



TRƯỜNG ĐẠI HỌC  
**SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**  
HCMC University of Technology and Education



**Hồ Chí Minh, ngày 23, tháng 06, năm 2022**

**Đỗ Công Danh – 19146163**

**MSSV:19146163**

### **NHẬN DIỆN NGŨ GẠT SỬ DỤNG GIẢI THUẬT CNN**

#### **Tóm tắt**

Những năm gần đây, ở nước ta, cùng với quá trình phát triển nhanh của các phương tiện giao thông, con số tai nạn giao thông ngày càng tăng nhiều, đặt ra một mối nguy hiểm nghiêm trọng cho cuộc sống xã hội và người tham gia giao thông. Tai nạn giao thông (TNGT) đã và đang trở thành nỗi đau lớn của nhiều gia đình, trong những vụ tai nạn giao thông, người thì mang tật suốt đời, người tử vong để lại những khoảng trống không gì bù đắp nổi cho người thân. Và một trong những nguyên nhân chính của tai nạn giao thông là sự thiếu tập trung của người lái xe do mệt mỏi hay buồn ngủ.

Theo báo cáo của Tổng cục Thống kê cho biết, tính chung 7 tháng đầu năm 2015 trên địa bàn cả nước đã xảy ra 12.910 vụ tai nạn giao thông và bình quân mỗi ngày 61 vụ. Và theo phân tích của Cục Cảnh Sát Giao Thông, gần 70% số vụ TNGT xảy ra vào khoảng thời gian từ 12h đến 24h, đây là khoảng thời gian người điều khiển phương tiện bị tác động tâm lý của sự mệt mỏi, căng thẳng, sự chênh lệch về nhiệt độ, ánh sáng giữa

ngày và đêm (đặc biệt đối với phương tiện vận tải hành khách, hàng hóa...)

Báo cáo về "Rối loạn giấc ngủ và tai nạn giao thông" tại hội nghị khoa học thường niên Hội Hồ hấp Việt Nam và Chương trình đào tạo y khoa liên tục 2015, giáo sư Telfilo Lee Chiong (Trung tâm National Jewish Health, Mỹ), cho biết buồn ngủ là một trong những nguyên nhân chính gây tai nạn giao thông trên thế giới. Ước tính khoảng 10-15% tai nạn xe có liên quan đến thiếu ngủ. Nghiên cứu về giấc ngủ ở các tài xế 19 quốc gia châu Âu cho thấy tỷ lệ buồn ngủ khi lái xe cao, trung bình 17%.

Trong đó 10,8% người buồn ngủ khi lái xe ít nhất một lần trong tháng, 7% từng gây tai nạn giao thông do buồn ngủ, 18% suýt xảy ra tai nạn do buồn ngủ.

Những số liệu thống kê đáng báo động chỉ ra sự cần thiết để thực hiện các hệ thống có khả năng theo dõi và cảnh báo tình trạng mệt mỏi, buồn ngủ của người lái xe để có thể ngăn chặn những vụ TNGT đáng tiếc có thể xảy ra.

## I. Tổng quan:

Trí tuệ nhân tạo hay trí thông minh nhân tạo (Artificial intelligence – viết tắt là AI) là một ngành thuộc lĩnh vực khoa học máy tính (Computer science). Là trí tuệ do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người.

Trí tuệ nhân tạo khác với việc lập trình logic trong các ngôn ngữ lập trình là ở việc ứng dụng các hệ thống học máy (machine learning) để mô phỏng trí tuệ của con người trong các xử lý mà con người làm tốt hơn máy tính.

Cụ thể, trí tuệ nhân tạo giúp máy tính có được những trí tuệ của con người như: biết suy nghĩ và lập luận để giải quyết vấn đề, biết giao tiếp do hiểu ngôn ngữ, tiếng nói, biết học và tự thích nghi,...

Tuy rằng trí thông minh nhân tạo có nghĩa rộng như là trí thông minh trong các tác phẩm khoa học viễn tưởng, nó là một trong những ngành trọng yếu của tin học. Trí thông minh nhân tạo liên quan đến cách cư xử, sự học hỏi và khả năng thích ứng thông minh của máy móc.

Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay.

CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh. Để tìm hiểu tại sao thuật toán này được sử dụng rộng rãi cho việc nhận dạng

(detection), chúng ta hãy cùng tìm hiểu về thuật toán này.

## II. Đối tượng và phạm vi nghiên cứu:

### 1. Đối tượng nghiên cứu:

- Ngôn ngữ lập trình Python
- Sử dụng mạng CNN
- Các thuật toán và kỹ thuật phát hiện tình trạng buồn ngủ dựa trên cử chỉ gương mặt

### 2. Phạm vi nghiên cứu:

- Nghiên cứu về các phương pháp đã được đề xuất phát hiện và cảnh báo tình trạng buồn ngủ của người lái xe trên thế giới theo những bài báo và nghiên cứu khoa học
- Chương trình demo sử dụng ngôn ngữ lập trình Python, sử dụng mạng CNN trên nền tảng hệ điều hành Windows.

