }
}
layer {
  name: "layer_512_1_relu1"
  type: "ReLU"
  bottom: "layer_512_1_bn1"
  top: "layer_512_1_bn1"
}
```

```
layer {
  name: "layer_512_1_conv1_h"
  type: "Convolution"
  bottom: "layer_512_1_bn1"
  top: "layer_512_1_conv1_h"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  convolution_param {
    num_output: 128
    bias_term: false
    pad: 1
    kernel_size: 3
    stride: 1 # 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
      value: 0.0
    }
  }
}
layer {
  name: "layer_512_1_bn2_h"
  type: "BatchNorm"
  bottom: "layer_512_1_conv1_h"
  top: "layer_512_1_conv1_h"
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
}
layer {
  name: "layer_512_1_scale2_h"
  type: "Scale"
  bottom: "layer_512_1_conv1_h"
  top: "layer_512_1_conv1_h"
  param {
    lr_mult: 1.0
```

```
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 1.0
  }
  scale_param {
    bias_term: true
  }
}
layer {
  name: "layer_512_1_relu2"
  type: "ReLU"
  bottom: "layer_512_1_conv1_h"
  top: "layer_512_1_conv1_h"
}
layer {
  name: "layer_512_1_conv2_h"
  type: "Convolution"
  bottom: "layer_512_1_conv1_h"
  top: "layer_512_1_conv2_h"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  convolution_param {
    num_output: 256
    bias_term: false
    pad: 2 # 1
    kernel_size: 3
    stride: 1
    dilation: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
      value: 0.0
    }
  }
}
layer {
  name: "layer_512_1_conv_expand_h"
  type: "Convolution"
  bottom: "layer_512_1_bn1"
  top: "layer_512_1_conv_expand_h"
```

```
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  convolution_param {
    num_output: 256
    bias_term: false
    pad: 0
    kernel_size: 1
    stride: 1 # 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
      value: 0.0
    }
  }
}
layer {
  name: "layer_512_1_sum"
  type: "Eltwise"
  bottom: "layer_512_1_conv2_h"
  bottom: "layer_512_1_conv_expand_h"
  top: "layer_512_1_sum"
}
layer {
  name: "last_bn_h"
  type: "BatchNorm"
  bottom: "layer_512_1_sum"
  top: "layer_512_1_sum"
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
  param {
    lr_mult: 0.0
  }
}
layer {
  name: "last_scale_h"
  type: "Scale"
  bottom: "layer_512_1_sum"
  top: "layer_512_1_sum"
```

```
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 1.0
  }
  scale_param {
    bias_term: true
  }
}
layer {
  name: "last_relu"
  type: "ReLU"
  bottom: "layer_512_1_sum"
  top: "fc7"
}

layer {
  name: "conv6_1_h"
  type: "Convolution"
  bottom: "fc7"
  top: "conv6_1_h"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 128
    pad: 0
    kernel_size: 1
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
```

```
layer {
  name: "conv6_1_relu"
  type: "ReLU"
  bottom: "conv6_1_h"
  top: "conv6_1_h"
}
layer {
  name: "conv6_2_h"
  type: "Convolution"
  bottom: "conv6_1_h"
  top: "conv6_2_h"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 256
    pad: 1
    kernel_size: 3
    stride: 2
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv6_2_relu"
  type: "ReLU"
  bottom: "conv6_2_h"
  top: "conv6_2_h"
}
layer {
  name: "conv7_1_h"
  type: "Convolution"
  bottom: "conv6_2_h"
  top: "conv7_1_h"
  param {
    lr_mult: 1
```

```
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 64
    pad: 0
    kernel_size: 1
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv7_1_relu"
  type: "ReLU"
  bottom: "conv7_1_h"
  top: "conv7_1_h"
}
layer {
  name: "conv7_2_h"
  type: "Convolution"
  bottom: "conv7_1_h"
  top: "conv7_2_h"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 128
    pad: 1
    kernel_size: 3
    stride: 2
    weight_filler {
      type: "xavier"
```

```
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv7_2_relu"
  type: "ReLU"
  bottom: "conv7_2_h"
  top: "conv7_2_h"
}
layer {
  name: "conv8_1_h"
  type: "Convolution"
  bottom: "conv7_2_h"
  top: "conv8_1_h"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 64
    pad: 0
    kernel_size: 1
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv8_1_relu"
  type: "ReLU"
  bottom: "conv8_1_h"
  top: "conv8_1_h"
}
```

