iNTUition v6.0
IEEE NTU Student Branch
--------------------------------------

# The Ternary

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

*Project Report on*
**All-in-a-cup**

*(A web-app with the integration of Face Recognition Login for coming to play games)*

***Nguyen Duy Khanh*, *Do Xuan Long*, *Le Ky Nam, Tran Anh Tai***
*Undergraduate Students, Nanyang Technological University, Singapore*

## Contents

*Part 1. Introduction*

1. Motivation & Problem
2. Overview
3. The most difficult part of the project
4. Audience & Stakeholders

*Part 2. Background*

1. What the reader needs to know in order to understand the rest of the report?

*Part 3. Methodology*

1. Acknowledgements
2. Design & Front-end
3. Back-end
4. Face-Recognition & Modelling
5. Games

*Part 4. Testing & Results*

1. Test plans
2. Results of modelling
3. Analysis and Evaluation

*Part 5. Conclusion*

1. Conclusion
2. Discussion & Further Work

# *Part 1. Introduction*

## *1. Motivation & Problem*

With the development and the augmentation of the internet and technology, nowadays, a great deal of young engineers around the world and even many perennial businesses have come up with problems of applications of **high tech** in their software products or operating branches of companies. This problem has arisen since the demands of customers pertaining to expected requirements in terms of their conveniences.

As a result, my team has come up with an idea of creating a page for users to relax by playing games. In this product, the users would be required to login. One special thing in our product is that **Users can Login by Their face** instead of their password. In addition, we develop the **Traditional Snake Game** to become a modern version which has an option is **Two-Player Snake Games.**

## *2. Overview*

To use our web-page, the users are required to login to save your scores. They have two options of login: **Login by Face** or **by password** (traditional way). In addition, Our game is an developed version of **Traditional Snake Game** to become a modern version which has an option is **Two-Player Snake Game.**

Our web-app is divided into four parts:

+ **Front-end**: The design (the face of the web-app) and the content which is see-able by users.

+ **Back-end:** All the contents which is in the work of archiving, querying, and linking between the Database and other web-pages.

+ **Login by Face Recognition:** We use networks and algorithms in Deep Learning to perform the task of **Face Verification + Face Recognition.**

+ **Games:** We develop the **Traditional Snake Game** to become a modern version which has an option is **Two-Player Snake Games.**

## *3. The most difficult part of the project*

The most challenges with us are the high requirements of tasks, which is two problems:

- **Face Verification + Face Recognition** task
- **Merging different languages together** including Java (Back-end), Python (Deep Learning task), HTML (Contents of Front-end), CSS (Mark-up the page), JavaScript (Games task)

### *4. Audience & Stakeholders*

Our product is towards the use of **new models** and **algorithms** today, which is potential in terms of solving the problem of demanding conveniences of customers. As a result, it has a potential to be developed to sell for **other businesses**.

Additionally, to perform these steps, the users **can access quickly and easily**. The range of users is: from kids to the elderly.

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

## *Part 2. Background*

### *1. What the reader needs to know in order to understand the rest of the report & our codes?*

In order to understand **the rest of our report,** the readers need to have a basic background of Computer Science and Data Science. Particularly, the Users need to know:

- Basic Client-side Web language HTML/CSS & JavaScript.

- Basic Java language, Object-Oriented Concept and using Java as a back-end language.

- Basics understanding of Neural Networks, Convolutional Neural Networks, Pooling and Padding methods and somethings in the field of Computer Visions such as OpenCV library for face recognition and extracting face through an human image.

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

## *Part 3. Methodology*

### *1. Acknowledgements*

On fulfilling this project, out team truly thanks to those below communities as well as the sources code we reuse for our product:

- Java WebCam: https://github.com/sarxos/webcam-capture

- Java OpenCV library: https://github.com/opencv

- Front-end template: https://codepen.io/

- Face Recognition: https://github.com/iwantooxxoox/Keras-OpenFace/blob/master/utils.py

- Snake game in p5.js: https://www.youtube.com/watch?v=AaGK-fj-BAM

## 2. *Design & Front-end*

On coding our web-app, we basically use pure HTML/CSS without any related JavaScript Frameworks( e.g. React, React Native, JQuery). Our project take a reference from the Codepen template for make our web app looks more professional and beautiful.

## 3. *Back-end*

Using Java as an server-side language( backend), basically, this support the Java-Server Page (.jsp file) which is merely a kind of HTML file with the support of reading the embedded Java code within the file, hence, makes the backend processing much convenient.