## III. Phương pháp:

### 1. Thu thập dữ liệu:

Với đề tài nhận diện ngủ gật thì đầu tiên không thể thiếu là data hình ảnh.



Ở đây ta đã thu thập được 100 ảnh mỗi loại tổng cộng 400 ảnh, được chia 30% test và 70% train.

### 2. Xử lý dữ liệu:

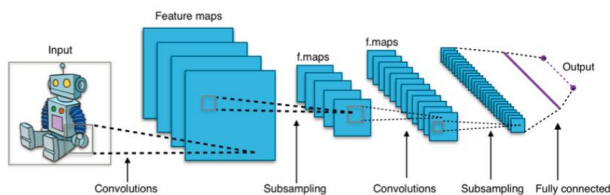
Ảnh sau khi thu thập đã được cắt đồng bộ kích thước dữ liệu bằng ảnh màu RGB 150x150

Tiếp tục chuẩn hóa dữ liệu đầu vào bằng cách chia toàn bộ giá trị màu cho 255 để được giá trị min là 0.0 và max

là 1.0 thay cho giá trị ban đầu giúp mô hình tính toán chuẩn hơn và ít tốn thời gian cũng như bộ nhớ

## 2. Tạo training model sử dụng CNN

Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.



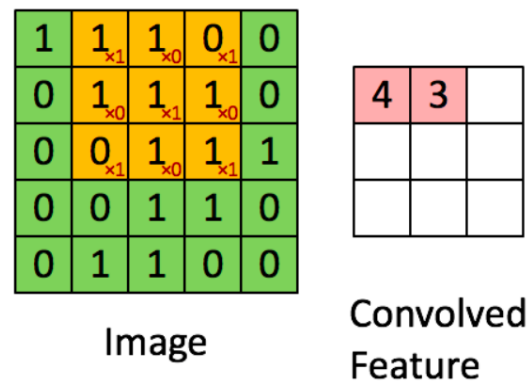
Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

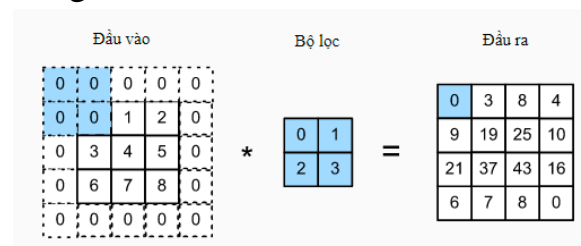
**Convolutional:** Là một cửa sổ trượt (Sliding Windows) trên một ma trận.

Các convolutional layer có các parameter(kernel) đã được học để tự điều chỉnh lấy ra những thông tin chính xác nhất mà không cần chọn các feature.

Convolution hay tích chập là nhân từng phần tử bên trong ma trận  $3 \times 3$  hoặc  $5 \times 5$  với ma trận ban đầu. Kết quả được một ma trận gọi là Convoled feature được sinh ra từ việc nhân ma trận Filter với ma trận ảnh ban đầu.

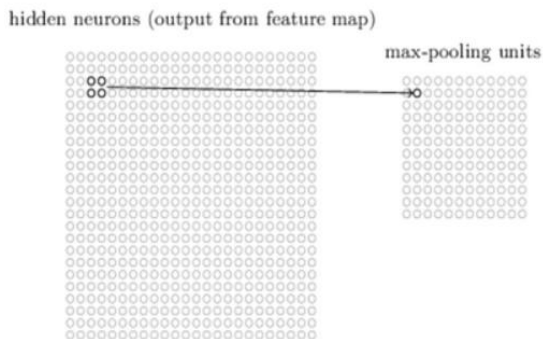


Khi áp dụng các tầng tích chập là việc chúng ta có thể mất một số điểm ảnh trên biên của ảnh. Vì chúng ta thường sử dụng các hạt nhân nhỏ, với một phép tích chập ta có thể chỉ mất một ít điểm ảnh, tuy nhiên sự mất mát này có thể tích lũy dần khi ta thực hiện qua nhiều tầng tích chập liên tiếp. Một giải pháp đơn giản cho vấn đề này là chèn thêm các điểm ảnh xung quanh đường biên trên bức ảnh đầu vào, nhờ đó làm tăng kích thước sử dụng của bức ảnh.



Tiếp theo là lớp pooling thường được sử dụng ngay sau lớp convolutional để đơn giản hóa thông tin đầu ra để giảm bớt số lượng neuron.

Thủ tục pooling phổ biến là max-pooling, thủ tục này chọn giá trị lớn nhất trong vùng đầu vào  $2 \times 2$ .



