

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**CT452
THỰC TẬP THỰC TẾ - TT&MMT**

Cán bộ hướng dẫn:
Prof. Đỗ Thanh Nghị
Giảng viên hướng dẫn:
Prof. Đỗ Thanh Nghị

Sinh viên thực hiện:
Nguyễn Thanh Huy
MASV: B1709280

Cần Thơ, 08/2021

LỜI CẢM ƠN

Em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin & Truyền Thông đã tạo cơ hội trải nghiệm và thực tập tại đây, và xin cảm ơn Prof. Đỗ Thanh Nghị – cán bộ hướng dẫn đã giành thời gian quý báu để hướng dẫn cho em trong suốt thời gian thực tập thực tế.

Thông qua quá trình thực tập thực tế, em phần nào củng cố và tích lũy được những kiến thức và kinh nghiệm mà cán bộ hướng dẫn đã truyền dạy cho em, và những đồng đội hỗ trợ hết mình và thân thiện.

Em xin chân thành cảm ơn !

Cần Thơ, ngày 28 tháng 07 năm 2021

Người viết

Nguyễn Thanh Huy

MỤC LỤC

PHẦN I. GIỚI THIỆU TỔ CHỨC.....	Error! Bookmark not defined.
PHẦN II. CÔNG VIỆC ĐƯỢC PHÂN CÔNG	Error! Bookmark not defined.
PHẦN III. KẾT QUẢ.....	1
A. Chuẩn bị.....	1
B. Chuẩn bị và xây dựng mô hình nhận diện gương mặt.....	3
C. Mô hình nhận diện gương mặt đeo khẩu trang	6
PHẦN IV. KẾT LUẬN	11
PHẦN V. TÀI LIỆU THAM KHẢO	12

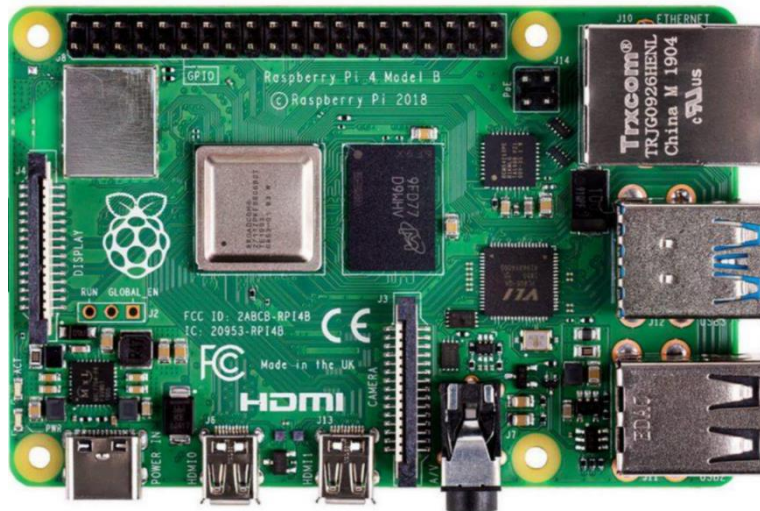
PHẦN III. KẾT QUẢ

A. Chuẩn bị

1. Phần cứng

1.1. Raspberry Pi 4B

Raspberry Pi 4B được trang bị vi xử lý Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5 GHz, RAM 8GB và đặc biệt hỗ trợ chuẩn Wifi 802.11n cùng với Bluetooth 5.0.



1.2. Raspberry Pi Camera

Raspberry Pi Camera được tích hợp camera 5 megapixel, độ phân giải 1080x720p @30fps. Có độ nhạy ánh sáng cao, có thể chụp hình và quay video tốt ở điều kiện ánh sáng khác nhau.

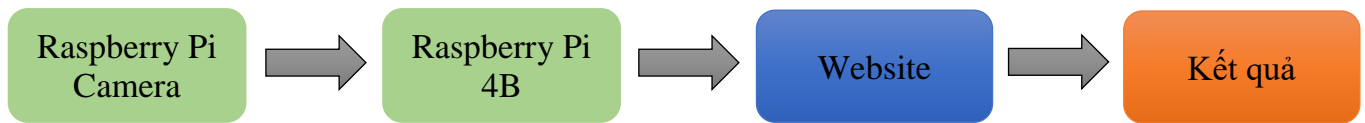


1.3. Thẻ nhớ Micro SD 64GB

2. Phần mềm

- Cài đặt hệ điều hành Raspbian vào thẻ nhớ Micro SD.
- Cài đặt và thiết lập biến môi trường trên web server Apache2.

3. Sơ đồ



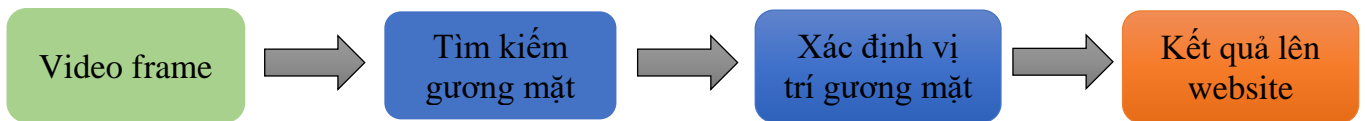
4. Tập dữ liệu

- Hình ảnh gương mặt người có đeo khẩu trang: 5521.
- Hình ảnh gương mặt người không đeo khẩu trang: 5521.

B. Chuẩn bị và xây dựng mô hình nhận diện gương mặt

1. Tài liệu

- Sơ đồ hệ thống



- API Tensorflow.js tf.min().
- API Tensorflow.js Blazeface.

2. Mô hình nhận diện gương mặt được xử lý trước

2.1. API Tensorflow.js Blazeface

Blazeface là một mô hình phát hiện khuôn mặt đơn giản dựa vào hình ảnh bằng cách sử dụng kiến trúc Single Shot Detector với bộ mã hóa tùy chỉnh của Blazeface.



Nhận xét:

- Có thể phát hiện gương mặt người có độ chính xác cao.
- Xác định các vị trí trên gương mặt như: mắt, tai, mũi, miệng.
- Có thể phát hiện các góc khác nhau trên gương mặt người.
- Khó khăn trong việc xác định các khuôn mặt người ở xa.

3. Lập trình video streaming trên web browser kết hợp javascript để nhận diện gương mặt

- Ở file **index.html** thì sẽ gọi hàm từ file **script.js** và gọi đến các API của Tensorflow gồm: hàm tf.min(), Blazeface.

```

<body>
  <h1 style="color:blue">Mô hình nhận diện gương mặt</h1>
  <video id="video" autoplay style="display: none"></video>
  <canvas id="canvas" width="600px" height="400px"></canvas>
</body>

<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs"></script>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow-models/blazeface"></script>
<script src="script.js"></script>
  
```

- Trong file **script.js** tạo biến và xây dựng các thông số cho camera để streaming.

```
let video = document.getElementById("video");
let model;
let canvas = document.getElementById("canvas");
let ctx = canvas.getContext("2d");

const setupCamera = () => {
  navigator.mediaDevices.getUserMedia({
    video: { width: 600, height: 400 },
    audio: false,
  })
  .then((stream) => {
    video.srcObject = stream;
  });
};
```

- Lấy giá trị API Tensorflow trả về để nhận diện gương mặt và xác định vị trí gương mặt bằng viền xanh dương.

```
const detectFaces = async () => {
  const prediction = await model.estimateFaces(video, false);

  // console.log(prediction);

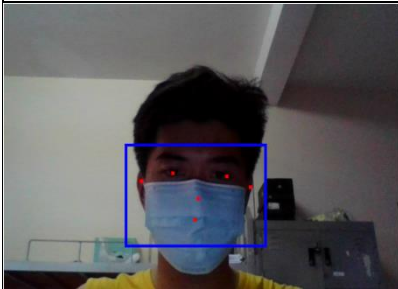
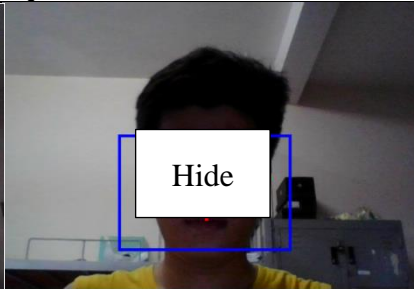
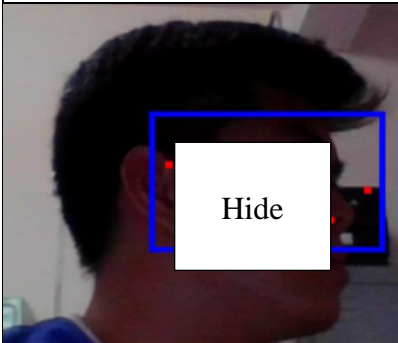

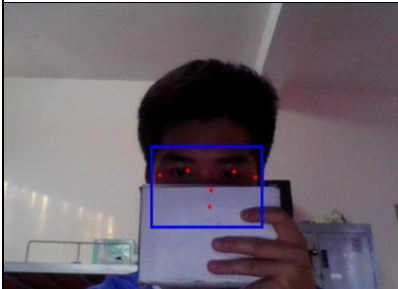
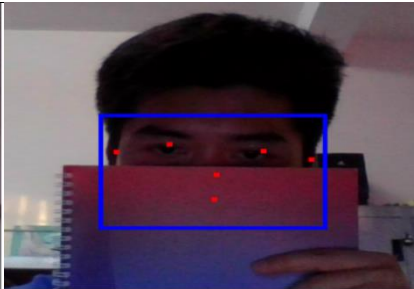
  ctx.drawImage(video, 0, 0, 600, 400);

  prediction.forEach((pred) => {
    ctx.beginPath();
    ctx.lineWidth = "4";
    ctx.strokeStyle = "blue";
    ctx.rect(
      pred.topLeft[0],
      pred.topLeft[1],
      pred.bottomRight[0] - pred.topLeft[0],
      pred.bottomRight[1] - pred.topLeft[1]
    );
    ctx.stroke();
  });
};
```

- Xác định vị trí trên gương mặt người gồm: mắt, mũi, miệng, tai.

```
ctx.fillStyle = "red";
pred.landmarks.forEach((landmark) => {
  ctx.fillRect(landmark[0], landmark[1], 5, 5);
});
});
```


4. Kiểm thử

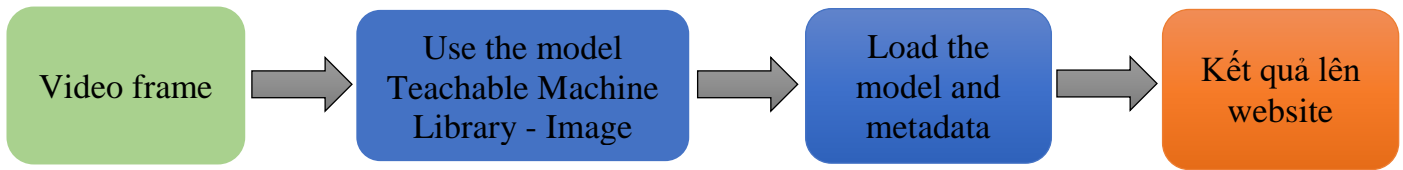
Kết quả		Nhận xét
		Khoảng cách gần, hướng mặt đối diện với camera và nhận diện gương mặt trong điều kiện thiếu sáng khá tốt.
		Phát hiện các hướng của gương mặt nhưng chưa được chính xác cao.
		Trong trường hợp che khuất nửa mặt, nhưng việc xác định gương mặt người khá đúng.

5. Nhận xét

Mô hình phát hiện gương mặt sử dụng API Blazeface của Tensorflow dựa trên kiến trúc Single Shot Detector khá hiệu quả, trong môi trường thiếu ánh sáng cũng như các hướng của gương mặt nhận diện khá ổn. Nhưng sử dụng mô hình phát hiện gương mặt, từ một luồng video trực tiếp trên web browser chạy trên Raspberry Pi 4B không ổn định, khả năng bị suy giảm khung hình cũng như hiện tượng trễ hơi cao.

C. Mô hình nhận diện gương mặt đeo khẩu trang

1. Sơ đồ hệ thống

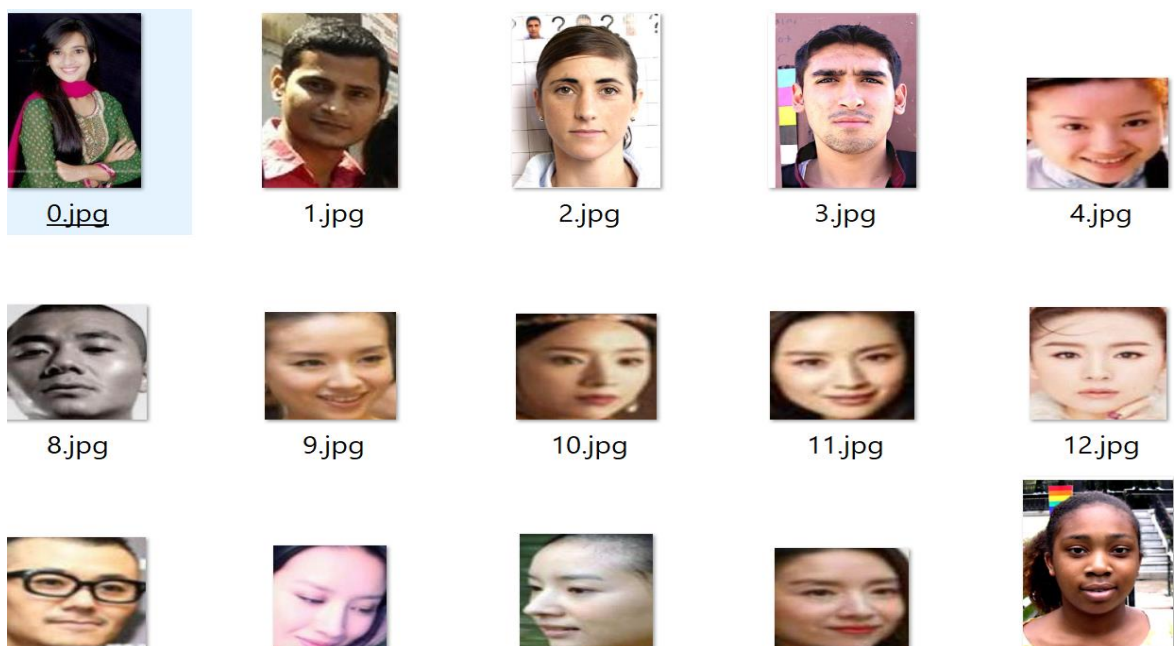


2. Tập dữ liệu

- Tập dữ liệu được chia làm 2 bộ dataset: đào tạo (85%), kiểm tra (15%)
- Số lượng hình ảnh trong tập dữ liệu, xây dựng mô hình thành 2 lớp:
 - Mask: 5521**



- No Mask: 5521**



3. Teachable Machine

Teachable Machine là một công cụ dựa trên web giúp cho mọi người tạo mô hình máy học nhanh chóng, dễ dàng và có thể truy cập được cho tất cả mọi người. Để huấn luyện máy tính có thể nhận dạng được hình ảnh, đồ vật hoặc gương mặt người có đeo khẩu trang hay không, thì chúng ta phải tải ảnh lên Teachable Machine để huấn luyện mô hình dựa vào hình ảnh.

Xây dựng mô hình nhận diện gương mặt đeo khẩu trang trên web browser:

- Ở file **facemaskdetection.html** gọi đến các API của Teachable Machine Library – Image, Tensorflow.

```
<head>
  <title>Face Mask Detection</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <link rel="stylesheet" type="text/css" href="styles.css" media="screen" />
  <div style="font-size:30px; color:red;">Mô hình nhận diện gương mặt đeo
khẩu trang</div>
  <button type="button" onclick="init()">Bắt đầu!</button>
  <div id="webcam-container"></div>
  <div id="label-container" style="font-size:30px; color:red;"></div>
  <script
src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@0.8/dist/teachablemachine-
image.min.js"></script>
```

- URL liên kết đến mô hình huấn luyện được tạo trên Teachable Machine và tải model cùng với metadata.

```
const URL = "https://teachablemachine.withgoogle.com/models/VAOMGfW6K/";
let model, webcam, labelContainer, maxPredictions;
async function init() {
  const modelURL = URL + "model.json";
  const metadataURL = URL + "metadata.json";
  model = await tmImage.load(modelURL, metadataURL);
  maxPredictions = model.getTotalClasses();
```

- Thiết lập webcam bao gồm: chiều dài và rộng khung webcam, yêu cầu quyền truy cập camera trên web browser.

```
const flip = true;
webcam = new tmImage.Webcam(600, 600, flip);
await webcam.setup();
await webcam.play();
window.requestAnimationFrame(loop);
```

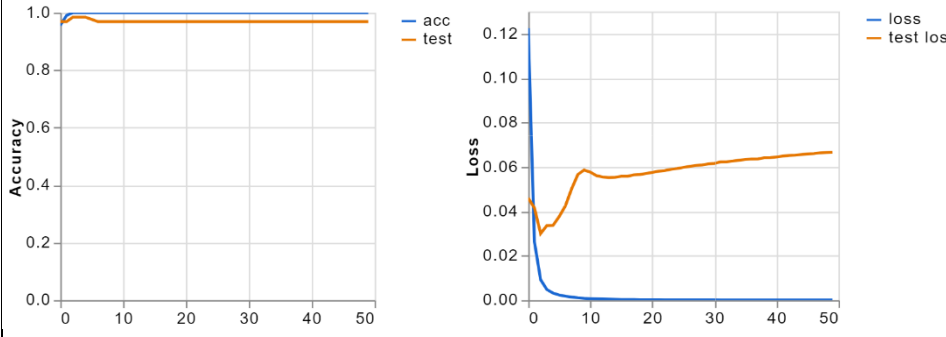
- Hiện thị số nhãn (Mask, No Mask) đã được huấn luyện tại Teachable Machine.

```
document.getElementById("webcam-container").appendChild(webcam.canvas);
labelContainer = document.getElementById("label-container");
for (let i = 0; i < maxPredictions; i++) {
    labelContainer.appendChild(document.createElement("div"));
}
```

- Cập nhật khung hình của webcam và dự đoán tỉ lệ trùng khớp của 2 nhãn(Mask, No Mask).

```
async function loop() {
    webcam.update();
    await predict();
    window.requestAnimationFrame(loop);
}
async function predict() {
    const prediction = await model.predict(webcam.canvas);
    for (let i = 0; i < maxPredictions; i++) {
        const classPrediction =
            prediction[i].className + ": " + (prediction[i].probability*100).toFixed(2)+"%";
        labelContainer.childNodes[i].innerHTML = classPrediction;
    }
}
```

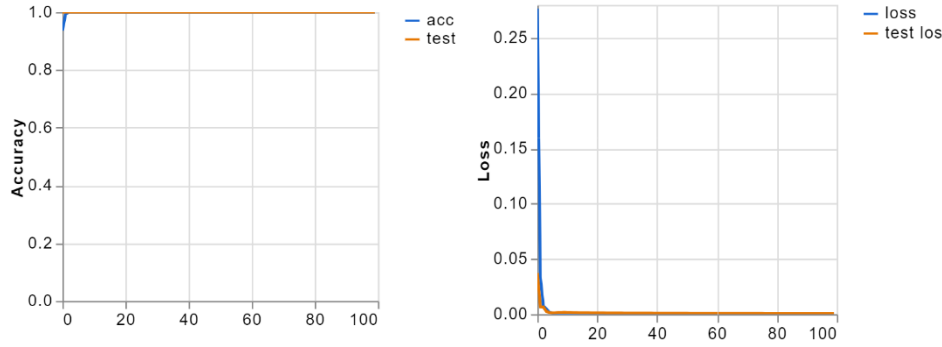
4. Huấn luyện mô hình

Kết quả đồ thị(train set + Test set)	
Độ chính xác(Test set)	96,52%
Mô hình	Teachable Machine

Nhận xét:

- Bộ kiểm tra thử, lại cho đồ thị dự đoán về độ lỗi (Loss) cao và không bám sát theo mô hình và cần lọc hình ảnh sai lệch và huấn luyện lại mô hình.




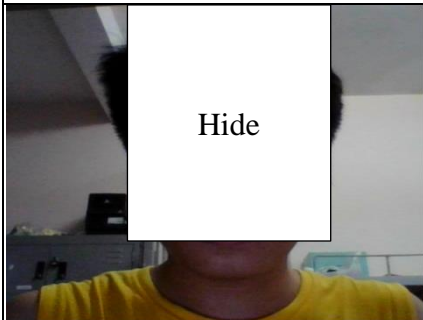

5. Điều chỉnh dữ liệu (dataset) và huấn luyện lại mô hình

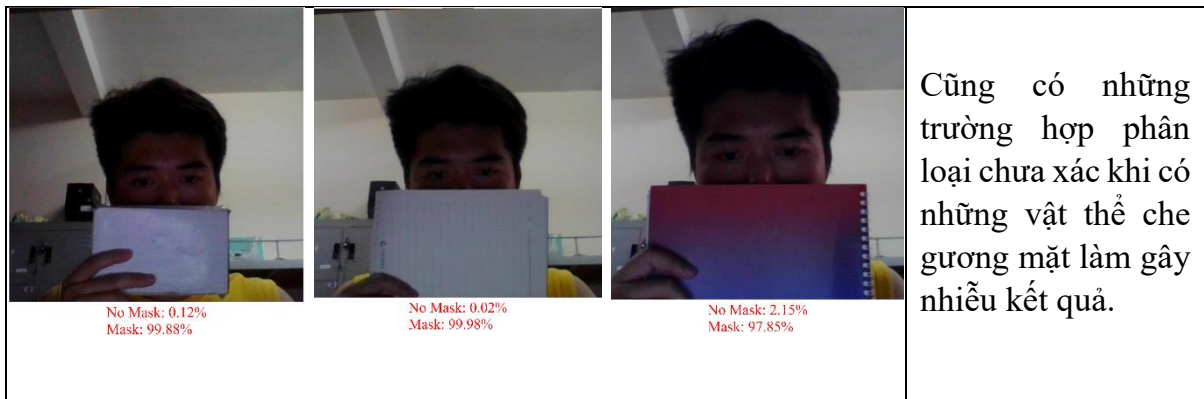
Kết quả đồ thị(train set + Test set)	
Độ chính xác(Test set)	99%
Mô hình	Teachable Machine

Nhận xét:

- Sau khi có sự điều chỉnh, cải thiện dataset và train lại model thì độ chính xác được tăng lên rất cao và dự đoán độ lỗi (Loss) được giảm xuống mức thấp nhất.

6. Kiểm thử trên webcam

Kết quả			Nhận xét
			Khoảng cách gần, hướng mặt đối diện và các hướng của gương mặt đeo khẩu trang nhận diện khá tốt.
			Phát hiện chính xác gương mặt người không đeo khẩu trang.



7. Nhận xét

Hiệu suất khi sử dụng Raspberry Pi Camera kết nối với Raspberry Pi 4B để nhận diện gương mặt đeo khẩu trang thấp, tốc độ xử lý chậm, có hiện tượng trễ frame. Về độ chính xác thì nằm ở mức cao. Tuy nhiên vẫn còn một số trường hợp ảnh hưởng đến độ chính xác khi nhận diện gương mặt đeo khẩu trang gồm:

- Khó phát hiện gương mặt khi ở cách quá xa.
- Có hơn 1 gương mặt cùng lúc trong khung hình thì có thể dẫn đến sự sai lệch kết quả nhận diện gương mặt có đeo khẩu trang hay không đeo khẩu trang.
- Một số vật thể che ngang mặt gây nhầm lẫn giữa việc có đeo khẩu trang và không đeo khẩu trang.

PHẦN IV. KẾT LUẬN

Sau thời gian thực tập thực tế tại Khoa Công nghệ thông tin & Truyền thông, em cảm thấy có một chút xiu tự tin để tốt nghiệp ra trường và đi xin việc làm. Khoa Công nghệ thông tin & Truyền thông và cán bộ hướng dẫn Prof. Đỗ Thanh Nghị đã tạo điều kiện cho em được tham gia vào một dự án thực tế và giúp em ôn lại những kiến thức đã quên. Ngoài việc giúp em nâng cao về mặt kiến thức mà còn hướng dẫn cách làm việc chuyên nghiệp và cộng tác cùng các đồng đội khi làm việc nhóm. Tuy kết quả mô hình còn nhiều thiếu sót, nhưng có thể một ngày nào đó không xa, có thể đem ra nghiên cứu và cải thiện thêm.

PHẦN V. TÀI LIỆU THAM KHẢO

- [1] <https://github.com/tensorflow/tfjs-models/tree/master/blazeface>
- [2] <https://www.geeksforgeeks.org/tensorflow-js-tf-min-function/>
- [3] <https://www.npmjs.com/package/@teachablemachine/image>
- [4] <https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/image>