## 4. *Face-Recognition & Modelling*

We use ConvNets, Pooling Methods, Padding Methods, Batch Normalization to accelerate the learning process and prevent Vanishing problem. Additionally, in the last three blocks, we used 3 blocks of Inception Nets to effectively our learning.

```
def faceRecoModel(input_shape):
    X_input = Input(input_shape)
    X = ZeroPadding2D((3, 3))(X_input)

    # First Block
    X = Conv2D(64, (7, 7), strides = (2, 2), name = 'conv1')(X)
    X = BatchNormalization(axis = 1, name = 'bn1')(X)
    X = Activation('relu')(X)

    # Zero-Padding + MAXPOOL
    X = ZeroPadding2D((1, 1))(X)
    X = MaxPooling2D((3, 3), strides = 2)(X)
```

```
# Second Block
X = Conv2D(64, (1, 1), strides = (1, 1), name = 'conv2')(X)
X = BatchNormalization(axis = 1, epsilon=0.00001, name = 'bn2')(X)
X = Activation('relu')(X)

# Zero-Padding + MAXPOOL
X = ZeroPadding2D((1, 1))(X)

# Second Block
X = Conv2D(192, (3, 3), strides = (1, 1), name = 'conv3')(X)
X = BatchNormalization(axis = 1, epsilon=0.00001, name = 'bn3')(X)
X = Activation('relu')(X)

# Zero-Padding + MAXPOOL
X = ZeroPadding2D((1, 1))(X)
X = MaxPooling2D(pool_size = 3, strides = 2)(X)

# Inception 1: a/b/c
X = inception_block_1a(X)
X = inception_block_1b(X)
X = inception_block_1c(X)

# Inception 2: a/b
X = inception_block_2a(X)
X = inception_block_2b(X)

# Inception 3: a/b
X = inception_block_3a(X)
X = inception_block_3b(X)

# Top layer
X = AveragePooling2D(pool_size=(3, 3), strides=(1, 1), data_format='channels_first')(X)
X = Flatten()(X)
X = Dense(128, name='dense_layer')(X)

X = Lambda(lambda  x: K.l2_normalize(x,axis=1))(X)
model = Model(inputs = X_input, outputs = X, name='FaceRecoModel')
return model
```

## 5. Games

### 5.1 Encryption and Decryption a text.

The first is written in Python which we converted a text file to a png file (each character correspond to a number in RGB) then another one to decode what we encoded. There are keys to determine the rule of translation from character to a number in RGB. To create the dictionary which match a character to a 8-bits integer value (0-255), we use random in python.

### 5.2 Modern-version of Traditional Snake Game

The second game is Snake, a really basic game made by OOP and we use p5.js to make the drawing step easier.  We use queue, a kind of data structure. The snake is defined by an array in javascript, each element of the array is another array with two elements contains horizontal and vertical positions of snake body, our convention is the first element is head (0-index).

Let total[ ] is the property define the snake’s position, total[0] is the head. The snake move by pop the tail and add an element [total[0]+v[0],total[1]+v[1]] to the first position of the total[ ]. Another innovation in this game is multiplayer mode, now there are two snake. We define a method call “collision”, in which if two a snake collides with other, the collided one will be cut from collided position to tail and the colliding one will grow equal the length lost. If their head collide, the longer one win.

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

# Part 4. Testing & Results

## 1. Results of modelling

### Face recognition part

```
from sklearn import metrics
from scipy.optimize import brentq
from scipy import interpolate

tpr, fpr, accuracy, val, val_std, far = lfw.evaluate(emb_array,
                actual_issame, nrof_folds=lfw_nrof_folds)

print('Accuracy: %1.3f+-%1.3f' % (np.mean(accuracy), np.std(accuracy)))
print('Validation rate: %2.5f+-%2.5f @ FAR=%2.5f' % (val, val_std, far)
)
auc = metrics.auc(fpr, tpr)
print('Area Under Curve (AUC): %1.3f' % auc)
```

```
eer = brentq(lambda x: 1. - x - interpolate.interp1d(fpr, tpr)(x), 0., 1.)
print('Equal Error Rate (EER): %1.3f' % eer)
```

```
Accuracy: 0.938+-0.013
Validation rate: 0.47469+-0.04227 @ FAR=0.00134
Area Under Curve (AUC): 0.979
Equal Error Rate (EER): 0.062
```

### 3. *Analysis and Evaluation*

**Our analysis** about the model is:

- The input image size is still small. This shortcoming is on account of the limited times in the contest so we cannot self-train the model.
- The model dose not include many layers. It is because the cost-computing problems. This shortcoming leads to the low accuracy of the model.
- The model can not self detect the face of users. This is because we cannot code YOLO Algorithm to detect the face on account of the limited time in the contest.
- The model can not detect the live-person with a person in an image. This would lead to the serious problem of confidenciality.

---

## *Part 5. Conclusion*

### 1. *Conclusion*

This web-page is a simple integration of web-app and Deep Learning application and Games. In this product, the users would be required to login. One special thing in our product is that **Users can Login by Their face** instead of their password. In addition, we develop the **Traditional Snake Game** to become a modern version which has an option is **Two-Player Snake Games.**

### 2. Discussion & Further Work

**Face recognition:**

- Implement **YOLO Algorithm** to make the process more professional,
- Detect the live-person with a not-live image based on the emotional characteristics of the face.
- Improve our models according to the above evaluations.

**Login more professional**: By speed: Natural Language Processing

In the future, **the game** can be designed again to have better look and can also be develop in term of engine. We hope the website will be a place for everyone to relax with simple platform application.

*Thank you for reading our project report!*