## Tiến hành tạo training model

```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dense
model = Sequential()

model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(150, 150, 3)))
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(MaxPooling2D((2, 2)))

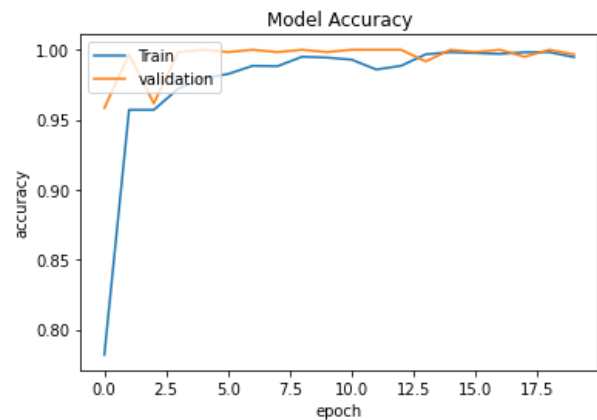
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(Dense(2, activation='softmax'))
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 150, 150, 32)	896
conv2d_1 (Conv2D)	(None, 150, 150, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 75, 75, 32)	0
conv2d_2 (Conv2D)	(None, 75, 75, 64)	18496
conv2d_3 (Conv2D)	(None, 75, 75, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 37, 37, 64)	0
conv2d_4 (Conv2D)	(None, 37, 37, 128)	73856
conv2d_5 (Conv2D)	(None, 37, 37, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 18, 18, 128)	0
flatten (Flatten)	(None, 41472)	0
dense (Dense)	(None, 128)	5308544
dense_1 (Dense)	(None, 2)	258
=====		
Total params:	5,595,810	
Trainable params:	5,595,810	
Non-trainable params:	0	

## IV. Kết quả:

Sau khi tiến train ta thu được kết quả



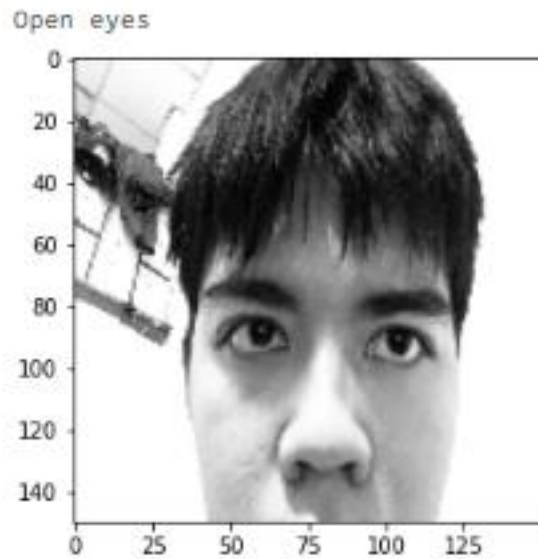
```
Score=model.evaluate(training_set,verbose=0)
print('Train Loss', Score[0])
print('Train Accuracy', Score[1])
```

Train Loss 0.01867879368364811  
Train Accuracy 0.9941176176071167

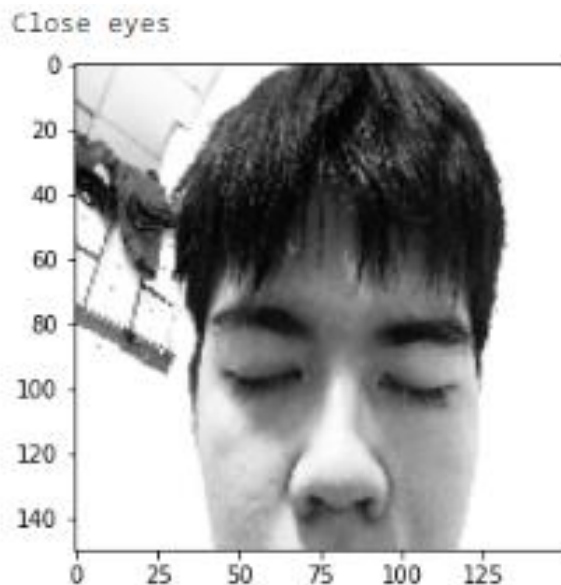
Ta tiến hành chạy nhận diện thử nghiệm.

Đối với hình ảnh:

Khi mở mắt:



Khi nhắm mắt:



Đối với video realtime:

Khi mở mắt:



Khi nhắm mắt:



Hướng phát triển:

- ✓ Sử dụng lên nhiều khuôn mặt.
- ✓ Ứng dụng trên nhiều mô hình.



## V. Kết luận:

- ✓ Sau quá trình chạy thử nghiệm thì có thể rút ra nhận xét như sau:
  - Ưu điểm:  
Hệ thống tận dụng các thiết bị đã có nên tiết kiệm chi phí.  
Vận dụng được kiến thức đã học để xây dựng nên model mạng CNN đã được học.
  - Nhược điểm:  
Dữ liệu được tạo còn ít nên nhận diện còn nhiều thiếu sót trong realtime.  
Ứng dụng chỉ mới chạy trên máy tính. Chưa có được mô hình thực tế.

## VI. Tài liệu tham khảo:

- ✓ Nghiên cứu tình trạng buồn ngủ - Thái Thị Hoa Văn.
- ✓ Link gifhub mô hình:

