



# HANRA JEONG

## PORTFOLIO

COMPUTER SCIENCE  
PORTFOLIO



# PORTFOLIO

Presentation

01

## Artificial Intelligence

Toxic Comment Classification  
Multi-agent path finding model  
Sentimental Analysis

02

## Software Development

Building Terminal  
Developing Chatbot through Socket Programming

03

## Web Development

Gamified Python Learning Website  
User-interactive 3D model through web developing

04

## Application Development

Apple Watch Game Development  
Community Application development for Seniors

05

## Data Base

Help-Hive : Community website

06

## Computer Vision

Edge Detection and Image matting program

# Natural Language Processing Research and Paper

# Artificial Intelligence

## Toxic Comment Classification

Identify toxic and harassing online comments

Authors  
Hanra Jeong / 30144973  
Trung Hieu Le / 30154401

1 Introduction

1.1 Motivation

With the development and easy access to the internet, toxic comments online has become problematic by making people stop communication and decrease the usage of online service. The ultimate goal of developing an automatic toxic comment detection system is to offer automatic filtering services to inappropriate comments to make a clean and better online environment. Some online platform like social network service, Twitter have offered toxic filtering systems. However, some of these services have been removed again by the platform itself. Moreover, even if it is done by artificial intelligence (AI), it caused some mistakes by filtering the non-insult comments, causing troubles in communication and usage of users.

1.2 Goals of project

Toxic and harassing online comments include large amounts of variety such as racism, crude language, disgrace and taboos.(13) Among these, we decided to work on 6 categories, "toxic", "severe toxic", "obscene", "profane", "sewage" and "identity hate". (1) This is because we chose to use the data offered by Kaggle, as reliable labeled data is important. By using this data mainly, we will focus on accuracy of detecting toxic comments. The data use for this project will be described in Experimental setup. To achieve our main goal, classification of toxic comments, we will build 3 different models, and compare their performance.

2 Related Work

There have been many related works done on toxic comments or words detection. Jayadev Bhaskaran et al (4) made the model to detect insults in social

commentary with Logistic Regression and Multi-layer Perception (MLP) for baseline and MLP, Convolutional Networks, Long short-term memory(LSTM) and Convolutional LSTMs for deep learning model. In this paper, it evaluates and compares these models and find out which one is better to the results, it says LSTM (GloVe) shows 0.771 F1 Score when SVM shows 0.734 and MLP (GloVe) shows 0.635 F1 Score performance. This demonstrates that LSTM model shows good performance in classification of toxic comments.