```
layer {
  name: "conv8_2_h"
  type: "Convolution"
  bottom: "conv8_1_h"
  top: "conv8_2_h"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 128
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv8_2_relu"
  type: "ReLU"
  bottom: "conv8_2_h"
  top: "conv8_2_h"
}
layer {
  name: "conv9_1_h"
  type: "Convolution"
  bottom: "conv8_2_h"
  top: "conv9_1_h"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
```

```
  convolution_param {
    num_output: 64
    pad: 0
    kernel_size: 1
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv9_1_relu"
  type: "ReLU"
  bottom: "conv9_1_h"
  top: "conv9_1_h"
}
layer {
  name: "conv9_2_h"
  type: "Convolution"
  bottom: "conv9_1_h"
  top: "conv9_2_h"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 128
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
```

```
}
layer {
  name: "conv9_2_relu"
  type: "ReLU"
  bottom: "conv9_2_h"
  top: "conv9_2_h"
}
layer {
  name: "conv4_3_norm"
  type: "Normalize"
  bottom: "layer_256_1_bn1"
  top: "conv4_3_norm"
  norm_param {
    across_spatial: false
    scale_filler {
      type: "constant"
      value: 20
    }
    channel_shared: false
  }
}
layer {
  name: "conv4_3_norm_mbox_loc"
  type: "Convolution"
  bottom: "conv4_3_norm"
  top: "conv4_3_norm_mbox_loc"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 16
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
```

```
  }
}
layer {
  name: "conv4_3_norm_mbox_loc_perm"
  type: "Permute"
  bottom: "conv4_3_norm_mbox_loc"
  top: "conv4_3_norm_mbox_loc_perm"
  permute_param {
    order: 0
    order: 2
    order: 3
    order: 1
  }
}
layer {
  name: "conv4_3_norm_mbox_loc_flat"
  type: "Flatten"
  bottom: "conv4_3_norm_mbox_loc_perm"
  top: "conv4_3_norm_mbox_loc_flat"
  flatten_param {
    axis: 1
  }
}
layer {
  name: "conv4_3_norm_mbox_conf"
  type: "Convolution"
  bottom: "conv4_3_norm"
  top: "conv4_3_norm_mbox_conf"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 8 # 84
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
```

```
      value: 0
    }
  }
}
layer {
  name: "conv4_3_norm_mbox_conf_perm"
  type: "Permute"
  bottom: "conv4_3_norm_mbox_conf"
  top: "conv4_3_norm_mbox_conf_perm"
  permute_param {
    order: 0
    order: 2
    order: 3
    order: 1
  }
}
layer {
  name: "conv4_3_norm_mbox_conf_flat"
  type: "Flatten"
  bottom: "conv4_3_norm_mbox_conf_perm"
  top: "conv4_3_norm_mbox_conf_flat"
  flatten_param {
    axis: 1
  }
}
layer {
  name: "conv4_3_norm_mbox_priorbox"
  type: "PriorBox"
  bottom: "conv4_3_norm"
  bottom: "data"
  top: "conv4_3_norm_mbox_priorbox"
  prior_box_param {
    min_size: 30.0
    max_size: 60.0
    aspect_ratio: 2
    flip: true
    clip: false
    variance: 0.1
    variance: 0.1
    variance: 0.2
    variance: 0.2
    step: 8
    offset: 0.5
  }
}
layer {
```

```
  name: "fc7_mbox_loc"
  type: "Convolution"
  bottom: "fc7"
  top: "fc7_mbox_loc"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 24
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "fc7_mbox_loc_perm"
  type: "Permute"
  bottom: "fc7_mbox_loc"
  top: "fc7_mbox_loc_perm"
  permute_param {
    order: 0
    order: 2
    order: 3
    order: 1
  }
}
layer {
  name: "fc7_mbox_loc_flat"
  type: "Flatten"
  bottom: "fc7_mbox_loc_perm"
  top: "fc7_mbox_loc_flat"
  flatten_param {
    axis: 1
  }
```

```
}
layer {
  name: "fc7_mbox_conf"
  type: "Convolution"
  bottom: "fc7"
  top: "fc7_mbox_conf"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 12 # 126
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "fc7_mbox_conf_perm"
  type: "Permute"
  bottom: "fc7_mbox_conf"
  top: "fc7_mbox_conf_perm"
  permute_param {
    order: 0
    order: 2
    order: 3
    order: 1
  }
}
layer {
  name: "fc7_mbox_conf_flat"
  type: "Flatten"
  bottom: "fc7_mbox_conf_perm"
  top: "fc7_mbox_conf_flat"
  flatten_param {
```

