

TRƯỜNG ĐẠI HỌC THỦY LỢI KHOA CÔNG NGHỆ THÔNG TIN

---&---



BÁO CÁO BÀI TẬP LỚN MÔN HỌC HỌC MÁY

Đề tài:

ÁP DỤNG SVM CHO BÀI TOÁN PHÂN NHÓM CHỮ SỐ VIẾT TAY

Nhóm sinh viên thực hiện: Nhóm 17

Lớp: 60TH4

Thành viên:

Đào Thùy Dương - 1851061470

Ngô Thị Huệ - 1851061355

Giảng viên hướng dẫn: PGS.TS. Nguyễn Hữu Quỳnh

Hà Nội, ngày 1 tháng 11 năm 2021

MỤC LỤC

LỜI MỞ ĐẦU	3
PHẦN I: TỔNG QUAN	4
1.Giới thiệu về học máy	4
2. Trình bày bài toán.....	7
3.Trình bày phương pháp học máy được sử dụng trong bài toán.....	7
PHẦN II: THỰC NGHIỆM	9
1. Mô tả tập ví dụ huấn luyện dữ liệu và tập test	9
2. Mô tả chương trình demo.....	9
<i>2.1: Train và test</i>	<i>9</i>
<i>2.2: Xây dựng chương trình nhập dữ liệu và dự đoán kết quả</i>	<i>10</i>
PHẦN III: KẾT LUẬN	14
Tài liệu tham khảo	14

LỜI MỞ ĐẦU

Nhận dạng là bài toán xuất hiện cách đây khá lâu và vẫn luôn thu hút được nhiều sự quan tâm, nghiên cứu. Đặc biệt là trong vài thập niên gần đây, do sự thúc đẩy của quá trình tin học hoá trong mọi lĩnh vực, bài toán nhận dạng không còn dừng lại ở mức độ nghiên cứu nữa mà nó trở thành một lĩnh vực để áp dụng vào thực tế. Các bài toán nhận dạng đang được ứng dụng trong thực tế hiện nay tập trung vào nhận dạng mẫu, nhận dạng tiếng nói và nhận dạng chữ. Trong số này, nhận dạng chữ số là bài toán được quan tâm rất nhiều và cũng đã đạt được nhiều thành tựu rực rỡ. Các ứng dụng có ý nghĩa thực tế lớn có thể kể đến như: nhận dạng chữ in dùng trong quá trình sao lưu sách báo trong thư viện, nhận dạng chữ số viết tay dùng trong việc phân loại thư ở bưu điện, thanh toán tiền trong nhà băng và lập thư viện sách cho người mù (ứng dụng này có nghĩa: scan sách bình thường, sau đó cho máy tính nhận dạng và trả về dạng tài liệu mà người mù có thể đọc được).

Xuất phát từ yêu cầu thực tế, đang rất cần có nhưng nghiên cứu về vấn đề này. Chính vì vậy chúng em đã chọn đề tài nhận dạng ký tự viết tay báo cáo bài tập lớn với mong muốn phần nào áp dụng vào bài toán thực tế.

PHẦN I: TỔNG QUAN

1. Giới thiệu về học máy

a, Lịch sử học máy

- 1950 – Alan Turing tạo ra phép thử Turing (Turing Test) để xác định xem một máy tính có trí thông minh thực sự hay không. Để vượt qua các phép thử, máy tính phải có khả năng đánh lừa một người tin rằng nó cũng là con người.
- 1952 – Arthur Samuel đã viết chương trình máy học đầu tiên. Chương trình này là trò cờ đam, và máy tính IBM càng chơi càng tiến bộ, học được các nước đi tạo lập nên chiến lược để chiến thắng và tích hợp các nước đi đó vào chương trình của mình.
- 1957 – Frank Rosenblatt thiết kế hệ thần kinh (neural network) đầu tiên cho máy tính (gọi là “perceptron”) mô phỏng các quá trình tư duy của não người.
- 1967 – Thuật toán “điểm lân cận gần nhất” (nearest neighbor) được viết ra cho phép các máy tính bắt đầu sử dụng dạng mẫu nhận thức rất cơ bản. Thuật toán này có thể được sử dụng để vẽ một tuyến đường cho nhân viên bán hàng lưu động, bắt đầu từ một thành phố ngẫu nhiên nhưng đảm bảo họ tới tất cả các thành phố trong một chuyến đi ngắn.
- 1979 – Sinh viên Đại học Stanford phát minh ra xe “Stanford Cart” có thể tự xác định được vật cản trong phòng.
- 1981 – Gerald Dejong giới thiệu khái niệm Học qua giải thích (Explanation Based Learning – EBL), trong đó một máy tính phân tích dữ liệu huấn luyện và tạo ra một quy tắc chung mà nó có thể làm theo bằng cách loại bỏ các dữ liệu không quan trọng.
- 1985 – Terry Sejnowski phát minh ra NetTalk, học cách phát âm các giống như một đứa trẻ.
- 1990s – Các nghiên cứu về machine learning chuyển từ cách tiếp cận hướng-kiến-thức sang cách tiếp cận hướng-dữ-liệu. Các nhà khoa học bắt đầu tạo các chương trình cho máy tính để phân tích một lượng lớn dữ liệu và rút ra kết luận – hay “học” – từ các kết quả.
- 1997 – Máy tính Deep Blue của IBM đánh bại nhà vô địch cờ vua thế giới.
- 2006 – Geoffrey Hinton tạo ra thuật ngữ “học sâu” (deep learning) để giải thích các thuật toán mới cho phép các máy tính “nhìn thấy” và phân biệt các đối tượng và văn bản trong hình ảnh và video.

- 2010 – Máy Microsoft Kinect có thể theo dõi 20 tính năng của con người ở tốc độ 30 lần mỗi giây, cho phép mọi người tương tác với máy tính thông qua các động tác và cử chỉ.
- 2011 – Máy IBM Watson đánh bại các đối thủ con người tại Jeopardy.
- 2011 – Google Brain được phát triển, và hệ thần kinh sâu (deep neural network) của nó có thể học để phát hiện và phân loại các đối tượng bằng với khả năng của một chú mèo.
- 2012 – X Lab của Google phát triển một thuật toán machine learning có khả năng tự duyệt video YouTube để xác định các video có chứa hình ảnh mèo.
- 2014 – Facebook phát triển DeepFace, một thuật toán phần mềm có thể nhận diện hoặc xác minh các cá nhân trong các bức ảnh ở mức độ ngang với con người.
- 2015 – Amazon ra mắt nền tảng machine learning riêng của mình.
- 2015 – Microsoft tạo ra bộ công cụ máy học được phân bổ (Distributed Machine Learning Toolkit), cho phép phân bổ hiệu quả machine learning trên nhiều máy tính
- 2015 – Hơn 3.000 nhà nghiên cứu về AI và Robotics, được Stephen Hawking, Elon Musk và Steve Wozniak (và nhiều người khác) hậu thuẫn, đã ký một bức thư ngỏ cảnh báo nguy cơ của vũ khí tự hành có thể lựa chọn và nhắm vào các mục tiêu mà không cần sự can thiệp của con người.
- 2016 – Thuật toán trí tuệ nhân tạo (artificial intelligence algorithm) của Google đã đánh bại một cầu thủ chuyên nghiệp trong trò cờ Go của Trung Quốc, vốn được coi là trò cờ phức tạp nhất thế giới và khó hơn cờ vua gấp nhiều lần. Thuật toán AlphaGo do Google DeepMind phát triển đã giành chiến thắng năm trên năm trận đấu Go.

b) Ưu, nhược điểm của học có giám sát và không giám sát:

	Học có giám sát	Học không giám sát
Khái niệm	Học tập có giám sát như tên gọi cho biết sự hiện diện của người giám sát như một giáo viên. Về cơ bản, học có giám sát là cách học mà chúng ta dạy hoặc đào tạo máy bằng cách sử dụng dữ liệu được gắn nhãn tốt.	Học không giám sát là việc đào tạo máy sử dụng thông tin không được phân loại cũng như không được gắn nhãn và cho phép thuật toán hoạt động trên thông tin đó mà không cần hướng dẫn. Ở đây, nhiệm vụ của

		máy là nhóm các thông tin chưa được sắp xếp theo những điểm tương đồng, kiểu mẫu và sự khác biệt mà không cần đào tạo trước dữ liệu.
Ưu điểm	<p>Học tập có giám sát cho phép thu thập dữ liệu và tạo ra dữ liệu đầu ra từ những kinh nghiệm trước đó.</p> <p>Giúp tối ưu hóa các tiêu chí hiệu suất với sự trợ giúp của kinh nghiệm.</p> <p>ML có giám sát giúp giải quyết nhiều loại vấn đề tính toán trong thế giới thực.</p>	<p>Mô hình học máy không giám sát tìm thấy tất cả các loại mẫu chưa biết trong dữ liệu. Do đó, nó có thể giúp bạn phát hiện các tính năng có thể hữu ích trong việc phân loại dữ liệu.</p> <p>Trong mô hình học tập không giám sát, không cần phải gán nhãn các đầu vào dữ liệu. Và dữ liệu không có nhãn, nhìn chung, dễ lấy hơn, vì nó có thể được lấy trực tiếp từ máy tính mà không cần thêm sự can thiệp của con người. Điều này làm cho việc học không có giám sát trở thành một mô hình ít phức tạp hơn so với các kỹ thuật học có giám sát.</p>
Nhược điểm	<p>Phân loại dữ liệu lớn có thể là một thách thức.</p> <p>Đào tạo cho việc học có giám sát cần rất nhiều thời gian tính toán, vì vậy, nó đòi hỏi rất nhiều thời gian.</p>	<p>Máy bị hạn chế tự tìm kiếm cấu trúc ẩn trong dữ liệu không được gán nhãn.</p> <p>Tính toán phức tạp</p> <p>Độ chính xác thấp.</p>

2. Trình bày bài toán

Nhận dạng chữ in: đã được giải quyết gần như trọn vẹn (sản phẩm FineReader 9.0 của hãng ABBYY có thể nhận dạng chữ in theo 192 ngôn ngữ khác nhau, phần mềm nhận dạng chữ Việt in VnDOCR 4.0 của Viện Công nghệ Thông tin Hà Nội có thể nhận dạng được các tài liệu chứa hình ảnh, bảng và văn bản với độ chính xác trên 98%).

Nhận dạng chữ số viết tay: vẫn còn là vấn đề thách thức lớn đối với các nhà nghiên cứu. Bài toán này chưa thể giải quyết trọn vẹn được vì nó hoàn toàn phụ thuộc vào người viết và sự biến đổi quá đa dạng trong cách viết và trạng thái sức khỏe, tinh thần của từng người viết

Input: Tập dữ liệu huấn luyện gồm hình ảnh chứa ký tự được chuyển về dạng nhị phân, ảnh sẽ được lưu dưới dạng ma trận điểm.

Output: Nhận dạng được các đặc điểm của từng ký tự, phân loại chúng và tạo ra các thông tin cần thiết để nhận dạng các ký tự.

3.Trình bày phương pháp học máy được sử dụng trong bài toán

a. Giới thiệu

- SVM là một khái niệm trong thống kê và khoa học máy tính cho một tập hợp các phương pháp học có giám sát để phân loại và phân tích hồi quy
- Support Vector Machine (SVM) là một thuật toán thuộc nhóm Supervised Learning (Học có giám sát) dùng để phân chia dữ liệu (Classification) thành các nhóm riêng biệt.
- Hình dung ta có bộ data gồm các điểm xanh và đỏ đặt trên cùng một mặt phẳng. Ta có thể tìm được đường thẳng để phân chia riêng biệt các bộ điểm xanh và đỏ.
- Với những bộ dữ liệu phức tạp hơn thì ta cần dùng thuật toán để ánh xạ bộ dữ liệu đó vào không gian nhiều chiều hơn từ đó tìm ra được siêu mặt phẳng để phân chia.

b. Tối ưu trong thuật toán SVM

- Ta có thể tìm ra được rất nhiều đường thẳng phân chia tách biệt 2 cụm điểm khác nhau. Nhưng cần phải tìm được đường thẳng tối ưu
- Đường tối ưu là đường tạo cho ta có cảm giác 2 lớp dữ liệu nằm cách xa nhau và cách xa đường đó nhất. Và trong SVM sử dụng thuật ngữ Margin

c. Margin

- Margin là khoảng cách giữa siêu phẳng (trong trường hợp không gian 2 chiều là đường thẳng) đến 2 điểm dữ liệu gần nhất tương ứng với 2 phân lớp
- SVM cố gắng tối ưu thuật toán bằng cách tìm cách maximize giá trị margin này, từ đó tìm ra siêu phẳng đẹp nhất để phân 2 lớp dữ liệu

d. Support Vector

- Trong không gian 2 chiều, đường thẳng phân chia có phương trình:

$$W_1x_1 + w_2x_2 + b = 0$$
- với cặp dữ liệu (x_n, y_n) bất kỳ, khoảng cách từ điểm đó tới mặt phân chia là:

$$\frac{y_n(W^T x_n + b)}{\|W\|}$$

y_n luôn cùng dấu với $x_n \rightarrow y_n$ cùng dấu với $(W^T x_n + b)$ nên tử số luôn là một số không âm.

- Margin được tính là khoảng cách gần nhất từ một điểm tới mặt phẳng đó:

$$\text{Margin} = \min \frac{y_n(W^T x_n + b)}{\|W\|}$$

- Bài toán tối ưu là bài toán tìm w và b sao cho margin đạt giá trị lớn nhất

$$\begin{aligned} (w, b) &= \operatorname{argmax} \left\{ \min \frac{y_n(W^T x_n + b)}{\|W\|} \right\} \\ &= \operatorname{argmax} \left\{ \frac{1}{\|w\|^2} \min y_n(W^T x_n + b) \right\} \end{aligned}$$

PHẦN II: THỰC NGHIỆM

1. Mô tả tập ví dụ huấn luyện dữ liệu và tập test

Trong bài tập lớn lần này, chúng em sử dụng tập dữ liệu MNIST. MNIST là tập dữ liệu chữ viết từ 0 đến 9. Trong đó, mỗi hình là một ảnh đen trắng chứa một số được viết tay có kích thước là 28x28. Bộ dataset vô cùng đồ sộ với khoảng 60k data training và 10k data test và được sử dụng phổ biến trong các thuật toán nhận dạng ảnh.

Website chính thức của tập dữ liệu:

<http://yann.lecun.com/exdb/mnist/>

2. Mô tả chương trình demo

2.1: Train và test

a, Import các thư viện

Ở đây, chúng em sử dụng thư viện mnist và thư viện SVM có sẵn của scikit-learn là SVC. Chú ý nếu chưa cài scikit-learn, mnist trong máy, ta có thể cài đặt đơn giản bằng pip (thay bằng pip3 nếu muốn cài cho Python 3).

```
1 import mnist # pip install mnist
2 from matplotlib import pyplot
3 from sklearn.svm import LinearSVC
4 import pickle
5
```

b, Mô-đun MNIST

Mô-đun MNIST cung cấp tập dữ liệu mnist. Nhưng nó là mảng ba chiều nên ta phải định hình trước khi sử dụng

```
6 #lấy tập dữ liệu
7 x_train = mnist.train_images()
8 y_train = mnist.train_labels()
9 x_test = mnist.test_images()
10 y_test = mnist.test_labels()
11
12 #Định hình bộ dữ liệu
13 n_samples, nx, ny = x_train.shape
14 n_samples_test, nx_test, ny_test = x_test.shape
15 x_train = x_train.reshape((n_samples, nx*ny))
16 x_test_ = x_test.reshape((n_samples_test, nx_test*ny_test))
17
```

c, Train và test

```
18  #bắt đầu đào tạo và tính độ chính xác, lưu mô hình ra file
19  svm = LinearSVC()
20  for i in range(10):
21      svm.fit(x_train, y_train)
22      acc = svm.score(x_train, y_train)
23      if acc >= 0.87:
24          print("Saving...")
25          model_file = open("mnist.pickle", "wb")
26          pickle.dump(svm, model_file)
27          print("Độ chính xác: ", acc)
28          break
29
```

Sau khi đào tạo, ta lấy ra độ chính xác của mô hình và lưu lại model để sử dụng cho bài toán

```
Saving...
Độ chính xác:  0.8769833333333333
```

2.2: Xây dựng chương trình nhập dữ liệu và dự đoán kết quả

a, Import các thư viện

```
1  import pygame
2  import numpy
3  import pickle
4  from tkinter import messagebox
5  from tkinter import *
6  import sys
7
```

b, Xây dựng chương trình

- Class scaledata

```
8 class ScaleData:
9     def __init__(self, mouse_pos, thick_x, thick_y):
10         self.mouse_pos = mouse_pos
11         self.width = 28
12         self.height = 28
13         self.grid_width = 500
14         self.grid_height = 500
15         self.thick_x = thick_x
16         self.thick_y = thick_y
17     def scale_img_data(self):
18         scale_grid = []
19         scale_grid_row = []
20         differ_x = self.width / self.grid_width
21         differ_y = self.height / self.grid_height
22         scale_width, scale_height = round(self.thick_x*differ_x), round(self.thick_y*differ_y)
23
24         for y in range(self.height):
25             for x in range(self.width):
26                 scale_grid_row.append(0)
27             scale_grid.append(numpy.array(scale_grid_row))
28             scale_grid_row.clear()
29         scale_grid = numpy.array(scale_grid)
30
31         for mouse_pos in self.mouse_pos:
32             x_pos, y_pos = mouse_pos
33             scale_x, scale_y = round(x_pos*differ_x), round(y_pos*differ_y)
34
35             for h in range(scale_height):
36                 for w in range(scale_width):
37                     scale_grid[scale_y+h][scale_x+w] = 255
38         return scale_grid
39
```

- Nhận đầu vào

```
40 def mouse_input():
41     is_clicked = pygame.mouse.get_pressed(num_buttons=5)
42     mouse_pos = pygame.mouse.get_pos()
43     if is_clicked[0]:
44         return mouse_pos
45
```

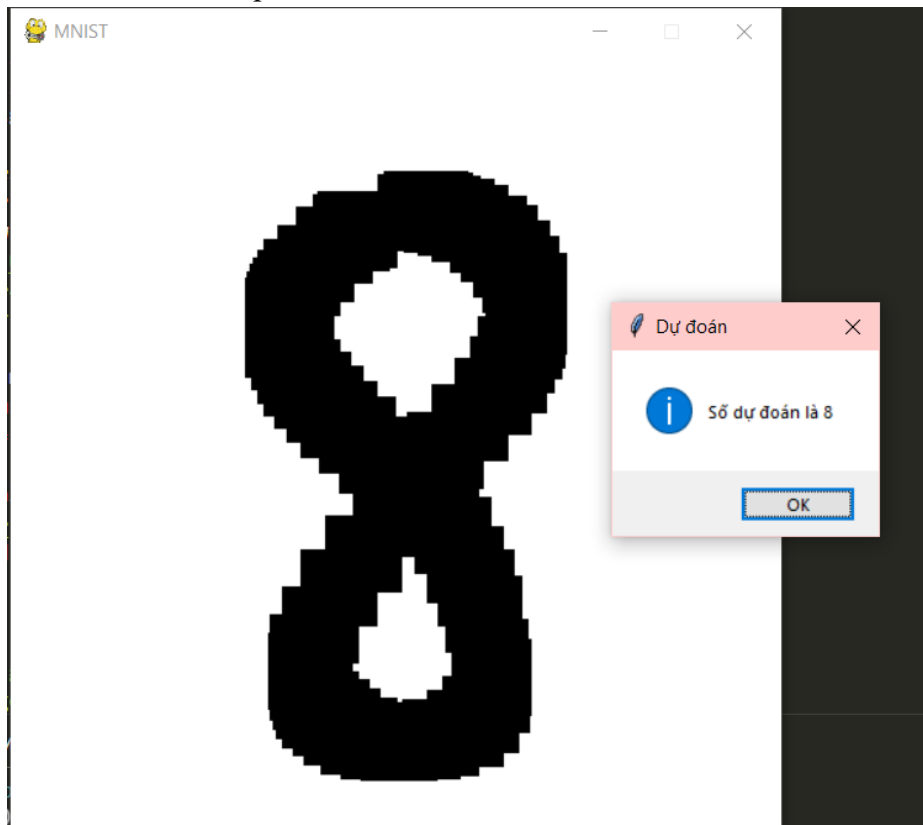
- Nhập dữ liệu và đoán kết quả

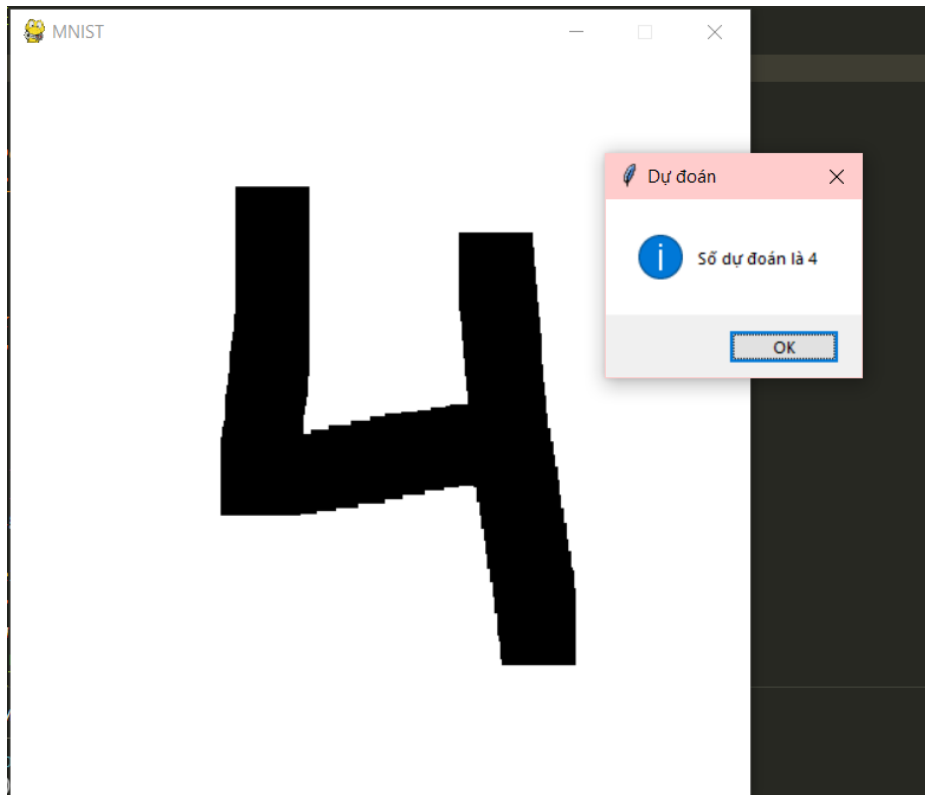
```

46 class MnistGui:
47     def __init__(self):
48         pygame.display.set_caption("Đào Thùy Dương - Ngô Thị Huệ")
49         self.width = self.height = 500
50         self.bg_color = (255, 255, 255)
51         self.screen = pygame.display.set_mode((self.width, self.height))
52         self.cursor_pos = []
53         self.thickness_x = self.thickness_y = 50
54         self.root = Tk()
55         self.root.eval(f"tk::PlaceWindow {self.root.winfo_toplevel()} center")
56         self.root.withdraw()
57
58     def run(self):
59         #chạy toàn bộ vòng lặp chính
60         while True:
61             self.mouse_pos = mouse_input()
62             for event in pygame.event.get():
63                 if event.type == pygame.QUIT:
64                     sys.exit(0)
65                 if event.type == pygame.KEYDOWN:
66                     try:
67                         scaled_data = numpy.array([ScaleData(self.cursor_pos, thick_x=self.thickness_x, thick_y=self.thickness_y).scale_img_data()])
68                         n_samples, nx, ny = scaled_data.shape
69                         scaled_data = scaled_data.reshape((n_samples, nx*ny))
70                         model_file = open("mnist.pickle", "rb")
71                         model = pickle.load(model_file)
72
73                         predict = model.predict(scaled_data)
74                         messagebox.showinfo("Dự đoán", f"Số dự đoán là {predict[0]}")
75                     except Exception:
76                         messagebox.showerror("Error", "Hệ thống không bắt được số, vui lòng thử lại!")
77             self.root.quit()

```

- Mô tả kết quả dự đoán





PHẦN III: KẾT LUẬN

Qua quá trình học tập và nghiên cứu, cùng với sự giúp đỡ của thầy Nguyễn Hữu Quỳnh , nhóm em đã hoàn thành đề tài môn học máy. Bài báo này đã đề xuất mô hình nhận dạng chữ số viết tay rời rạc trên cơ sở phương pháp máy hỗ trợ véc tơ . Các kết quả thực nghiệm cho thấy mô hình này có kết quả nhận dạng chính xác hơn so với các mô hình khác.

Tuy nhiên, khi áp dụng SVM vào bài toán nhận dạng cũng gặp phải một số hạn chế nhất định: số lượng máy hỗ trợ véc tơ thu được sau khi huấn luyện sẽ dẫn đến việc phân lớp chậm.

Qua bài tập lớn này, nhóm em đã tiếp thu được nhiều kiến thức liên quan đến môn học và áp dụng được các thuật toán vào các bài toán thực tế. Ngoài ra học phần còn giúp chúng em có thể nâng cao kỹ năng làm việc nhóm. Vì thời gian và kinh nghiệm còn hạn chế, bài làm của chúng em vẫn còn nhiều hạn chế và thiếu sót. Nhóm em mong nhận được đánh giá từ thầy cô để giúp nhóm em có thể hoàn thiện hơn bài tập lớn này.

Nhóm em xin chân thành cảm ơn thầy!

Tài liệu tham khảo

1. <https://machinelearningcoban.com>
2. <https://www.w3schools.com/python/>
3. <https://realpython.com/pygame-a-primer/>