2.1 Model architecture

2.2 Pre-processing

2.3 Feature extraction

2.4 Model training

2.5 Model evaluation

2.6 Model deployment

2.7 Model improvement

2.8 Model maintenance

2.9 Model evolution

2.10 Model retirement

2.11 Model re-use

2.12 Model re-training

2.13 Model re-deployment

2.14 Model re-evaluation

2.15 Model re-improvement

2.16 Model re-maintenance

2.17 Model re-evolution

2.18 Model re-retirement

2.19 Model re-use

2.20 Model re-training

2.21 Model re-deployment

2.22 Model re-evaluation

2.23 Model re-improvement

2.24 Model re-maintenance

2.25 Model re-evolution

2.26 Model re-retirement

2.27 Model re-use

2.28 Model re-training

2.29 Model re-deployment

2.30 Model re-evaluation

2.31 Model re-improvement

2.32 Model re-maintenance

2.33 Model re-evolution

2.34 Model re-retirement

2.35 Model re-use

2.36 Model re-training

2.37 Model re-deployment

2.38 Model re-evaluation

2.39 Model re-improvement

2.40 Model re-maintenance

2.41 Model re-evolution

2.42 Model re-retirement

2.43 Model re-use

2.44 Model re-training

2.45 Model re-deployment

2.46 Model re-evaluation

2.47 Model re-improvement

2.48 Model re-maintenance

2.49 Model re-evolution

2.50 Model re-retirement

2.51 Model re-use

2.52 Model re-training

2.53 Model re-deployment

2.54 Model re-evaluation

2.55 Model re-improvement

2.56 Model re-maintenance

2.57 Model re-evolution

2.58 Model re-retirement

2.59 Model re-use

2.60 Model re-training

2.61 Model re-deployment

2.62 Model re-evaluation

2.63 Model re-improvement

2.64 Model re-maintenance

2.65 Model re-evolution

2.66 Model re-retirement

2.67 Model re-use

2.68 Model re-training

2.69 Model re-deployment

2.70 Model re-evaluation

2.71 Model re-improvement

2.72 Model re-maintenance

2.73 Model re-evolution

2.74 Model re-retirement

2.75 Model re-use

2.76 Model re-training

2.77 Model re-deployment

2.78 Model re-evaluation

2.79 Model re-improvement

2.80 Model re-maintenance

2.81 Model re-evolution

2.82 Model re-retirement

2.83 Model re-use

2.84 Model re-training

2.85 Model re-deployment

2.86 Model re-evaluation

2.87 Model re-improvement

2.88 Model re-maintenance

2.89 Model re-evolution

2.90 Model re-retirement

2.91 Model re-use

2.92 Model re-training

2.93 Model re-deployment

2.94 Model re-evaluation

2.95 Model re-improvement

2.96 Model re-maintenance

2.97 Model re-evolution

2.98 Model re-retirement

2.99 Model re-use

2.100 Model re-training

2.101 Model re-deployment

2.102 Model re-evaluation

2.103 Model re-improvement

2.104 Model re-maintenance

2.105 Model re-evolution

2.106 Model re-retirement

2.107 Model re-use

2.108 Model re-training

2.109 Model re-deployment

2.110 Model re-evaluation

2.111 Model re-improvement

2.112 Model re-maintenance

2.113 Model re-evolution

2.114 Model re-retirement

2.115 Model re-use

2.116 Model re-training

2.117 Model re-deployment

2.118 Model re-evaluation

2.119 Model re-improvement

2.120 Model re-maintenance

2.121 Model re-evolution

2.122 Model re-retirement

2.123 Model re-use

2.124 Model re-training

2.125 Model re-deployment

2.126 Model re-evaluation

2.127 Model re-improvement

2.128 Model re-maintenance

2.129 Model re-evolution

2.130 Model re-retirement

2.131 Model re-use

2.132 Model re-training

2.133 Model re-deployment

2.134 Model re-evaluation

2.135 Model re-improvement

2.136 Model re-maintenance

2.137 Model re-evolution

2.138 Model re-retirement

2.139 Model re-use

2.140 Model re-training

2.141 Model re-deployment

2.142 Model re-evaluation

2.143 Model re-improvement

2.144 Model re-maintenance

2.145 Model re-evolution

2.146 Model re-retirement

2.147 Model re-use

2.148 Model re-training

2.149 Model re-deployment

2.150 Model re-evaluation

2.151 Model re-improvement

2.152 Model re-maintenance

2.153 Model re-evolution

2.154 Model re-retirement

2.155 Model re-use

2.156 Model re-training

2.157 Model re-deployment

2.158 Model re-evaluation

2.159 Model re-improvement

2.160 Model re-maintenance

2.161 Model re-evolution

2.162 Model re-retirement

2.163 Model re-use

2.164 Model re-training

2.165 Model re-deployment

2.166 Model re-evaluation

2.167 Model re-improvement

2.168 Model re-maintenance

2.169 Model re-evolution

2.170 Model re-retirement

2.171 Model re-use

2.172 Model re-training

2.173 Model re-deployment

2.174 Model re-evaluation

2.175 Model re-improvement

2.176 Model re-maintenance

2.177 Model re-evolution

2.178 Model re-retirement

2.179 Model re-use

2.180 Model re-training

2.181 Model re-deployment

2.182 Model re-evaluation

2.183 Model re-improvement

2.184 Model re-maintenance

2.185 Model re-evolution

2.186 Model re-retirement

2.187 Model re-use

2.188 Model re-training

2.189 Model re-deployment

2.190 Model re-evaluation

2.191 Model re-improvement

2.192 Model re-maintenance

2.193 Model re-evolution

2.194 Model re-retirement

2.195 Model re-use

2.196 Model re-training

2.197 Model re-deployment

2.198 Model re-evaluation

2.199 Model re-improvement

2.200 Model re-maintenance



General LSTM without classifying clean text	0.53494
General LSTM with classifying clean text	0.95953
Bidirectional LSTM	0.96139
Convolutional Neural Networks (CNN)	0.5

