

Face Mask Detection Using Convolutional Neural Network and Machine Learning

Nguyễn Lê Thanh^{1,2}, Lê Thị Phương Thanh^{1,2}, Nguyễn Ngọc Thủy^{1,2}
19522238@gm.uit.edu.vn, 19522231@gm.uit.edu.vn, 19522324@gm.uit.edu.vn

¹ University of Information Technology, Ho Chi Minh City, Vietnam

² Vietnam National University, Ho Chi Minh City, Vietnam

Abstract. Đại dịch Covid đang hoành hành bởi tác động của các biến thể của virus Corona, khả năng lây lan mạnh mẽ và thích ứng với vaccine của các biến thể ngày càng mạnh mẽ hơn. Nó lây lan từ người sang người thông qua đường hô hấp, để lại những mầm bệnh trên nhiều bề mặt khác nhau, sau đó những người khác hấp thụ vào người và bị lây nhiễm. Việc bảo vệ bản thân tránh khỏi mầm bệnh trở nên cần thiết hơn bao giờ hết. Thực hiện quy tắc 5K, duy trì khoảng cách khi giao tiếp, đeo khẩu trang đúng cách,... và luôn đeo khẩu trang khi ra khỏi nhà. Việc sử dụng khẩu trang mọi lúc mọi nơi trở thành điều quan trọng trong thời điểm này. Tuy nhiên, việc giám sát người dân về độ che phủ của khẩu trang trở nên khó khăn trong các khu đô thị và những nơi tụ tập đông người. Do đó các quốc gia phát triển về công nghệ thông tin chuyển sang sử dụng AI để xác định việc đeo/không đeo hay đeo khẩu trang sai cách đối với các cá nhân ở nơi công cộng để tạo điều kiện cho việc truy dấu, tiếp xúc, theo dõi sự lây lan của COVID-19 và dự đoán các khu vực có khả năng bùng phát dịch bệnh. Bài báo cáo này đề xuất phương pháp phát hiện người có đeo khẩu trang hay không và có đeo đúng cách.

Keywords: Face Mask Detection, Mask classification, Covid-19, Convolutional Neural Network, Machine learning, K-Nearest Neighbors, Decision Trees, Naive Bayes, Support Vector Machine.

1 Introduction

Vào năm 2020, sự lây lan nhanh chóng của COVID-19 đã khiến cho Tổ chức Y tế Thế giới tuyên bố COVID-19 là một đại dịch trên toàn thế giới. Tuy nhiên, các cơ quan chức năng gặp phải một số khó khăn trong quá trình giám sát một số lượng lớn dân cư có thói quen sinh hoạt khác người [5] [1]. Các nhà chức trách cần một giải pháp để có thể kiểm soát việc thực thi luật một cách hợp lệ, bắt đầu từ việc cung cấp dữ liệu một cách nhanh chóng và chính xác. Một trong những giải pháp là sử dụng nhận dạng khẩu trang tự động theo khu vực để phân biệt giữa những người đeo khẩu trang và những người không đeo [7]. Chính vì thế, mục tiêu của bài báo này là tạo ra một mô hình xác định chính

xác xem một cá nhân có đeo khẩu trang, không đeo khẩu trang hay đeo khẩu trang không đúng cách hay không. Hiện tại, bài báo[5] “A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic”, đã tạo ra một mô hình phát hiện phạm vi che phủ của khẩu trang hoạt động với độ chính xác 100% trên một bộ dữ liệu thử nghiệm [5]. Mặc dù kết quả cho ra có vẻ tốt nhưng nghiên cứu này chỉ giải thích cho phân loại nhị phân giữa đeo khẩu trang và không đeo khẩu trang. Về mặt ứng dụng trong thế giới thực thì có vẻ không được khả quan, bởi vì người ta đã chỉ ra rằng đeo khẩu trang không đúng cách không bảo vệ bạn khỏi bị nhiễm vi-rút COVID-19. Chính vì thế, chúng tôi đã tạo một lớp phân loại thứ ba trong bài báo cáo này, bằng cách kết hợp 2 bộ dữ liệu khẩu trang riêng biệt, sau đó tạo ra bộ dữ liệu mới được gán nhãn WithMask, WithoutMask, IncorrectlyWornMask. Chúng tôi cũng sẽ xây dựng thêm một số bộ phân loại, bao gồm bộ phân loại KNN, Decision Trees, Suport Vector Machine, Naive Bayes và mạng nơ-ron CNN. Điều này sẽ giúp chúng tôi xác định mô hình nào là công cụ dự đoán tốt nhất cho bài toán của chúng tôi và cũng cho phép chúng tôi đánh giá độ chệch lệch trong các thuật toán của mình trên nhiều bộ phân loại khác nhau.

2 Related Works

Bài báo [5] cung cấp cho chúng tôi độ chính xác cơ bản tương đối cao để phát hiện Khẩu trang trên ba bộ dữ liệu Mask /No Mask khác nhau và chứa giải pháp SVM đạt 100% testing accurancy cho các khuôn mặt được gán nhãn trong cơ sở dữ liệu Wild. Nó khám phá các phương pháp học sâu để cải thiện việc trích xuất tính năng và phân loại độ che phủ của khẩu trang và cung cấp một loạt các bộ phân loại học máy cổ điển như Cây quyết định, SVM và Ensemble và các chỉ số hiệu suất tương ứng của chúng (bao gồm cả thời gian tính toán). Tính hiệu quả của nó đã khiến nhiều quốc gia trên khắp thế giới bắt buộc phải đeo khẩu trang khi ở nơi công cộng, tuy nhiên việc giám sát tất cả mọi người đeo khẩu trang trở nên khó khăn trong các khu đô thị và những nơi tụ tập đông người.

Bài báo [5] sử dụng cả phương pháp học máy cổ điển và học sâu để phát triển thuật toán phát hiện khẩu trang như SVM, cây quyết định và ensemble model không trang bị quá nhiều dữ liệu đào tạo. Việc trích xuất các tính năng từ tập dữ liệu hình ảnh theo cách cải thiện tốc độ phân loại và tính toán tổng thể là thách thức chính đối với nghiên cứu này. Nó sẽ ảnh hưởng đến tính hữu ích của thuật toán trong các ứng dụng ở ngoài đời sống. Một khía cạnh mới của bài báo [5] các tác giả đã chọn sử dụng phương pháp trích xuất tính năng của Residual Neural Network để giảm Vanishing Gradient từ bộ dữ liệu và cải thiện kết quả phân loại của chúng.

Loey và các cộng sự [5] kết luận rằng hiện tại, bộ phân loại học chuyển tiếp sâu không mang lại độ chính xác có thể chấp nhận được. Vì thế đã bỏ qua kết quả của bộ phân loại đó. Trong số các bộ phân loại học máy cổ điển được sử dụng, người ta kết luận rằng thuật toán SVM là bộ phân loại tốt nhất vì nó không chỉ có testing accurancy tương đối cao trên mỗi tập dữ liệu mà còn tiêu tốn ít thời gian nhất. Điều thú vị là các tập dữ liệu sử dụng khẩu trang mô

phỏng như tập dữ liệu SFMD dẫn đến training accuracy thấp hơn cho Decision Tree Classifier. Chúng tôi sẽ coi bài báo[5] là bài báo cơ sở trong suốt bài báo cáo và sử dụng nghiên cứu được trình bày ở đây để đánh giá hiệu quả các model của chúng tôi. Điều chú ý ở đây là mặc dù mở rộng về số lượng các mô hình mà bài báo [5] giới thiệu để giải quyết vấn đề, nhưng nó không sao chép bất kỳ mô hình nào. Hầu hết các nghiên cứu liên quan đến điều chỉnh các siêu tham số cho bài báo cáo của chúng tôi được thực hiện độc lập so với bài báo.

3 Dataset

3.1 Task Definition

Trong phần này, chúng tôi tóm tắt nhiệm vụ phát hiện khẩu trang. Nhiệm vụ này mục đích phát hiện xem người đó có đeo khẩu trang hoặc không đeo khẩu trang, hoặc đeo khẩu trang không đúng cách. Về mặt hình thức, nhiệm vụ được mô tả như sau:

Input: Đưa vào hình ảnh con người.

Output: Bộ phân loại dự đoán một trong ba nhãn khác nhau.

- **Có đeo khẩu trang (2: WithMask)** hình ảnh người có đeo khẩu trang và đeo đúng cách.

- **Không đeo khẩu trang (0: WithoutMask)** hình ảnh người không đeo khẩu trang.

- **Đeo khẩu trang sai cách (1: IncorrectlyWornMask)** hình ảnh người có đeo khẩu trang nhưng đeo sai cách.

3.2 Exploratory Data Analysis

Chúng tôi sử dụng tập dữ liệu hình ảnh do Kaggle cung cấp bao gồm có 2 nhãn là Đeo khẩu trang đúng cách (2: WithMask) và Không đeo khẩu trang (0: WithoutMask). Sau đó, chúng tôi tìm thêm 6044 ảnh Đeo khẩu trang sai cách (1:IncorrectlyWornMask) được cung cấp bởi Cornell để tạo thành 1 bộ dữ liệu mới.

Để đào tạo các mô hình với bộ dữ liệu của chúng tôi, trước tiên phải xử lý dữ liệu thô. Sau khi load dữ liệu hình ảnh, chúng tôi đã convert tập X (features) và tập y (target) sang kiểu Numpy Array. Sau đó chúng tôi đã định dạng hình ảnh thành Grayscale và thay đổi kích thước ảnh thành 64x64. Bảng 1 thể hiện bộ dữ liệu của chúng tôi sau khi được xử lý.

Table 1: Bộ dữ liệu với sau khi đã xử lý.

Features (X.shape)	Target (y.shape)	Data shape
(17836, 64, 64)	(17836,)	(17836, 4096)

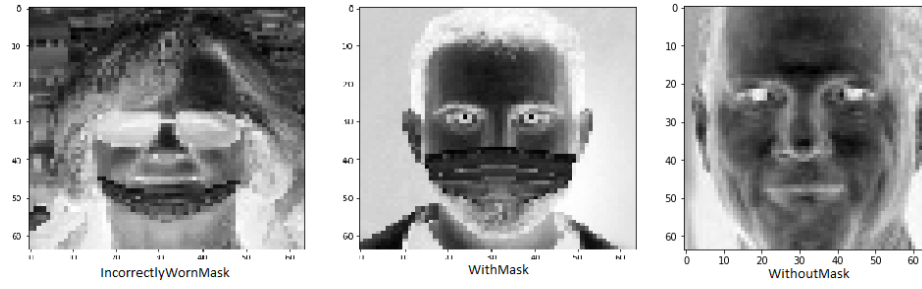


Fig. 1: Một số hình ảnh đã được xử lý.

Tiếp theo, chúng ta sẽ đếm xem mỗi lớp có bao nhiêu tấm ảnh. Với nhãn đeo khẩu trang thì có 5883 ảnh, không đeo khẩu trang có 5909 ảnh và đeo khẩu trang sai cách có 6044 ảnh (bảng 2)

Table 2: Giá trị mỗi lớp phân loại.

Without a mask	Incorrectly Worn Mask	With a mask
5909	6044	5883

Để có thể đưa dữ liệu vào huấn luyện các mô hình học máy cổ điển như: KNN, SVM, Decision tree,... thì tập dữ liệu được convert sang kiểu dataframe.