```
    axis: 1
  }
}
layer {
  name: "fc7_mbox_priorbox"
  type: "PriorBox"
  bottom: "fc7"
  bottom: "data"
  top: "fc7_mbox_priorbox"
  prior_box_param {
    min_size: 60.0
    max_size: 111.0
    aspect_ratio: 2
    aspect_ratio: 3
    flip: true
    clip: false
    variance: 0.1
    variance: 0.1
    variance: 0.2
    variance: 0.2
    step: 16
    offset: 0.5
  }
}
layer {
  name: "conv6_2_mbox_loc"
  type: "Convolution"
  bottom: "conv6_2_h"
  top: "conv6_2_mbox_loc"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 24
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
```

```
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv6_2_mbox_loc_perm"
  type: "Permute"
  bottom: "conv6_2_mbox_loc"
  top: "conv6_2_mbox_loc_perm"
  permute_param {
    order: 0
    order: 2
    order: 3
    order: 1
  }
}
layer {
  name: "conv6_2_mbox_loc_flat"
  type: "Flatten"
  bottom: "conv6_2_mbox_loc_perm"
  top: "conv6_2_mbox_loc_flat"
  flatten_param {
    axis: 1
  }
}
layer {
  name: "conv6_2_mbox_conf"
  type: "Convolution"
  bottom: "conv6_2_h"
  top: "conv6_2_mbox_conf"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 12 # 126
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
```

```
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv6_2_mbox_conf_perm"
  type: "Permute"
  bottom: "conv6_2_mbox_conf"
  top: "conv6_2_mbox_conf_perm"
  permute_param {
    order: 0
    order: 2
    order: 3
    order: 1
  }
}
layer {
  name: "conv6_2_mbox_conf_flat"
  type: "Flatten"
  bottom: "conv6_2_mbox_conf_perm"
  top: "conv6_2_mbox_conf_flat"
  flatten_param {
    axis: 1
  }
}
layer {
  name: "conv6_2_mbox_priorbox"
  type: "PriorBox"
  bottom: "conv6_2_h"
  bottom: "data"
  top: "conv6_2_mbox_priorbox"
  prior_box_param {
    min_size: 111.0
    max_size: 162.0
    aspect_ratio: 2
    aspect_ratio: 3
    flip: true
    clip: false
    variance: 0.1
    variance: 0.1
    variance: 0.2
    variance: 0.2
    step: 32
```

```
    offset: 0.5
  }
}
layer {
  name: "conv7_2_mbox_loc"
  type: "Convolution"
  bottom: "conv7_2_h"
  top: "conv7_2_mbox_loc"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 24
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv7_2_mbox_loc_perm"
  type: "Permute"
  bottom: "conv7_2_mbox_loc"
  top: "conv7_2_mbox_loc_perm"
  permute_param {
    order: 0
    order: 2
    order: 3
    order: 1
  }
}
layer {
  name: "conv7_2_mbox_loc_flat"
  type: "Flatten"
  bottom: "conv7_2_mbox_loc_perm"
```

```
  top: "conv7_2_mbox_loc_flat"
  flatten_param {
    axis: 1
  }
}
layer {
  name: "conv7_2_mbox_conf"
  type: "Convolution"
  bottom: "conv7_2_h"
  top: "conv7_2_mbox_conf"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 12 # 126
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv7_2_mbox_conf_perm"
  type: "Permute"
  bottom: "conv7_2_mbox_conf"
  top: "conv7_2_mbox_conf_perm"
  permute_param {
    order: 0
    order: 2
    order: 3
    order: 1
  }
}
layer {
  name: "conv7_2_mbox_conf_flat"
```

```
  type: "Flatten"
  bottom: "conv7_2_mbox_conf_perm"
  top: "conv7_2_mbox_conf_flat"
  flatten_param {
    axis: 1
  }
}
layer {
  name: "conv7_2_mbox_priorbox"
  type: "PriorBox"
  bottom: "conv7_2_h"
  bottom: "data"
  top: "conv7_2_mbox_priorbox"
  prior_box_param {
    min_size: 162.0
    max_size: 213.0
    aspect_ratio: 2
    aspect_ratio: 3
    flip: true
    clip: false
    variance: 0.1
    variance: 0.1
    variance: 0.2
    variance: 0.2
    step: 64
    offset: 0.5
  }
}
layer {
  name: "conv8_2_mbox_loc"
  type: "Convolution"
  bottom: "conv8_2_h"
  top: "conv8_2_mbox_loc"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 16
    pad: 1
    kernel_size: 3
    stride: 1
```