Table 1: Summary of model performance : Average AUC scores computed by Kaggle

- ## Implement 3 different models:
1. General LSTM
  2. Bidirectional LSTM
  3. CNN

With the pre-processing of Natural Language data, it shows **96% accuracy** for toxic comment classification

# Artificial Intelligence

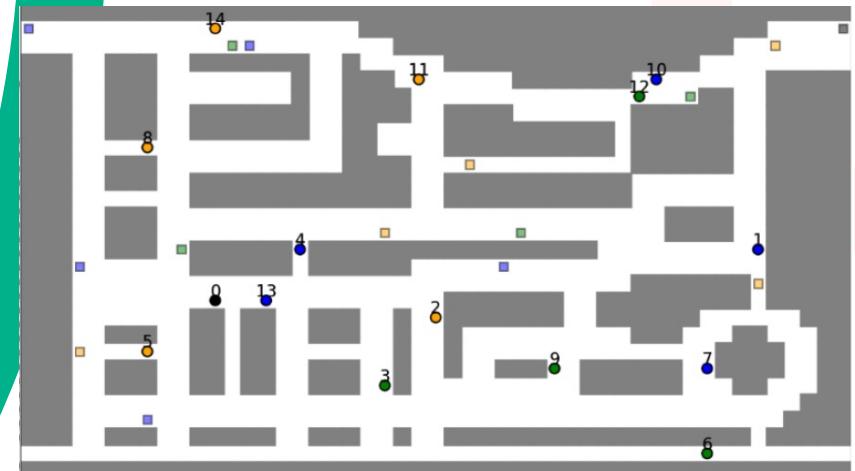
## Multi-agent Path Finding

Multi-agent Path Finding (MAPF) Design in Dynamic Environment

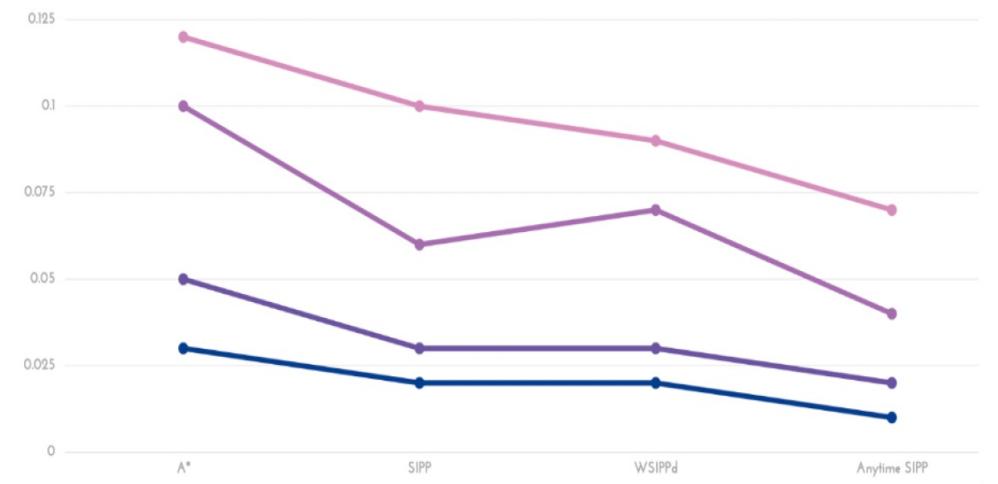
Implemented with 3 different algorithm:

1. A\* Algorithm
2. SIPP Algorithm
3. WSIPP<sub>d</sub> Algorithm
4. Anytime SIPP Algorithm

Adopted Dynamic Environment and real-life map  
to simulate self-driving environment



Running MAPF Simulation



Running Time Comparison of 4 different algorithms

# Artificial Intelligence

## Sentimental Analysis

- **Exec member** of Open Source Development Club in Simon Fraser University
- Leader of Sentimental Analysis Deep-learning Development
- **Awarded “Open Source Development Award”**
- Designed an overall architecture and pipelines of project
- Used BERT model
- Used Tweeter data labeled with positive, neutral and negative sentiment offered by Stanford

Showed **86% of accuracy** for predicting the sentiment of random text input

# Software Development

## Building Terminal

- Built OS command interpreter : SHELL
- In Unix System, with C
- Used system calls and kernel to design my own terminal

```
cshell$ ls
Makefile
cshell
cshell.c
cshell.h
cshell.o
script.txt
syscall_implementation
test.txt
test_script.txt
cshell$ theme red
cshell$ print "ABC"
"ABC"
cshell$ theme blue
```

```
cshell$ print 123
123
cshell$ log
Thu Jan 27 11:04:30 2022
  ls 0
Thu Jan 27 11:04:35 2022
    theme 0
Thu Jan 27 11:04:42 2022
    print 0
Thu Jan 27 11:04:51 2022
    theme 0
Thu Jan 27 11:04:55 2022
    print 0
cshell$ 
```

```
cshell$ $var = "foo"
Variable value expected
cshell$ $var=foo
cshell$ print $var
foo
cshell$ $var="foo"
cshell$ print $var
"foo"
cshell$ 
```



## The user at terminal 1

```
Welcome to Lets-Talk! Please type your messages now.  
Hi, how are you?  
I'm doing well, thank you!  
Alright, talk to you soon  
bye  
!exit
```

```
$ ./lets-talk 3000 localhost 3001
```

```
$ 
```

## The user at terminal 2

```
Welcome to LetS-Talk! Please type your messages now.  
Hi, how are you?  
I'm doing well, thank you!  
Alright, talk to you soon  
bye  
!exit
```

```
$ ./lets-talk 3001 localhost 3000
```

```
$ 
```

- Built inter-terminal chat-bot program with C
- Using Pthreads and UDP through socket programming under LINUX environment

# Software Development

## Developing ChatBot

# Web development

## Gamified Python Learning website

A screenshot of a web-based Python learning platform. The top navigation bar includes links for 'Intro to Python', 'Modules', 'Discussion', 'Competition', and 'My Profile'. A sidebar on the left shows 'Learning Material' with sections for 'Basic' and 'Coding Challenge'. The main content area displays a question titled 'Chained Conditionals' with a time limit of '1 minute(s)'. The question asks: 'What will the following code print if x = 5, y = 5, and z = 2?' followed by a code snippet:

```
if x < y & c:
    print(y - x)
elif y > x and y < z:
    print(y)
else:
    print(z)
```

Three options are provided: 'a', 'b', and 'c'. Below the options is a 'Submit' button. To the right of the question is a 'Hint' section with explanatory text and a flowchart diagram. At the bottom, there are navigation buttons for 'Prev', 'Page 3/3', and 'Next'.

- Built gamified python learning website
- Used React, JavaScript, HTML and CSS
- Built Firebase Database
- Designed an overall architecture and gamification factors of the website to teach python for novice.

A screenshot of the website's sign-up process. The top navigation bar is identical to the previous screenshot. The main content area features a 'Sign Up' form with fields for 'Name', 'Email address', 'Password', and 'Repeat password'. Below the form is a note about programming experience and a 'Yes' button. To the right of the sign-up form are three code snippets for users to answer:

1. "What is the output of the following code?"

```
myNumber = 1
if myNumber > 2:
    print("Correct")
else:
    print(myNumber)
```
2. "What is the output of the following code?"

```
def calculate(num1, num2):
    result = 0
    for i in range(0, num1, num2):
        result += i
    return result

result = calculate(0,2)
print(result)
```
3. "What is the output of the following code?"

```
def calculate(num1, num2=1):
    result = []
    for i in range(0, num1, num2):
        result.append(i)
    result = sum(result)
    return result

result = calculate(0,3)
print(result)
```

Each code snippet has an 'Answer' button below it. At the bottom of the page is a 'Sign up' button.