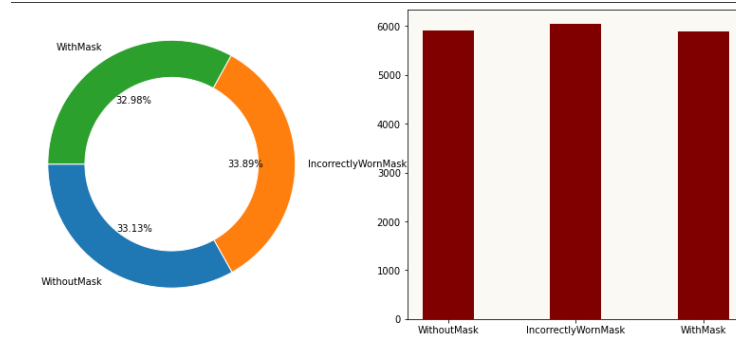


Fig. 2: Visualisation bộ dataset.

Để kiểm tra bias của model, chúng tôi đã tạo thêm 1 tập dữ liệu Test-bias gồm 3 thư mục WithMask, WithoutMask, IncorrectlyWornMask có tổng cộng 509 tấm ảnh được sưu tập từ các hoạt động hằng ngày của chúng tôi và trên internet (bảng 3)

Table 3: Giá trị mỗi lớp phân loại của bộ dữ liệu Test-bias.

Without a mask	Incorrectly Worn Mask	With a mask
152	144	213

4 Our Approach

4.1 Proposed System

Trong phần này, chúng tôi đề xuất cách tiếp cận đơn giản và hiệu quả cho nhiệm vụ này. Không giống như bài báo [5] dùng pretrain model ResNet để trích xuất các tính năng và sử dụng nó đào tạo mô hình. Thay vào đó, chúng tôi tìm hiểu và tự xây dựng mạng CNN để trích xuất các thuộc tính quan trọng trong việc phân loại. Hình 3 cho thấy tổng quan các bước mà chúng tôi sẽ thực hiện trong bài báo cáo này.



Fig. 3: Phương pháp tiếp cận đề xuất của chúng tôi để phát hiện người đeo khẩu trang, không đeo khẩu trang và đeo khẩu trang sai.

4.2 Data Preprocessing

Bước tiền xử lý dữ liệu là một bước quan trọng trong hầu hết các dự án Học máy hoặc Học sâu. Do đó, để làm sạch tốt tập dữ liệu đã cho, chúng tôi đề xuất một quá trình tiền xử lý dữ liệu với các bước như Hình 4 trình bày tổng quan về quy trình Data Preprocessing.

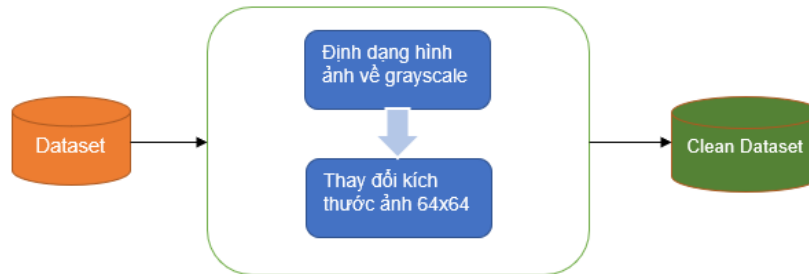


Fig. 4: Các bước chuẩn bị dữ liệu.

• **Convert color image to Grayscale** : Chúng tôi cảm thấy rằng thông tin bổ sung được cung cấp bởi hình ảnh RGB sẽ không cần thiết vì màu sắc không có ý nghĩa trong việc phát hiện khẩu trang. Một lý do khác để chọn Grayscale là xử lý ảnh grayscale nhanh hơn vài lần so với xử lý ảnh RGB [4].

• **Resize image 64x64**: Đối với kích thước của hình ảnh, chúng tôi quyết định thay đổi thành 64x64 vì chúng tôi đã thử nghiệm nhiều kích thước ảnh khác nhau và kích thước 64x64 cho kết quả xử lý nhanh và tốt nhất. Hình 5 dưới đây là một ví dụ về một trong những hình ảnh được xử lý trước.

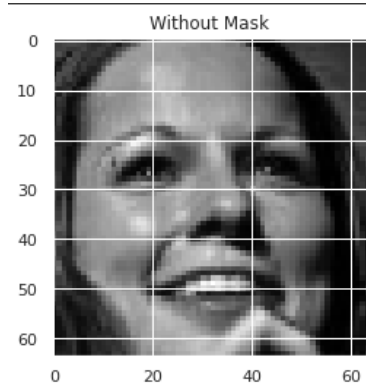


Fig. 5: Hình ảnh đã được xử lý trước.

4.3 Convolutional Neural Network

Mô hình mạng neural tích chập (CNN) rất phổ biến trong các tác vụ thị giác máy tính [8] [5] [11] như xác định xem người trong hình có mang khẩu trang hay không. Nói ngắn gọn về mạng CNN: nó là một mạng lưới thần kinh với các lớp bổ sung ở phía trước để trích xuất các thuộc tính nhất định trong hình ảnh. Các lớp bổ sung đó là các lớp tích chập và phân biệt các thuộc tính độc đáo mà một hình ảnh cần phải có. Hình 6 cho thấy tổng quan về mạng CNN.

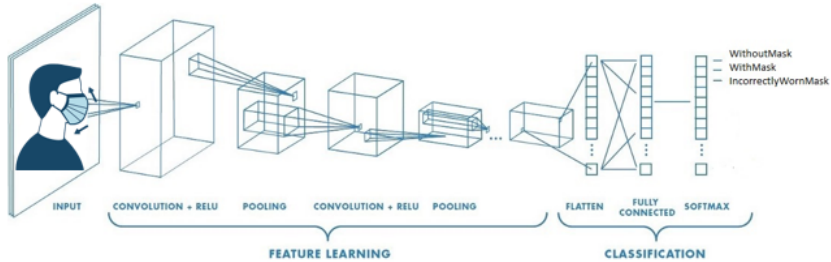


Fig. 6: Mạng Neural Tích chập (CNN)

Chúng tôi thực hiện phương pháp này vì mô hình CNN đạt hiệu suất cao trong tác vụ phân loại ảnh từ những bài báo [8] [10] [5]. Mạng CNN nhận nhiệm vụ trích xuất các thuộc tính từ những hình ảnh đầu vào có mang khẩu trang hoặc không mang khẩu trang. Hình 7 trình bày cấu trúc phương pháp tiếp cận của chúng tôi về phát hiện người đeo khẩu trang.

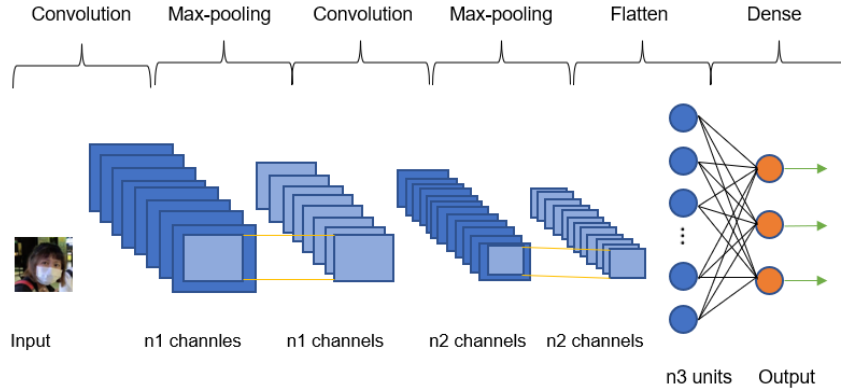


Fig. 7: Tổng quan về hệ thống phát hiện người đeo khẩu trang, không mang khẩu trang, mang khẩu trang sai cách của chúng tôi.

5 Experiments and Results

5.1 Evaluation Metric

Trước khi đi qua các kết quả thử nghiệm, trước tiên chúng tôi sẽ đề cập tới các thước đo đánh giá được sử dụng trong bài báo này. Trong bài báo cáo [5] sử dụng F1 – macro để đánh giá các mô hình của mình. Do đó, chúng tôi cũng quyết định sử dụng F1-macro (%) để đo hiệu suất của các mô hình.

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (1)$$

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (2)$$

$$Recall = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (3)$$

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

5.2 Baseline System

5.2.1 Multinomial Naive Bayes: Multinomial Naive Bayes, một phân loại của Naïve Bayes classifier, là một thuật toán dựa trên lý thuyết xác suất thống kê Bayes' Theorem để đưa ra dự đoán và phân loại dữ liệu dựa trên dữ liệu quan sát và thống kê.

Multinomial Naive Bayes (MultinomialNB) là một trong nhiều thuật toán được sử dụng trong học máy để đưa ra các dự đoán chính xác nhất trên một tập dữ liệu được thu thập vì nó tương đối dễ đào tạo và đạt được hiệu suất cao [3]. MultinomialNB sử dụng phương pháp đếm tần suất để tính khả năng. Thích hợp với các tập dữ liệu có dữ liệu dạng phân loại. Nó thuộc về nhóm học máy có giám sát.

5.2.2 Gaussian Naive Bayes: Cũng là một phân loại của Naïve Bayes classifier, Gaussian Naive Bayes được sử dụng chủ yếu trong loại dữ liệu mà các thành phần là các biến liên tục. Với mỗi chiều dữ liệu và một lớp tuân theo một phân phối chuẩn có kỳ vọng và phương sai.

Trong Gaussian Naive Bayes, các giá trị liên tục được liên kết với mỗi đối tượng được giả định là phân phối theo phân phối Gauss(phân phối Chuẩn). Khi được vẽ, nó cho một đường cong hình chuông đối xứng về giá trị trung bình của các giá trị đối tượng.

5.2.3 K-Nearest Neighbors: K-Nearest Neighbor là một trong những thuật toán supervised-learning đơn giản nhất (rất hiệu quả trong một vài trường hợp) trong Machine Learning. KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong training set gần nó nhất, không quan tâm đến việc có một vài điểm dữ liệu trong những điểm gần nhất này là nhiễu [6] [5]

Khi training thuật toán này không học một điều gì từ dữ liệu training nên thuật toán này được xếp vào loại lazy learning.

5.2.4 Decision Trees: Cây quyết định là công cụ mạnh mẽ và phổ biến nhất để phân loại và dự đoán. Cây quyết định là một lưu đồ giống như cấu trúc cây, trong đó mỗi nút bên trong biểu thị một thử nghiệm trên một thuộc tính, mỗi nhánh biểu thị một kết quả của thử nghiệm và mỗi nút lá (nút đầu cuối) lưu trữ một nhãn lớp [5].

Mục tiêu của việc sử dụng Cây quyết định là tạo ra một mô hình đào tạo có thể sử dụng để dự đoán lớp hoặc giá trị của biến mục tiêu bằng cách học các quy tắc quyết định đơn giản được suy ra từ dữ liệu sơ bộ (dữ liệu đào tạo).

Trong Cây quyết định, chúng tôi bắt đầu từ gốc của cây để dự đoán nhãn lớp cho một bản ghi. Chúng tôi so sánh các giá trị của thuộc tính gốc với thuộc tính của bản ghi. Dựa trên sự so sánh, chúng tôi theo dõi nhánh tương ứng với giá trị đó và nhảy đến nút tiếp theo.

5.2.5 Support Vector Machine: Support Vector Machines (SVM) là kỹ thuật phổ biến đối với việc phân lớp dữ liệu, là phương pháp sử dụng không gian giả thuyết các hàm tuyến tính trên không gian đặc trưng nhiều chiều, dựa trên lý thuyết tối ưu và lý thuyết thống kê. Trong kỹ thuật SVM không gian dữ liệu nhập ban đầu sẽ được ánh xạ vào không gian đặc trưng và trong không gian đặc trưng này mặt siêu phẳng phân chia tối ưu sẽ được xác định [5].

SVM thực chất là một bài toán tối ưu, mục tiêu của thuật toán này là tìm được một không gian F và siêu phẳng quyết định f trên F sao cho sai số phân loại là thấp nhất.

5.2.6 Convolutional Neural Network: Không giống như bài báo nghiên cứu [5] sử dụng pretrained model ResNet để trích xuất các tính năng và sử dụng để đào tạo các mô hình của họ, chúng tôi xây dựng CNN từ đầu và trích xuất các tính năng riêng quan trọng trong bài toán phân loại của chúng tôi. Để xây dựng mô hình, chúng tôi sẽ sử dụng Keras của Tensorflow và GPU của Google Colab vì nó giúp tăng tốc quá trình đào tạo. Khi ở trên Google Colab, hãy đảm bảo đặt thời gian chạy GPU.

5.3 Experimental Settings

Như đã mô tả trong phần 4, chúng tôi thử nghiệm trên các tập dữ liệu đã cho hai phương pháp: Học máy và mạng Neural tích chập.

5.3.1 Machine Learning approach: Bài báo cáo này sử dụng các mô hình Machine Learning: KNN, Naive Bayes, SVM, Decision Trees.

- **Naive Bayes:** Sau khi chuẩn hóa tập dữ liệu bằng cách chia tất cả các pixel cho 255, chúng tôi chọn ngẫu nhiên ảnh để kiểm tra. Sau đó, tạo mô hình GaussianNB mà không điều chỉnh siêu tham số. Tiếp theo, chúng tôi điều chỉnh dữ liệu đào tạo cùng với các nhãn tương ứng của nó và kiểm tra dữ liệu thử nghiệm của chúng tôi. Lặp lại các bước cho MultinomialNB. Tiếp theo, chúng tôi chuyển sang điều chỉnh siêu tham số cho cả hai mô hình. Đối với GaussianNB, siêu tham số là "**var_smoothing**" và Đa thức là "**alpha**".

- Tạo GridSearchCV cho Gaussian Naive Bayes với các tham số được đặt là: "**var_smoothing**": [1e-09, 1e-07, 1e-05, 1e-03, 1, 100]".

- Các tham số của MultinomialNB thành "**alpha**": [0,1, 0,5 , 1, 2, 5, 100]. Đặt các mô hình với các thông số tối ưu và chạy lại bài kiểm tra đánh giá trên chúng.

- **K-Nearest Neighbors:** Mô hình được chuẩn hóa để mọi pixel chứa giá trị từ 0-1 rồi mới tiến hành đào tạo. Lần lặp đầu tiên của mô hình K-NN được phát triển sử dụng các tham số mặc định cho một K-Neighbors Classifier do scikit-learning cung cấp. Các tham số này là **n_neighbors = 5, weights = "Uniform", p = "2", metric = "minkowski"**. Theo mặc định, bộ phân loại K-NN của scikit-learning sẽ tự động quyết định thuật toán K-NN nào là thích

hợp nhất để sử dụng để tính toán các hàng xóm gần nhất (ví dụ: cây, bóng, cây kd, brute force K-NN) (Sklearn.neighbors.KNeighborsClassifier, 2020). Các bộ đào tạo và kiểm tra giống nhau được sử dụng để đánh giá tất cả các mô hình và bản chất phân tách giữa các bộ kiểm tra và đào tạo được thực hiện với trạng thái ngẫu nhiên là 45 và kích thước kiểm tra là 20%.

Do tính chất lười học của bộ phân loại K-NN, một nhược điểm lớn của việc sử dụng nó là do các lệnh gọi đào tạo và phân loại tốn kém về mặt tính toán và tốn kém về bộ nhớ của phiên bản mới. Do đó, lần lặp tiếp theo của mô hình tập trung vào việc giảm kích thước bằng phương pháp Principal Component Analysis (PCA) [2] để chuyển tập dữ liệu từ không gian chiều cao gồm 4.096 đối tượng thành không gian chiều thấp chỉ có 153 đối tượng. Việc phân tách mô hình PCA đã được thiết lập để bảo toàn 90% phương sai trong tập dữ liệu và ngăn mô hình không được trang bị đầy đủ.

Lần lặp thứ ba, GridSearchCV đã được sử dụng để kiểm tra sự kết hợp của các giá trị tham số được xác định trước cho **n_neighbors**, chỉ số khoảng cách và p, tham số công suất cho chỉ số khoảng cách **minkowski**. GridSearchCV cũng rất tốn kém về mặt tính toán (mất hơn 15 giờ để tính toán tất cả các tổ hợp được chọn cho các tham số). Ở lần lặp lại cải tiến mô hình này, GridSearchCV không được sử dụng cùng với phép chuyển đổi PCA.

Lần lặp lại thứ tư của cải tiến mô hình đã sử dụng RandomizedSearchCV thay vì GridSearchCV vì nó được mong đợi sử dụng ít bộ nhớ hơn và hoạt động nhanh hơn với cơ hội cao hơn để tìm thấy sự kết hợp tối ưu của các tham số cho mô hình so với GridSearchCV. RandomizedSearchCV mất khoảng 3 giờ để tính toán và phát hiện ra giải pháp tham số tối ưu với độ chính xác 86,7%. Các thông số tối ưu là **n_neighbors = 2, weight = distance, và distance = manhattan**.

- **Support Vector Machine:** Các mô hình SVM rất phù hợp để phân loại hình ảnh và cho các bộ dữ liệu có kích thước vừa và nhỏ, đó là lý do tại sao chúng tôi xây dựng SVM để phân loại bộ dữ liệu của mình. Vì thời gian tính toán là một mối quan tâm, chúng tôi đã tận dụng lợi thế là SVM rất tốt để phân loại các tập dữ liệu có kích thước vừa và nhỏ. Thay vì sử dụng tất cả 17,836 ảnh, chúng tôi đã chia nhỏ bộ dữ liệu để đào tạo theo tỉ lệ phần trăm 100-75-50-25.

- **Decision Trees:** Trước khi mô hình được đào tạo, các pixel được chuẩn hóa thành phạm vi 0-1 từ phạm vi 0-255 và phân tách với kích thước thử nghiệm là 20% và trạng thái ngẫu nhiên là 45. Lần lặp đầu tiên của mô hình cây quyết định được phát triển đã sử dụng các tham số mặc định cho một DecisionTreeClassifier() do scikit-learning cung cấp. Tuy nhiên, lần lặp thứ hai của mô hình cây quyết định khám phá việc giảm kích thước với hy vọng loại bỏ các tính năng không quan trọng có thể ảnh hưởng tiêu cực đến Phân tích thành phần chính đã được áp dụng với việc bảo toàn 90% phương sai trong tập huấn luyện.

5.3.2 Our approach (CNN): Trước tiên, áp dụng phân tích dữ liệu, chuẩn hóa tất cả các pixel trong tập dữ liệu thành phạm vi $[0,1]$ để mô hình có thời gian hội tụ nhanh hơn. Một vấn đề xảy ra nếu có một tập dữ liệu lớn, nếu chia cho 255 sẽ hạn chế mức sử dụng RAM vì các dấu chấm động và đảm bảo hạn chế được dung lượng mà nó chiếm. Một điều quan trọng khác cần lưu ý về Keras là đầu vào (hình ảnh) phải được tạo theo kích thước (rộng x dài x kích thước); trong đó kích thước là số channel, (bằng 1 nếu hình ảnh có thang độ xám và 3 nếu hình ảnh có màu hoặc RGB.)

Vì chúng tôi đào tạo về thang độ xám 64×64 cho nên đã thay đổi kích thước dữ liệu thành 64×64 bằng cách sử dụng Cv2 hoặc Pillow. Sau đó sử dụng numpy để định hình lại kích thước.

- `X = X.reshape (-1, 64, 64, 1)`.

Chúng ta sẽ sử dụng 2 lớp chập và 2 lớp Max-pooling. Đầu tiên, tạo mô hình tuần tự và thêm các lớp vào đó. CNN có các lớp tích chập và lớp Max-pooling ở trước để trích xuất các tính năng, vì thế chúng tôi sẽ thêm tổng cộng 6 lớp: 3 lớp tích chập và 3 lớp Max-pooling.

- Max-pooling
- Conv2d parmas: `filters=[8, 16, 32]; kernel_size=3; activation='relu'`
- Làm phẳng các filter thành một chiều bằng cách sử dụng Flatten.
- Dense: `[512, 256, 128]; activation='relu'`
- Dropout layer: 0.3
- Dense: 3; `activation='softmax'`

Lý do cho số lớp Filter ít hơn lớp Conv và lớp Dropout là để mô hình ít bị quá tải (overfitting). Do đó, mô hình hoạt động tốt hơn cho nhiệm vụ phân loại. Chúng tôi không muốn mô hình trang bị quá nhiều dữ liệu bởi vì sau đó chúng tôi sẽ sử dụng nó để đưa ra dự đoán về những hình ảnh khác mà mô hình chưa từng thấy trước đây.

Cuối cùng, chúng tôi biên dịch mô hình bằng cách sử dụng trình biên dịch, và thực hiện phương pháp training/fitting tương tự như các mô hình khác.

5.4 Experimental Result

Sau khi chạy thử nghiệm, tìm ra các siêu tham số và điều chỉnh siêu tham số của các model. Chúng tôi đã thống kê lại các kết quả trên bảng 4 và cũng rút ra được nhiều nhận xét về các model học máy và mạng CNN.

- **Mô hình Naive Bayes** không phù hợp trong trường hợp này và làm ảnh hưởng đến việc phân loại. Chúng tôi quyết định không sử dụng thuật toán này để kiểm tra bất kỳ sự sai lệch nào trong mô hình hoặc dữ liệu của chúng tôi vì hiệu suất của nó không như mong muốn. Cũng từ kết quả của chúng tôi, chúng tôi có thể hiểu tại sao các nhà nghiên cứu cơ sở lại chọn loại bỏ thuật toán này trong các nghiên cứu của họ.

Table 4: Kết quả đánh giá của bộ dữ liệu Face-Mask phát hiện người đeo khẩu trang, không đeo khẩu trang và đeo khẩu trang sai cách.

Models	Face-Mask
	F1-score
MultinomialNB (default Hyperparameter)	67.97
GaussianNB (default Hyperparameter)	78.14
MultinomialNB with Hyperparameter tuning	68.02
GaussianNB with Hyperparameter tuning	78.14
KNN (default Hyperparameter)	76.68
KNN using PCA (keep 90% of variance)	85.20
KNN using PCA with (best-params with Hyperparameter tuning)	82.76
SVM using 100% of dataset	96.28
SVM using 75% of dataset	95.16
SVM using 50% of dataset	95.02
SVM using 25% of dataset	93.81
Decision Trees	85.01
Decision Trees with Hyperparameter tuning	85.37
Decision Trees using PCA (keep 90% of variance)	77.30
Decision Trees using PCA (best-params)	78.00
Our approach (CNN)	98.68

• **Đối với mô hình KNN:** Khi không áp dụng điều chỉnh siêu tham số, mô hình đạt f1-score khoảng 76.68%. Việc áp dụng PCA[2] cho mô hình và sử dụng RandomizedSearchCV để tìm kiếm ngẫu nhiên các siêu tham số đã được thử trong lần lặp lại thứ 3 của quá trình. Tuy nhiên, mô hình chỉ đạt được độ chính xác 82.76% khi sử dụng các siêu tham số tốt nhất cho mô hình K-NN được thiết lập.