```
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv8_2_mbox_loc_perm"
  type: "Permute"
  bottom: "conv8_2_mbox_loc"
  top: "conv8_2_mbox_loc_perm"
  permute_param {
    order: 0
    order: 2
    order: 3
    order: 1
  }
}
layer {
  name: "conv8_2_mbox_loc_flat"
  type: "Flatten"
  bottom: "conv8_2_mbox_loc_perm"
  top: "conv8_2_mbox_loc_flat"
  flatten_param {
    axis: 1
  }
}
layer {
  name: "conv8_2_mbox_conf"
  type: "Convolution"
  bottom: "conv8_2_h"
  top: "conv8_2_mbox_conf"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 8 # 84
    pad: 1
```

```
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv8_2_mbox_conf_perm"
  type: "Permute"
  bottom: "conv8_2_mbox_conf"
  top: "conv8_2_mbox_conf_perm"
  permute_param {
    order: 0
    order: 2
    order: 3
    order: 1
  }
}
layer {
  name: "conv8_2_mbox_conf_flat"
  type: "Flatten"
  bottom: "conv8_2_mbox_conf_perm"
  top: "conv8_2_mbox_conf_flat"
  flatten_param {
    axis: 1
  }
}
layer {
  name: "conv8_2_mbox_priorbox"
  type: "PriorBox"
  bottom: "conv8_2_h"
  bottom: "data"
  top: "conv8_2_mbox_priorbox"
  prior_box_param {
    min_size: 213.0
    max_size: 264.0
    aspect_ratio: 2
    flip: true
    clip: false
    variance: 0.1
    variance: 0.1
```

```
    variance: 0.2
    variance: 0.2
    step: 100
    offset: 0.5
  }
}
layer {
  name: "conv9_2_mbox_loc"
  type: "Convolution"
  bottom: "conv9_2_h"
  top: "conv9_2_mbox_loc"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 16
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv9_2_mbox_loc_perm"
  type: "Permute"
  bottom: "conv9_2_mbox_loc"
  top: "conv9_2_mbox_loc_perm"
  permute_param {
    order: 0
    order: 2
    order: 3
    order: 1
  }
}
layer {
```

```
  name: "conv9_2_mbox_loc_flat"
  type: "Flatten"
  bottom: "conv9_2_mbox_loc_perm"
  top: "conv9_2_mbox_loc_flat"
  flatten_param {
    axis: 1
  }
}
layer {
  name: "conv9_2_mbox_conf"
  type: "Convolution"
  bottom: "conv9_2_h"
  top: "conv9_2_mbox_conf"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 8 # 84
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
      value: 0
    }
  }
}
layer {
  name: "conv9_2_mbox_conf_perm"
  type: "Permute"
  bottom: "conv9_2_mbox_conf"
  top: "conv9_2_mbox_conf_perm"
  permute_param {
    order: 0
    order: 2
    order: 3
    order: 1
  }
```

```
}
layer {
  name: "conv9_2_mbox_conf_flat"
  type: "Flatten"
  bottom: "conv9_2_mbox_conf_perm"
  top: "conv9_2_mbox_conf_flat"
  flatten_param {
    axis: 1
  }
}
layer {
  name: "conv9_2_mbox_priorbox"
  type: "PriorBox"
  bottom: "conv9_2_h"
  bottom: "data"
  top: "conv9_2_mbox_priorbox"
  prior_box_param {
    min_size: 264.0
    max_size: 315.0
    aspect_ratio: 2
    flip: true
    clip: false
    variance: 0.1
    variance: 0.1
    variance: 0.2
    variance: 0.2
    step: 300
    offset: 0.5
  }
}
layer {
  name: "mbox_loc"
  type: "Concat"
  bottom: "conv4_3_norm_mbox_loc_flat"
  bottom: "fc7_mbox_loc_flat"
  bottom: "conv6_2_mbox_loc_flat"
  bottom: "conv7_2_mbox_loc_flat"
  bottom: "conv8_2_mbox_loc_flat"
  bottom: "conv9_2_mbox_loc_flat"
  top: "mbox_loc"
  concat_param {
    axis: 1
  }
}
layer {
  name: "mbox_conf"
```

```
  type: "Concat"
  bottom: "conv4_3_norm_mbox_conf_flat"
  bottom: "fc7_mbox_conf_flat"
  bottom: "conv6_2_mbox_conf_flat"
  bottom: "conv7_2_mbox_conf_flat"
  bottom: "conv8_2_mbox_conf_flat"
  bottom: "conv9_2_mbox_conf_flat"
  top: "mbox_conf"
  concat_param {
    axis: 1
  }
}
layer {
  name: "mbox_priorbox"
  type: "Concat"
  bottom: "conv4_3_norm_mbox_priorbox"
  bottom: "fc7_mbox_priorbox"
  bottom: "conv6_2_mbox_priorbox"
  bottom: "conv7_2_mbox_priorbox"
  bottom: "conv8_2_mbox_priorbox"
  bottom: "conv9_2_mbox_priorbox"
  top: "mbox_priorbox"
  concat_param {
    axis: 2
  }
}

layer {
  name: "mbox_conf_reshape"
  type: "Reshape"
  bottom: "mbox_conf"
  top: "mbox_conf_reshape"
  reshape_param {
    shape {
      dim: 0
      dim: -1
      dim: 2
    }
  }
}
layer {
  name: "mbox_conf_softmax"
  type: "Softmax"
  bottom: "mbox_conf_reshape"
  top: "mbox_conf_softmax"
  softmax_param {
```

```
    axis: 2
  }
}
layer {
  name: "mbox_conf_flatten"
  type: "Flatten"
  bottom: "mbox_conf_softmax"
  top: "mbox_conf_flatten"
  flatten_param {
    axis: 1
  }
}

layer {
  name: "detection_out"
  type: "DetectionOutput"
  bottom: "mbox_loc"
  bottom: "mbox_conf_flatten"
  bottom: "mbox_priorbox"
  top: "detection_out"
  include {
    phase: TEST
  }
  detection_output_param {
    num_classes: 2
    share_location: true
    background_label_id: 0
    nms_param {
      nms_threshold: 0.45
      top_k: 400
    }
    code_type: CENTER_SIZE
    keep_top_k: 200
    confidence_threshold: 0.01
  }
}
```

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

- 1. Using a tape measure, the number of testing lanes across the width of the pool were marked out using cones, and the same repeated across the pool length to clearly mark out the test areas for the detectiontest. Between the differing pool sizes, this resulted in 42-54 test areas per pool.
- Using alternative coloured cones from step 1, the test area for the non-detection test was marked out in the shallow end (<1.5m) of the pool to mark out the 5 non-detection test positions.
- A light reading was taken and recorded at all four corners of the pool using a lux meter.
- The test start time and pool orientation was recorded into the testing document.
- Next the non-detection test was carried out. An assistant stood stationary in each of the 5 test positions in the empty pool for 40 seconds, timed using a stopwatch. If the alarm was activated by the
- stationary swimmer within the 40 seconds, this was recorded as a fail. If the alarm did not react for the 40 second duration, this was recorded as a pass.
- During the non-detection test, the system was observed to check whether the system's reaction was due to the assistant in the pool rather than a false alarm detection, such as detecting darker pool tiles.
- The standard orange lifeguard training dummy was used and prepared by dressing it in a woman's dark coloured swimming costume and tying a rope with measured markings around its neck to help guide the dummy into each test position.
- Next the detection test was carried out. The dummy was placed in the first test area of the test lane in the empty pool and submerged.
- Once the dummy was laid stationary on the pool floor, a stopwatch was used to time and record the number of seconds until the alarm reacted.
- The system monitor was observed to ensure the alarm was triggered by the dummy rather than a false alarm reaction and if reactive, the alarm then dismissed. If the system did not react within 30 seconds, this was recorded as a non-detection within this time.
- Other observations such as whether the dummy rested on the darker tiles of the pool floor were also recorded. To calculate a fair number of tests to be conducted on dark pool floor tiles, the proportion of dark tiles on the pool floor were calculated as a percentage of the total pool floor surface area, and 10% of this was used to determine the number of dark tile tests to conduct.
- Once the alarm was dismissed or a non-detection recorded, the rope was pulled from the opposite end of the lane, pulling the dummy into the next test area and the procedure repeated until all test areas of the pool were tested and recorded.
- Next, the dummy was placed in each corner of the pool and the procedure repeated to record the system's performance in each corner.
- Testing was also carried out to identify whether the system was able to detect simultaneous incidents.This was done by submerging two standard dummies in different areas of the pool at the same time.
- The test finish time was recorded into the document

9. RESULTS

Results.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Title</title>
</head>
<body>
<h1>result</h1>
</body>
</html>
```

9.1 Performance Metrics

The supplier is responsible for providing a manual to the pool operators, with the following outlined in ISO_20380 as a minimum requirement: 'supplier reference number of the computer vision system installed; how to operate the system; a detailed description of the performances and limits of the system; a description of the performance measurement procedures to be carried out for regular testing; supplier contact details and contact details of the help line (opening hours, telephone number and e-mail address)'. Training must be provided by the supplier following installation of the DDS before it is used by the operator for the first time.

10. ADVANTAGES

- This system don't have to wait until life guard comes to rescue because it has uplifting mesh.
- This is very fast process
- More effecttive and cost Efficient than previous other models.
- spending more time for family, freedom for safety guards near the
- Swimming pool
- Swimmers, resort are gain in the financal
- drowning should be monitored

DISADVANTAGES

- Internet connection is necessary to use gps or sending alert messages. somtimes to send message SIM balance may be required.
- For uneducated people will suffer from this technology
- Electricity will be required
- Software and hardware requirement will need

11. CONCLUSION

This literature review has discussed the various complexities of DDS within the health and safety landscape, as well as the wider implications of their use on the sport and leisure industry. It has also shed light on needed for more evidence in this area. From reviewing what literature currently exists on the topic, it is clear the evidence-base would benefit from qualitative research on the experiences of lifeguards and their interactions with DDS, as well as quantitative evidence showing DDS application to real-world scenarios. Claims expressing the risks of DDS negatively affecting lifeguarding performance should also be further investigated, and efforts made across the industry to ensure all publicly available
information and guidance surrounding DDS is current and up-to-date. The Drowning Detection System Briefing note was published before documents such which is periodically updated, and operators should ensure that the sources they are using for DDS research do not draw from

predated editions of health and safety law and guidance. Again, co-operation is required between all with an interest in the improvement of pool safety, to share data, information and learning on DDS, including but not restricted to results and findings from any DDS standards tests carried out.

12. FUTURE SCOPE

This lifeguard system consists of three main components, i.e., the drowning detection, the rescuing drone, and the hazardous activity detection. All three components combined will create a system capable of detecting drowning victims, dispatching an inflatable tube using a drone (as depicted in Fig.9) and detecting hazardous activities—eventually becoming an entity that could assist a lifeguard. The system is accessible to its primary user, presumably a pool owner or a lifeguard, in the form of an interface with a sound alarm and an android mobile service that holds the capabilities of receiving Firebase notifications. Confined with a few of the hardware limitations, such as the use of a single camera and the Jetson Nano at the presence of better-quality hardware, could affect the speed and accuracy of the overall system is becoming a state-of-theart. This limitation could be omitted with the use of multiple cameras that could be placed over the premises in several ground coordinates, increasing the accuracy of the computer vision algorithms. Moreover, due to the inability to fly a drone in extreme weather conditions such as rain, strong winds or lightning, the system is limited to be used under few specifications. As swimming in extreme weather conditions is not preferred either, the system could be further improved to emit a warning signal if a person was to swim in any of the above weather conditions, bypassing the need to fly the drone. Additionally, all the processing is done on the clientside of the applications on the Jetson Nano board, preventing any security and privacy issues that might arise due to the sensitive information inputted through the cameras. For future developments convenience wise, the system could benefit by having an additional set of cameras to identify and verify a drowning or a hazardous activity on the premises. Accessibility could also be improved by extending the Android service to be an application both in Android and iOS platforms that could hold the details of each premise individually, making a centralized system that watches over the decentralized pool premises. Both drown and hazardous activity detection could be improved by gathering a night time dataset that increases the accuracy of the data in low light.

13. APPENDIX

Source Code

Login HTMl

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login & Register</title>
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.3/css/all.css">
```

```
</head>
<body>
  <div class="container" id="container">
    <div class="form-container sign-up-container">
      <form action="#">
        <h1>Sign Up</h1>
        <div class="social-container">
          <a href="#" class="social"><i class="fab fa-facebook-f"></i></a>
          <a href="#" class="social"><i class="fab fa-google-plus-g"></i></a>
          <a href="#" class="social"><i class="fab fa-linkedin-in"></i></a>
        </div>
        <span>or use your email for registration</span>
        <input type="text" placeholder="Username">
        <input type="email" placeholder="Email">
        <input type="password" placeholder="Password">
        <button>Sign Up</button>
      </form>
    </div>
    <div class="form-container sign-in-container">
      <form action="#">
        <h1>Sign In</h1>
        <div class="social-container">
          <a href="#" class="social"><i class="fab fa-facebook-f"></i></a>
          <a href="#" class="social"><i class="fab fa-google-plus-g"></i></a>
          <a href="#" class="social"><i class="fab fa-linkedin-in"></i></a>
        </div>
        <span>or use your account</span>
        <input type="email" placeholder="Email">
        <input type="password" placeholder="Password">
        <button>Sign In</button>
      </form>
    </div>
    <div class="overlay-container">
      <div class="overlay">
        <div class="overlay-panel overlay-left">
          <h1>Sample Text</h1>
          <p>Sample Text</p>
          <button class="press" id="signIn">Sign In</button>
        </div>
        <div class="overlay-panel overlay-right">
          <h1>Sample Text</h1>
          <p>Sample Text</p>
          <button class="press" id="signUp">Sign Up</button>
        </div>
      </div>
    </div>
```

```
        </div>

        <script>
            const signUpButton = document.getElementById("signUp");
            const signInButton = document.getElementById("signIn");
            const container = document.getElementById("container");

            signUpButton.addEventListener('click',()=>{
                container.classList.add("right-panel-active");
            })

            signInButton.addEventListener('click',()=>{
                container.classList.remove("right-panel-active");
            })
        </script>
    </body>
</html>
```

Login CSS

```
@import url('https://fonts.googleapis.com/css?family=Montserrat:400,800');

*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
body{
    background-color: #f6f5f7;
    display: flex;
    justify-content: center;
    align-items: center;
    font-family: 'Montserrat',sans-serif;
    height: 100vh;
    margin: 0 auto;
}
.container{
    background-color: #fff;
    border-radius: 10px;
    width: 768px;
    max-width: 100%;
    min-height: 480px;
    position: relative;
    overflow: hidden;
    box-shadow: 0px 14px 28px rgba(0, 0, 0, 0.25),
            0px 10px 10px rgba(0, 0, 0, 0.22);
}
.form-container{
    position: absolute;
    top: 0;
    height: 100%;
    transition: all 0.6 ease-in-out;
```

```
}
.sign-up-container{
    left: 0;
    width: 50%;
    z-index: 1;
    opacity: 0;
}
form{
    background-color: #FFFFFF;
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    padding: 0 50px;
    height: 100%;
    text-align: center;
}
h1{
    font-weight: bold;
    margin: 0;
}
.social-container{
    margin: 20px 0;
}
.social-container a{
    border: 1px solid gray;
    border-radius: 50%;
    display: inline-flex;
    justify-content: center;
    align-items: center;
    margin: 0 5px;
    height: 40px;
    width: 40px;
}
a{
    color: #333;
    font-size: 14px;
    text-decoration: none;
    margin: 15px 0;
}
span{
    font-size: 12px;
}
input{
    background-color: #eee;
    border: none;
    text-decoration: none;
    margin: 8px 0;
    padding: 12px 15px;
    width: 100%;
}
button{
    border-radius: 20px;
    border: 1px solid #FF4B2B;
```

```
    background-color: #FF4B2B;
    color: #FFFFFF;
    font-size: 12px;
    font-weight: bold;
    padding: 12px 45px;
    letter-spacing: 1px;
    text-transform: uppercase;
    transition: transform .8s ease-in;
}
.sign-in-container{
    left: 0;
    width: 50%;
    z-index: 2;
}
.overlay-container{
    position: absolute;
    top: 0;
    left: 50%;
    width: 50%;
    height: 100%;
    overflow: hidden;
    z-index: 100;
    transition: transform 0.6s ease-in-out;
}
.overlay{
    background-color: #FF416C;
    background: linear-gradient(to right,#FF4B2B,#FF416C);
    background-repeat: no-repeat;
    background-size: cover;
    background-position: 0 0;
    color: #FFFFFF;
    position: relative;
    left: -100%;
    height: 100%;
    width: 200%;
    transform: translateX(0);
    transition: transform .6s ease-in-out;
}
.overlay-panel{
    position: absolute;
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    padding: 0px 40px;
    text-align: center;
    top: 0;
    height: 100%;
    width: 50%;
    transform: translateX(0);
    transition: transform .6s ease-in-out;
}
.overlay-left{
    transform: translateX(-20%);
```

```
}
.overlay-right{
    right: 0;
    transform: translateX(0);
}
button:active{
    transform: scale(.95);
}
button.press{
    background-color: transparent;
    border-color: #FFFFFF;
}
p{
    font-size: 14px;
    font-weight: 100;
    line-height: 20px;
    letter-spacing: .5;
    margin: 20px 0 30px;
}
.container.right-panel-active .sign-in-container{
    transform: translateX(100%);
}
.container.right-panel-active .sign-up-container{
    transform: translateX(100%);
    opacity: 1;
    z-index: 5;
    animation: slide 0.6s;
}
@keyframes slide{
    0%,49.99%{
        opacity: 0;
        z-index: 1;
    }
    50%,100%{
        opacity: 1;
        z-index: 5;
    }
}
.container.right-panel-active .overlay-container{
    transform: translateX(-100%);
}
.container.right-panel-active .overlay{
    transform: translateX(50%);
}
.container.right-panel-active .overlay-left{
    transform: translateX(0);
}
.container.right-panel-active .overlay-right{
    transform: translateX(20%);
}
```

Index.HTML

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>VirtualEye|Home</title>
   <!-- ==== CSS ==== -->
   <link rel="stylesheet" href="./style.css">
</head>
<body>
   <!-- Navbar-->
   <nav class="navbar">
      <h1 class="logo">VirtualEye</h1>
      <ul class="nav-links">
         <li class="active"><a href="#"></a>Home</li>
         <li><a href="#"></a>Login</li>
         <li><a href="#"></a>Register</li>
         <li><a href="#"></a>Demo</li>
      </ul>
      <img src="./Images/toggle.png" alt="" class="menu-btn">
   </nav>

   <header>
      <div class="header-content">
         <h2>VirtualEye saves lives</h2>
         <div class="line"></div>
         <h1>Intelligent detection system for any pool</h1>
         <a href="#" class="ctn">Learn More</a>
      </div>
   </header>

   <!--== About Project == -->
   <section class="events">
      <div class="title">
         <h1>About Project</h1>
         <div class="line"></div>
      </div>
      <div class="row">
         <div class="col">
            <h4>Problem</h4>
            <p>Swimming is one of the best exercises that helps people to reduce stress in this urban lifestyle. Swimming pools are found larger in number in hotels, and weekend tourist spots and barely people have them in their house backyard. Beginners, especially, often feel it difficult to breathe underwater which causes breathing trouble which in turn causes a drowning accident. Worldwide, drowning produces a higher rate of mortality without causing injury to children. Children under six of their age are found to be suffering the highest drowning mortality rates worldwide. Such kinds of deaths account for the third cause of unplanned death globally, with about 1.2  million cases yearly.</p>
```

```
            <a href="#" class="ctn">Learn More</a>
          </div>
          <div class="col">
            <h4>Solution</h4>
            <p>To overcome this conflict, a meticulous system is to be implemented along the swimming pools to save human life.By studying body movement patterns and connecting cameras to artificial intelligence (AI) systems we can devise an underwater pool safety system that reduces the risk of drowning.  Usually, such systems can be developed by installing more than 16 cameras underwater and ceiling and analyzing the video feeds to detect any anomalies. but AS a POC we make use of one camera that streams the video underwater and analyses the position of swimmers to assess the probability of drowning, if it is higher then an alert will be generated to attract lifeguards' attention</p>
            <a href="#" class="ctn">Learn More</a>
          </div>
        </div>
      </section>

      <section class="explore">
        <div class="explore-content">
          <h1>A drowning detection system</h1>
          <div class="line"></div>
          <p>VirtualEye works like an “extra lifeguard” under the water of your pool.Our object recognition software tracks the movements of all swimmers in a pool.  And in the event of a serious drowning incident, VirtualEye will provide an alarm to pool lifeguards.  This will help lifeguards improve their reaction-time, as they initiate a rescue.</p>
        <a href="#" class="ctn">Learn More</a>
        </div>

      </section>

      <section class="footer">
        <p>Thank You</p>
      </section>

      <script>
        const menuBtn = document.querySelector('.menu-btn')
        const navlinks = document.querySelector('.nav-links')

        menuBtn.addEventListener('click',()=>{
          navlinks.classList.toggle('mobile-menu')
        })
      </script>
    </body>
    </html>
```

main.css:

```
$(document).ready(function () {
  // Init
  $('.image-section').hide();
  $('.loader').hide();
```

```
$('#result').hide();

// Upload Preview
function readURL(input) {
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        reader.onload = function (e) {
            $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
            $('#imagePreview').hide();
            $('#imagePreview').fadeIn(650);
        }
        reader.readAsDataURL(input.files[0]);
    }
}
$("#imageUpload").change(function () {
    $('.image-section').show();
    $('#btn-predict').show();
    $('#result').text('');
    $('#result').hide();
    readURL(this);
});

// Predict
$('#btn-predict').click(function () {
    var form_data = new FormData($('#upload-file')[0]);

    // Show loading animation
    $(this).hide();
    $('.loader').show();

    // Make prediction by calling api /predict
    $.ajax({
        type: 'POST',
        url: '/predict',
        data: form_data,
        contentType: false,
        cache: false,
        processData: false,
        async: true,
        success: function (data) {
            // Get and display the result
            $('.loader').hide();
            $('#result').fadeIn(600);
            $('#result').text('Prediction: '+data);
            console.log('Success!');
        },
```

```
        });
    });

});
```

GitHub & Project Demo Link

Gitup:https://github.com/IBM-EPBL/IBM-Project-28182-1660108194
Demo link:https://github.com/IBM-EPBL/IBM-Project-28182-1660108194/blob/main/Final%20Deliverables/Final%20Deliverables_alarm.mp3