A screenshot of a module page titled 'Conditional Statements'. The top navigation bar is consistent. The main content area shows the module's purpose: 'This module covers conditional statements.' Below this are three buttons: 'Start', 'Review', and 'Compete'. To the right is a 'Recent Activity' section which currently shows 'No recent activity.'

# Web development

User-interactive 3D modelling



- Built and rendered Stanford-bunny
- Done with JavaScript
- Allowed user-interaction on the web through HTML

1. Moving bunny by dragging
2. Rotating with the right click of mouse, press "r" to reset the bunny
3. Point light, press "p" to turn on and off
4. Spot light, press "s" to turn on and off
5. Phong reflection and shading



- Needs :
  - Need user-interactive applications for apple watch to promote the usage other then using it for exercising purpose
  - Inspired by companion animals that can easily communicate with users
- Built with Swift on Xcode
- Used Iphone Application Built to enable user-interaction and modified for Apple Watch app

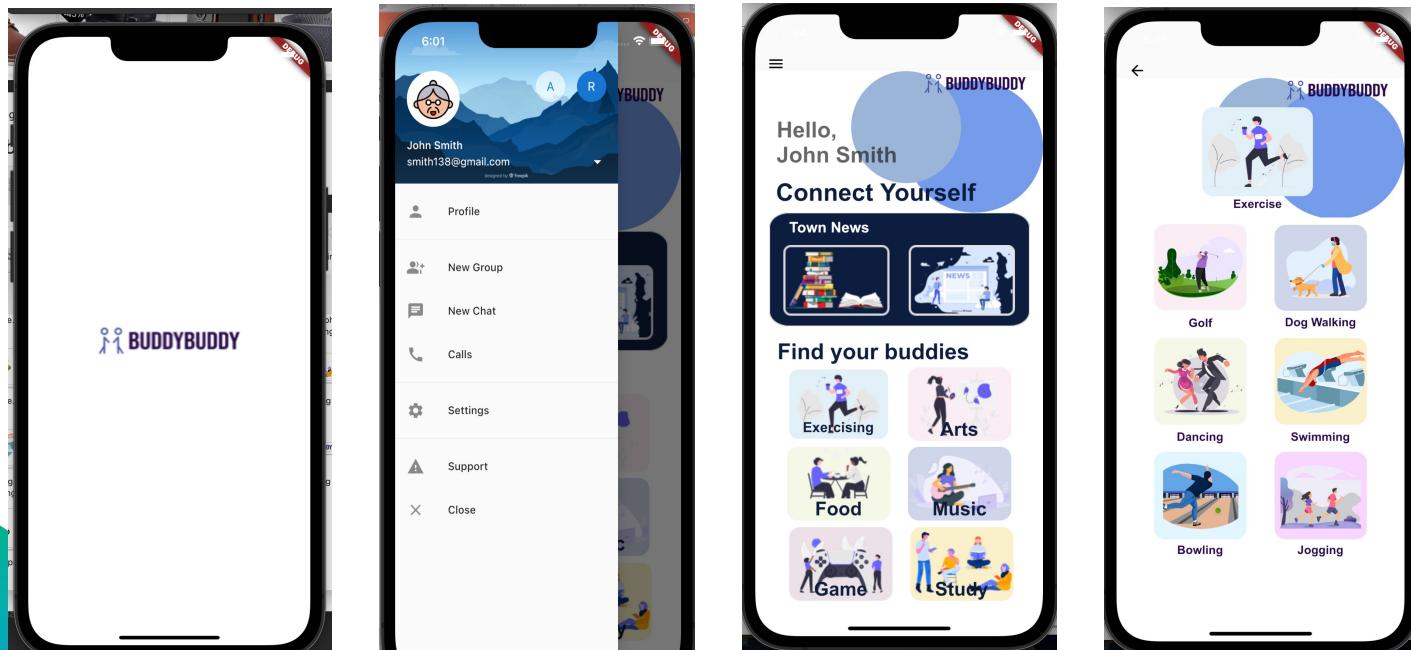
# Application Development

## Apple Watch Game Development

# Application Development

## BuddyBuddy: Community Application for elders

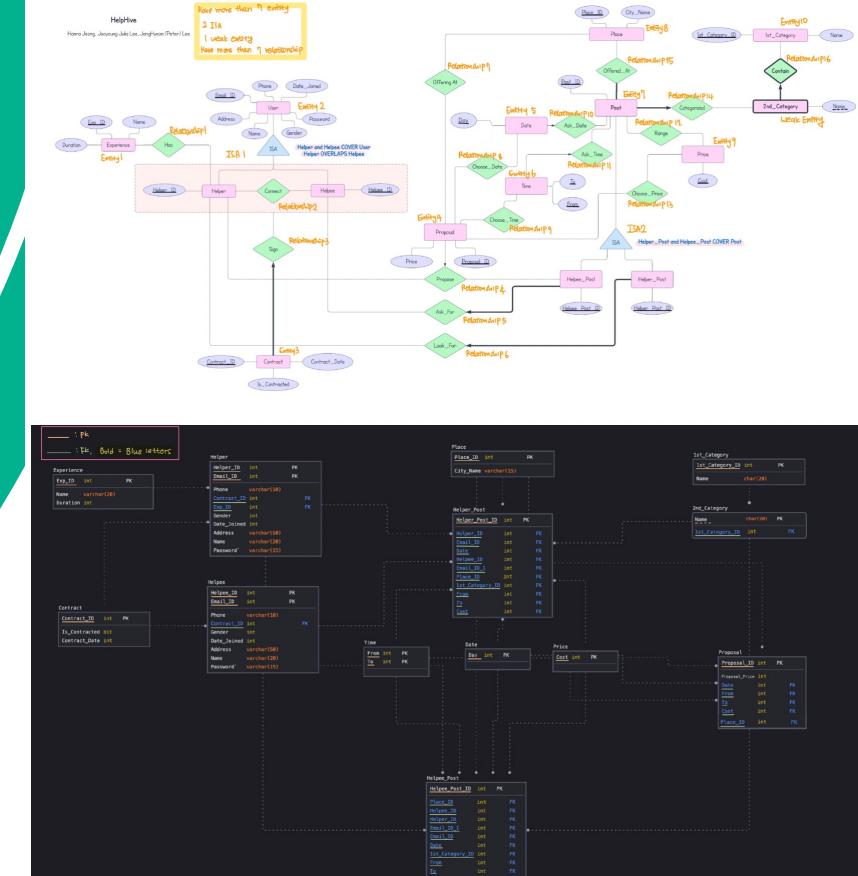
- Built Community Application that old people can easily communicate and find friends to enjoy together
- Done through Flutter (Dart)