• **Các mô hình SVM** rất tốt để phân loại hình ảnh và rất tốt cho các bộ dữ liệu có kích thước vừa và nhỏ [5], đó là lý do tại sao chúng tôi xây dựng mô hình SVM. Vì thời gian tính toán là một mối quan tâm, chúng tôi đã tận dụng thực tế là SVM rất tốt để phân loại các tập dữ liệu có kích thước vừa và nhỏ. Thay vì sử dụng tất cả 17.836 ảnh, chúng tôi đã sử dụng 50% bộ dataset Face-Mask để đào tạo mô hình SVM của mình.

• **Decision Trees:** Khi không có bất kỳ điều chỉnh siêu tham số nào, mô hình cây quyết định có accuracy là 84,6% và độ chính xác tổng thể tốt hơn cho từng nhãn phân loại so với mô hình K-NN. Hơn nữa, bộ phân loại cây quyết định có thời gian tính toán nhanh hơn khoảng 3-7 phút để điều chỉnh và dự đoán một mô hình duy nhất. Tuy nhiên, điều thú vị là với PCA được áp dụng [2], mô hình thực hiện với độ chính xác 78,00%, giảm 7% so với mô hình cây quyết định mà không áp dụng bất kỳ sự giảm kích thước nào. Việc thay đổi bảo toàn phương sai PCA từ 90% thành 99% và 80% cũng không cải thiện độ chính xác của mô hình. Vì việc giảm các tính năng thông qua PCA dường như có ảnh hưởng không

tốt, vì vậy chúng tôi tập trung nhiều hơn trong các lần lặp tiếp theo vào việc tìm kiếm các thông số tối ưu cho mô hình.

- **Our approach (CNN):** Khi đánh giá mô hình, chúng tôi thấy mô hình CNN hoạt động rất tốt trên tập dữ liệu của chúng tôi. Mô hình đạt độ chính xác khoảng 99% trên dữ liệu thử nghiệm của mình. Như vậy, trong tất cả các mô hình chúng tôi đã xây dựng thì mô hình CNN đạt hiệu suất cao nhất (98.68%) sau đó là tối mô hình SVM (96.28%).

5.5 Error Analysis

Phân tích lỗi được thực hiện để phân tích các lỗi mà chúng tôi gặp phải trong hệ thống của mình bằng phân tích định lượng sử dụng ma trận nhầm lẫn của mô hình hoạt động tốt nhất. Hình 8 cho thấy ma trận nhầm lẫn của mô hình tốt nhất của chúng tôi khi phân loại đeo khẩu trang, không đeo khẩu trang và đeo khẩu trang sai cách trong bộ thử nghiệm. Chúng tôi thấy rằng khả năng hệ thống của chúng tôi phân loại nhãn IncorrectlyWornMask tốt hơn nhãn WithoutMask và nhãn WithMask.

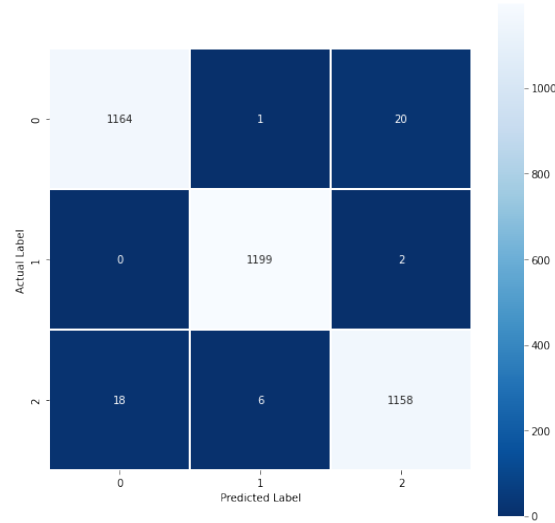


Fig. 8: Ma trận nhầm lẫn của model CNN.

Chúng tôi đã xây dựng bộ dữ liệu Test-bias để đánh giá độ sai lệch của mô hình. Xử lý dữ liệu đầu vào giống với bộ dữ liệu 17,836 ảnh. Sau khi hoàn thành các bước trên, chúng tôi dùng các mẫu thử nghiệm mà mô hình chưa từng thấy để đánh giá.

Với mô hình CNN đạt độ chính xác cao nhất trong tất cả các mô hình của bài báo cáo này. Từ những kết quả mới này về bộ dataset Test-bias, chúng tôi thu được kết quả rất thú vị.

Table 5: Kết quả đánh giá bias của mô hình CNN.

	precision	recall	f1-score	support
WithoutMask	0.50	0.11	0.18	152
Incorrectly Worn Mask	0.26	0.12	0.17	144
WithMask	0.50	0.95	0.65	213
accuracy			0.47	509
macro avg	0.42	0.40	0.33	509
weighted avg	0.43	0.47	0.38	509

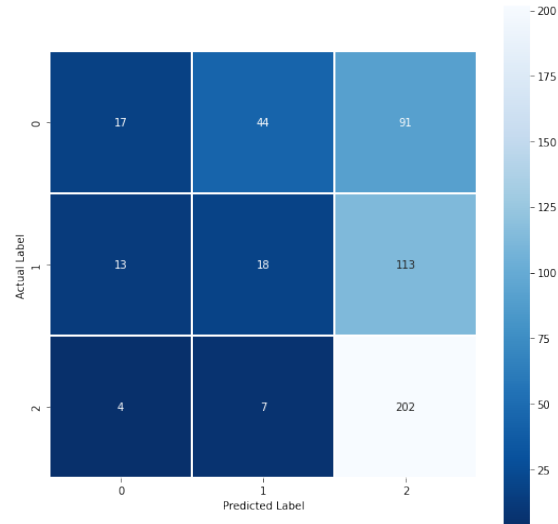


Fig. 9: Ma trận nhầm lẫn của model CNN.

