ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC AN GIANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO MÔN HỌC**

**TRÍ TUỆ NHÂN TẠO**

Ứng dụng cảnh báo khi sờ lên mặt

**NHÓM 13 – LỚP DH19TH2**

**AN GIANG, 05-2021**

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC AN GIANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO MÔN HỌC**

**TRÍ TUỆ NHÂN TẠO**

Ứng dụng cảnh báo khi sờ lên mặt

**NHÓM 13 – LỚP DH19TH2**

**GVHD: NGUYỄN VĂN HÒA**

**AN GIANG, 05-2021**

**NHẬN XÉT CỦA GVHD**

**Giảng viên hướng dẫn**

*(Ký và ghi rõ họ tên)*

Nội dung nhận xét:

* Đồng ý hay không đồng ý cho sinh viên báo cáo. Nếu không đồng ý cần ghi rõ lý do.
* Kết quả đạt được so với yêu cầu.
* Ý kiến khác (nếu có)

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc73192884)

[CHƯƠNG 1: TỔNG QUAN 2](#_Toc73192885)

[1.1 Mô tả bài toán 2](#_Toc73192886)

[1.2 Yêu cầu hệ thống 2](#_Toc73192887)

[CHƯƠNG 2: THƯ VIỆN 3](#_Toc73192888)

[2.1 TensorFlow.JS 3](#_Toc73192889)

[2.2 Tensorflow-models 3](#_Toc73192890)

[CHƯƠNG 3: THUẬT TOÁN KNN 4](#_Toc73192891)

[3.1 Ví dụ đơn giản 4](#_Toc73192892)

[3.2 Ý tưởng của KNN 5](#_Toc73192893)

[CHƯƠNG 4: CÁC BƯỚC TIẾN HÀNH 6](#_Toc73192894)

[CHƯƠNG 5: TỔNG KẾT 7](#_Toc73192895)

[Ưu điểm: 7](#_Toc73192896)

[Nhược điểm: 7](#_Toc73192897)

[Hướng phát triển: 7](#_Toc73192898)

[Nguồn tham khảo: 7](#_Toc73192899)

Danh sách nhóm

|  |  |
| --- | --- |
| Họ và tên | MSSV |
| Phạm Ngọc Hòa | DTH185278 |
| Trần Tấn Duy | DTH185250 |
| Dương Khởi Minh | DTH185488 |
| Lê Ngọc Thắng | DTH185388 |
| Mai Quang Huy | DTH185258 |

# LỜI MỞ ĐẦU

Trí tuệ nhân tạo là một lĩnh vực của khoa học máy tính với mục tiêu nghiên cứu xây dựng và ứng dụng các hệ thống thông minh nhân tạo. Đây là một trong những lĩnh vực được quan tâm nghiên cứu nhiều nhất của khoa học máy tính hiện nay với nhiều kết quả ứng dụng rộng rãi.

Với sự hướng dẫn của thầy Nguyễn Văn Hòa, nhóm chúng em thực hiện đề tài **“Ứng dụng cảnh báo khi sờ tay lên mặt”** để vận dụng những gì được học vào thực tiễn, xây dựng một ứng dụng cảnh báo đơn giản dành cho những người có thói quen sờ tay lên mặt nhằm ngăn ngừa nguy cơ lây nhiễm covid-19.

Dù đã rất cố gắng, chúng em cũng khó có thể tránh khỏi những thiếu sót nhất định. Rất mong nhận được sự đóng góp ý kiến của các Thầy Cô, để chương trình ngày càng hoàn thiện hơn.

Xin Chân Thành Cảm Ơn!

# CHƯƠNG 1: TỔNG QUAN

Tutorials xây dựng ứng dụng "Cảnh báo khi sờ tay lên mặt" sử dụng mã nguồn mở TensorFlow cho Machine learning.

## 1.1 Mô tả bài toán

Hầu hết chúng ta đều có thói quen sờ tay lên mặt. Cứ 1 lúc lại chạm 1 lần để vuốt tóc, dùng ngón tay để ngoáy mũi, lấy tay để nặn mụn, dụi mắt, mát xa, còn các em bé có thói quen mút tay.

Nhưng thói quen này làm tăng nguy cơ mắc cúm, cảm lạnh và đặc biệt là COVID-19 do virus Corona chủng mới gây ra.

Các chuyên gia còn cho biết, mắt và miệng là 2 bộ phận các virrus, vi khuẩn dễ xâm nhập nhất vào cơ thể…

Vì thế ứng dụng cảnh báo khi sờ tay lên mặt có nhiệm vụ cảnh báo người dung bằng audio thông qua webcam để nhận diện, phù hợp cho HS-SV học online hay nhân viên văn phòng trong mùa covid-19 đang diễn biến phức tạp thời gian qua.

## 1.2 Yêu cầu hệ thống

Giúp người dùng có thể dễ dàng sử dụng và hạn chế thói quen sờ tay lên mặt. Thao tác nhanh gọn, tiện lợi. Hệ thống với giao diện gần gũi giúp cho người dùng dễ sử dụng, âm thanh cảnh báo to, rõ ràng, giúp cảnh báo hiệu quả.

Ứng dụng chạy trên nền tảng web, có thể đưa ra cảnh báo cho người dùng khi họ đưa tay lên mặt.

Ứng dụng sử dụng dữ liệu cục bộ, không lưu trữ bất ký thông tin gì của người dùng.

# CHƯƠNG 2: THƯ VIỆN

## 2.1 TensorFlow.JS

TensorFlow là một thư viện mã nguồn mở (open-source) cho các ứng dụng machine learning. TensorFlow cũng có thể được sử dụng để implement các neural networks hay các thuật toán deep learning.

TensorFlow.JS Là một phiên bản của thư viện machine learning TensorFlow nổi tiếng, giúp mang deep learning tới thế giới Javascript. Giờ đây, bạn có thể define, train, và chạy Machine Learning với Nodejs.

Thông qua thư viện này, các nhà phát triển có thể sử dụng API Javascript để training, load, và chạy models.

Link: <https://github.com/tensorflow/tfjs>

## 2.2 Tensorflow-models

**-Mobilenet**: @tensorflow-models/mobilenet

Giúp phân loại hình ảnh theo nhãn dựa trên cơ sở dữ liệu của [ImageNet database](https://www.image-net.org/)

TensorFlow.js Mobilenet Demo: <https://storage.googleapis.com/tfjs-models/demos/mobilenet/index.html>

**- knn-classifier**: @tensorflow-models/knn-classifier

Gói này cung cấp một tiện ích để tạo bộ phân loại bằng thuật toán K-Nearest Neighbors. Có thể được sử dụng để transfer learning , phân loại ảnh dựa trên thông tin trả về từ moblenet.

**-React Js**: làm frontend cho tool

React Js là một thư viện viết bằng **javascript,** dùng để **xây dựng giao diện người dùng**(UI). React được sử dụng rộng rãi và có hệ sinh thái đa dạng phong phú

# CHƯƠNG 3: THUẬT TOÁN KNN

**K-nearest neighbors là thuật toán học máy có giám sát, đơn giản và dễ triển khai. Thường được dùng trong các bài toán phân loại và hồi quy.**

## 3.1 Ví dụ đơn giản

Bài toán đặt ra: Bạn có điểm của một môn học nhưng bạn không biết thuộc loại nào (Giỏi, khá, trung bình, yếu). Giả sử bạn không biết bất kì quy tắc nào để phân loại cả.

Có một cách giải quyết là bạn phải đi khảo sát những người xung quanh. Để biết điểm của mình thuộc loại nào thì bạn phải đi hỏi những đứa có điểm gần số điểm mình nhất. Giả sử trong lớp 50 đứa, mình khảo sát 5 đứa gần điểm mình nhất và được dữ liệu như sau:

Điểm của tôi: 7

Điểm của bạn tôi:

* 7.1 => Khá
* 7.2 => Khá
* 6.7 => Khá
* 6.6 => Khá
* 6.4 => Trung bình

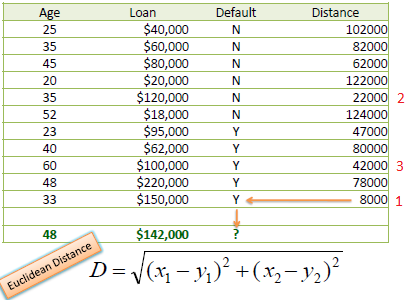
Qua kết quả trên thì mình sẽ mạnh dạng đoán là mình loại khá đúng không nào? Với cách này chúng ta có thể phân loại dữ liệu 1 chiều (1 feature) bằng cách làm khá đơn giản. Và các bạn có nhận thấy rằng dữ liệu mình khảo sát càng nhiều, càng rộng thì dự đoán đưa ra càng chính xác (Giả sử lớp bạn không có ai loại khá ngoài bạn thì cho dù bạn lấy bao nhiêu người gần điểm bạn nhất củng sẽ ra kết quả sai).

## 3.2 Ý tưởng của KNN

Thuật toán KNN cho rằng những dữ liệu tương tự nhau sẽ tồn tại **gần nhau** trong một không gian, từ đó công việc của chúng ta là sẽ tìm k điểm gần với dữ liệu cần kiểm tra nhất. Việc tìm khoảng cách giữa 2 điểm củng có nhiều công thức có thể sử dụng, tùy trường hợp mà chúng ta lựa chọn cho phù hợp.

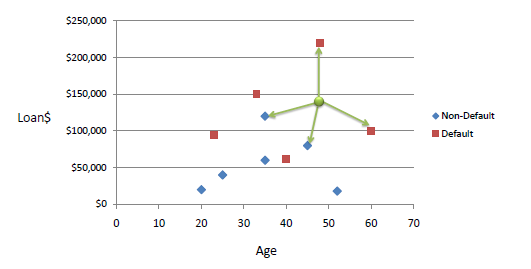
Đây là 3 cách cơ bản để tính khoảng cách 2 điểm dữ liệu x, y có k thuộc tính:

Ví dụ chúng ta có dữ liệu là tuổi, khoản vay và khả năng vỡ nợ như hình:



Thực hiện tương tự, ta sẽ tính được khoảng cách ở cột Distance, từ đó chọn ra k = 3 khoảng cách nhỏ nhất (gần với dữ liệu vào nhất). Với 3 khoảng cách này chúng ra nhận được 3 label là (Yes, No, Yes). Trong 3 label này Yes xuất hiện nhiều hơn nên chúng ta sẽ đưa ra dự đoán người này có khả năng vở nợ.

Vì đây là dử liệu 2 chiều nên chúng ta củng có thể biểu diễn dữ liệu trong hệ tọa độ như hình:



Trên hệ tọa độ này chúng ta thể dễ dàng nhận thấy cách chúng ta chọn k điểm gần nhất. Nhưng với dữ liệu lớn, nhiều chiều thì việc biểu diễn dữ liệu trên một không gian là không hề dễ dàng.

Tóm lại, KNN (K-Nearest Neighbors) là một trong những thuật toán học có giám sát đơn giản nhất được sử dụng nhiều trong khai phá dữ liệu và học máy. Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới.  
      Lớp (nhãn) của một đối tượng dữ liệu mới có thể dự đoán từ các lớp (nhãn) của k hàng xóm gần nó nhất.

# CHƯƠNG 4: CÁC BƯỚC TIẾN HÀNH

\* Bước 1: Train cho máy khuôn mặt không chạm tay

    \* Bước 2:Train cho máy khuôn mặt đã chạm tay

\* Bước 3:lấy hình ảnh hiện tai,phân tích và so sánh với data đã

cho trước đó.

* nếu matching với khuôn mặt chạm tay => cảnh báo
* chụp ảnh webcam tại thời điểm training sang hình ảnh (HTMLVideoElement) (img),
* Đẩy dữ liệu cho mobilenet phân tích qua cơ sở dữ liệu [ImageNet database](https://image-net.org/) sau đó trả lại thông tin(nhãn).
* Sau đó đưa dữ liệu cho knn-classifier học dựa trên thuật toán KNN ,knn cls phân tích và trả vê dl dưới dạng label và confidences,
* Dựa vào lable và confidences trả về ta thiết lập điều kiện cảnh báo tương ứng lúc nào giơ tay lên lúc nào không

# CHƯƠNG 5: TỔNG KẾT

## Ưu điểm:

* Chương trình dễ sử dụng, thuận tiện cho người sử dụng.
* Khả năng cảnh cáo linh hoạt.
* Làm quen với tư tưởng "Máy học".
* Biết áp dụng làm tool có concept tương tự
* Biết thêm một số kỹ thuật với Javascript

## Nhược điểm:

* Chưa hiểu sâu thuật toán, cách thức tổ chức dữ liệu đầu vào, đầu ra.

## Hướng phát triển:

* Tìm hiểu và áp dụng thêm một số chức năng vào ứng dụng.
* Ứng dụng thuật toán đã tìm hiểu vào các vấn đề khác của đời sống.

## Nguồn tham khảo:

* Slide bài giảng: Trí tuệ nhân tạo - Giảng viên: Nguyễn Văn Hòa Khoa CNTT - ĐH An Giang
* Từ Minh Phương(2014). Giáo trình nhập môn trí tuệ nhân tạo. Học viện công nghệ bưu chính viễn thông.
* Github: <https://github.com/tensorflow>
* Khóa học f8: <https://fullstack.edu.vn/>