# Data Base

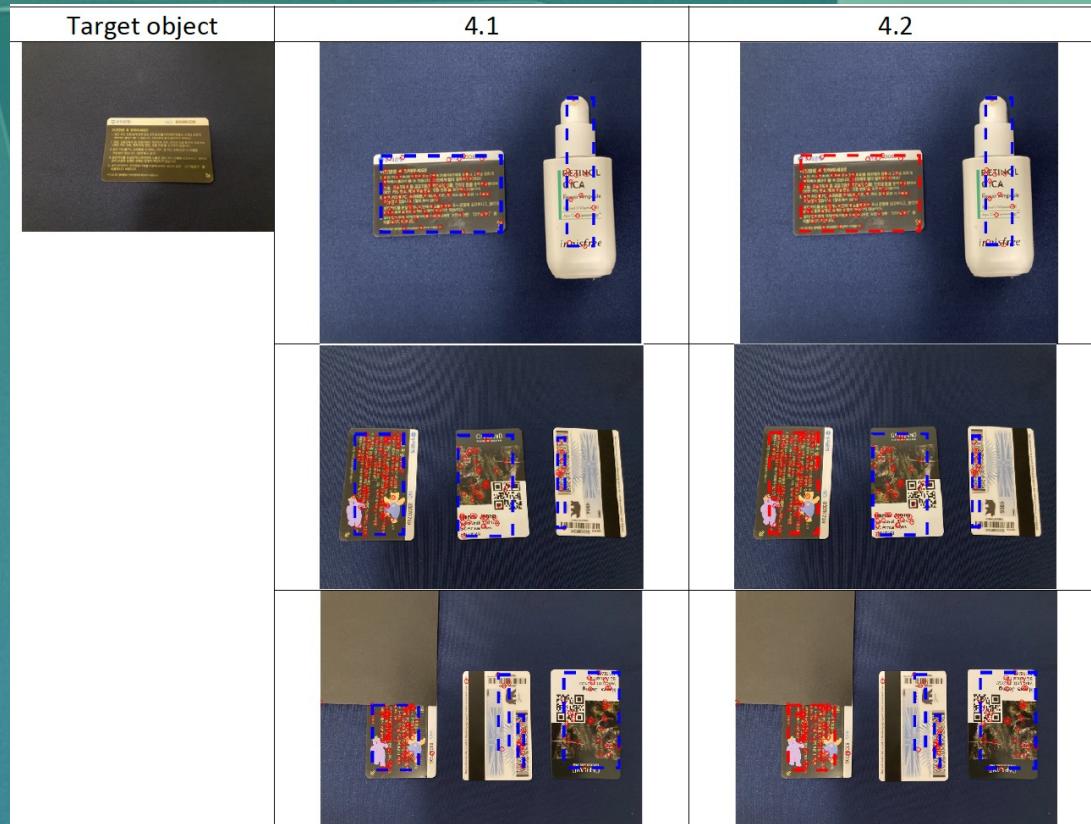
## Help-Hive : Community Website for students

- Built website to look for and ask for help within SFU students
- Designed Sequal data base and an overall architecture of the project including schema for DB
- UI design is done through HTML and CSS
- Gained expertise in database and website design, especially about building sequal based database with PHP



# Computer Vision

## Edge Detection and Image Matting



Edge Detection : Harris Corner Detection  
Object Detection : SIFT

Done through Matlab

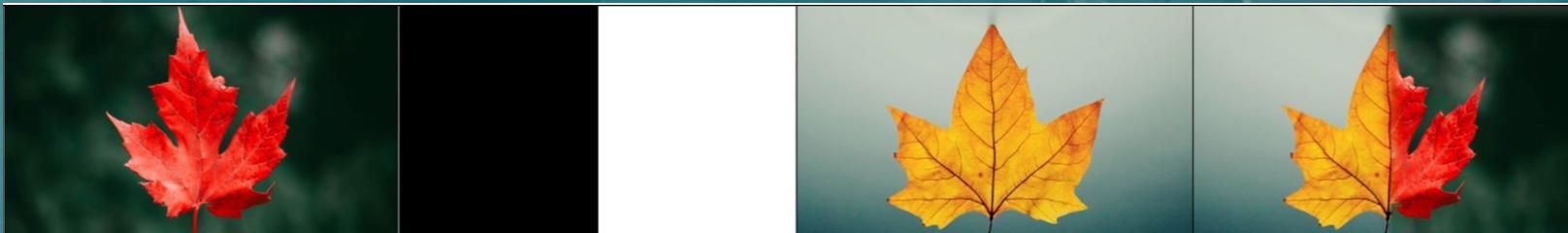
# Computer Vision

## Edge Detection and Image Matting

Non-binary Image Matting by generating alpha channel



Image Blending



Done through Matlab