Từ kết quả trên cho chúng tôi thấy, độ chính xác của mô hình chỉ đạt khoảng 47%. Điều này nói lên được mô hình CNN của chúng tôi không thực sự tốt khi kiểm tra bằng tập dữ liệu Test-bias. Bởi vì tập dữ liệu đào tạo của chúng tôi không được xử lý tốt, do có nhiều hình ảnh bị mờ và hình ảnh của nhân IncorrectlyWornMask được photoshop khẩu trang. Mô hình nhầm lẫn giữa ảnh đeo khẩu trang và đeo khẩu trang không đúng cách khá nhiều (113 ảnh / 144 ảnh).

6 Conclusion and Future works

Thông qua thử nghiệm với các mô hình phi tham số (non-parametric) cho KNN và Decision Tree. Hình ảnh có 4.096 thuộc tính và khi được áp dụng cho các bộ phân loại sử dụng nhiều bộ nhớ như KNN, mô hình này sẽ tiêu tốn rất nhiều tài nguyên máy tính để hoàn thành việc tính toán. Bằng cách giảm các thuộc tính không cần thiết, chúng tôi không chỉ có thể cải thiện thời gian chạy mà còn có thể tăng độ chính xác của mô hình bằng cách loại bỏ các thuộc tính không liên quan. Chúng tôi cũng học được tầm quan trọng của việc xử lý trước dữ liệu và lựa chọn tập dữ liệu. Có khả năng là sự khác biệt trong bộ dữ liệu đào tạo và chất lượng hình ảnh của chúng tôi, chẳng hạn như hình ảnh phóng to và khẩu trang được mô phỏng bằng photoshop đã dẫn đến sai sót trong việc đào tạo các mô hình của chúng tôi và cản trở khả năng dự đoán các trường hợp những hình ảnh mà mô hình chưa từng nhìn thấy (Test-bias).

Tóm lại, chúng tôi đã kết hợp được 2 bộ dữ liệu riêng biệt để tạo thành một bộ dữ liệu mới và có các mô hình đạt hiệu suất cao như CNN và SVM, tất cả các mô hình của chúng tôi đều hoạt động kém khi phân loại các trường hợp mới khi được đưa vào giai đoạn kiểm tra bias của mô hình. Cần kiểm tra thêm về những hạn chế của bộ dữ liệu đào tạo cũng như cấu hình của các mô hình để cải thiện độ chính xác của chúng tôi trên các nhóm người cũng như các trường hợp đào tạo mới.

Trong tương lai, chúng tôi có kế hoạch triển khai một số mô hình học tập chuyển giao (Faster R-CNN) và YOLOv5 [9] và thiết lập các thử nghiệm với nhiều thông số khác nhau cho phương pháp tiếp cận mà chúng tôi đề xuất nhằm tìm ra một mô hình tốt nhất, tìm hiểu thêm về Data Augmentation để cải thiện chất lượng của tập dữ liệu và có hiệu suất tốt hơn trong việc phát hiện người đeo khẩu trang, không đeo khẩu trang và đeo khẩu trang sai cách.

References

1. Adnane Cabani, Karim Hammoudi, Halim Benhabiles, and Mahmoud Melkemi. Maskedface-net—a dataset of correctly/incorrectly masked face images in the context of covid-19. *Smart Health*, 19:100144, 2021.
2. Md Sabbir Ejaz, Md Rabiul Islam, Md Sifatullah, and Ananya Sarker. Implementation of principal component analysis on masked and non-masked face recognition. In *2019 1st international conference on advances in science, engineering and robotics technology (ICASERT)*, pages 1–5. IEEE, 2019.
3. P Geetha Pavani, Birendra Biswal, MVS Sairam, and N Bala Subrahmanyam. A semantic contour based segmentation of lungs from chest x-rays for the classification of tuberculosis using naïve bayes classifier. *International Journal of Imaging Systems and Technology*, 2021.
4. Tarun Kumar and Karun Verma. A theory based on conversion of rgb image to gray image. *International Journal of Computer Applications*, 7(2):7–10, 2010.
5. Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N Taha, and Nour Eldeen M Khalifa. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the covid-19 pandemic. *Measurement*, 167:108288, 2021.

6. Abdellah Oumina, Nouredine El Makhfi, and Mustapha Hamdi. Control the covid-19 pandemic: Face mask detection using transfer learning. In *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, pages 1–5. IEEE, 2020.
7. Bosheng Qin and Dongxiao Li. Identifying facemask-wearing condition using image super-resolution with classification network to prevent covid-19. *Sensors*, 20(18):5236, 2020.
8. G Saranya, Dipshikha Sarkar, Sayan Ghosh, Lokenath Basu, K Kumaran, and N Ananthi. Face mask detection using cnn. In *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, pages 426–431. IEEE, 2021.
9. Sunil Singh, Umang Ahuja, Munish Kumar, Krishan Kumar, and Monika Sachdeva. Face mask detection using yolov3 and faster r-cnn models: Covid-19 environment. *Multimedia Tools and Applications*, 80(13):19753–19768, 2021.
10. K Suresh, MB Palangappa, and S Bhuvan. Face mask detection by using optimistic convolutional neural network. In *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, pages 1084–1089. IEEE, 2021.
11. Jesús Tomás, Albert Rego, Sandra Viciano-Tudela, and Jaime Lloret. Incorrect facemask-wearing detection using convolutional neural networks with transfer learning. In *Healthcare*, volume 9, page 1050. Multidisciplinary Digital Publishing Institute, 2021.