Tạp chí Khoa học Trường Đại học Cần Thơ
Phần A: Khoa học Tự nhiên, Công nghệ và Môi trường
website: sj.ctu.edu.vn

DOI:10.22144/ctu.jvn.2021.111

NHẬN DẠNG TIẾNG NÓI ĐIỀU KHIỂN VỚI CONVOLUTIONAL NEURAL NETWORK (CNN)

Thái Thuận Thương*
Khoa Công nghệ Thông tin, Trường Đại học Yersin Đà Lạt
**Người chịu trách nhiệm về bài viết: Thái Thuận Thương (email: thuongtt@yersin.edu.vn)*

Thông tin chung:
Ngày nhận bài: 04/01/2021
Ngày nhận bài sửa: 08/06/2021
Ngày duyệt đăng: 20/08/2021

Title:
Voice recognition control with convolutional neural network (CNN)

Từ khóa:
Convolutional neural network (CNN), deep neural network (DNN), keyword spooting (KWS)

Keywords:
Convolutional neural network (CNN), deep neural network (DNN), keyword spooting (KWS)

ABSTRACT

Voice control is an important function in many mobile devices and smart home systems, especially it is also a solution to help disabled people controlling common devices in their life. This paper indicates a short-controlled speech recognition method using MFCC (Mel frequency cepstral coefficients) and convolutional neural network (CNN) models. The input audio data are wave files that are assumed to be exactly 1 second in duration. A sliding window of size 30 ms with a step of 10 ms slides in turn over the input data to calculate the MFCC parameters. Each input file will obtain 98 MFCC features, each MFCC feature is a 40-dimensional vector (corresponding to 40 coefficients of Mel-scales filters). The research has used 3 Neural Network models to classify these control speech files: 1-layer Vanilla Neural Network model (1 softmax layer), Deep Neural Network - DNN (with 3 fully connected hidden layers) enough and 1 output layer) and the Convolution Neural Network model - CNN. Experiments were performed on Google's "Speech Commands Dataset" dataset. (https://ai.googleblog.com/2017/08/launching-speech-commands-dataset.html) consisting of 65,000 samples divided into 30 classes. Experimental results show that the CNN model achieves the highest classification accuracy of 94.5%.

TÓM TẮT

Điều khiển bằng giọng nói là một chức năng quan trọng trong nhiều thiết bị di động, hệ thống nhà thông minh, đặc biệt đó là một giải pháp giúp cho người khuyết tật có thể điều khiển được các thiết bị thông dụng trong cuộc sống. Bài báo trình bày một phương pháp nhận dạng tiếng nói điều khiển ngắn sử dụng đặc trưng MFCC (Mel frequency cepstral coefficients) và mô hình convolutional neural network (CNN). Dữ liệu âm thanh đầu vào là các file wave được giả định có thời lượng đúng 1 giây. Một cửa sổ trượt kích thước 30 ms với bước dịch chuyển 10 ms lần lượt trượt trên dữ liệu đầu vào để tính các thông số MFCC. Với mỗi tập tin đầu vào sẽ thu được 98 đặc trưng MFCC, mỗi đặc trưng MFCC là một vector 40 chiều (tương ứng 40 hệ số của các bộ lọc Mel-scales). Nghiên cứu đã để xuất sử dụng 3 mô hình Neural Network để phân lớp các tập tin tiếng nói điều khiển này: Mô hình Vanilla Neural Network 1 layer (1 softmax layer), Deep Neural Network - DNN (với 3 layers ẩn kết nối đầy đủ và 1 lớp output) và mô hình Convolution Neural Network - CNN. Các thực nghiệm được thực hiện trên tập dữ liệu "Speech Commands Dataset" của Google (https://ai.googleblog.com/2017/08/launching-speech-commands-dataset.html) gồm 65.000 mẫu được chia thành 30 lớp. Kết quả thực nghiệm cho thấy mô hình CNN đạt được độ chính xác phân lớp là cao nhất: 94,5%.

1. GIỚI THIỆU

Ngày nay với sự phát triển nhanh chóng của các thiết bị di động, việc tương tác với các thiết bị bằng công nghệ giọng nói ngày càng trở nên phổ biến. Các sản phẩm liên quan như Google Now hay iPhone's Siri đều khai thác công nghệ ra lệnh bằng giọng nói. Google cũng đã cung cấp dịch vụ tìm kiếm bằng giọng nói (Schalkwyk et al., 2010) trên điện thoại Android và điều khiển hoàn toàn bằng tiếng nói có tên là "Ok Google" (Chen et al., 2014). Trên thực tế, công nghệ phát hiện từ khóa (KWS) là một kỹ thuật tiềm năng để cung cấp giao diện hoàn toàn rảnh tay và điều này đặc biệt thuận tiện cho các thiết bị di động so với việc gõ bằng tay. Và nó cũng là kỹ thuật mong muốn trong các tình huống như lái xe hoặc một số trường hợp khẩn cấp. Vì hệ thống nhận dạng tiếng nói điều khiển thường chạy trên điện thoại thông minh hoặc máy tính bảng, do đó nó phải có độ trễ thấp, bộ nhớ rất ít và yêu cầu tính toán nhỏ. Do đó, mục tiêu của nghiên cứu là xây dựng một hệ thống nhận dạng tiếng nói với những từ khóa được xác định trước nhằm giúp tương tác với các thiết bị dựa trên những lệnh yêu cầu. Tập dữ liệu được sử dụng trong các thử nghiệm của bài báo này là do nhóm TensorFlow và AIY của Google cung cấp, chứa 65.000 tập tin âm thanh dạng WAVE của ba mươi từ khác nhau (Warden, 2017). Mỗi đoạn âm thanh kéo dài một giây và chứa một từ duy nhất, theo các yêu cầu khác nhau chẳng hạn như *"yes" / "no" hoặc "up" / "down" / "left" / "right" hoặc "stop" / "go"* mỗi nhãn chứa 2377 tập tin âm thanh. Các từ khóa này có thể được cấu hình lại, do đó cho phép hệ thống hoạt động linh hoạt với các nhãn khác nhau. Hệ thống cố gắng phân loại các đoạn âm thanh với thời lượng một giây là "*silence*"*, " unknown"* hoặc một trong các từ khóa được xác định trước. Sau đó, sử dụng mô hình Vanilla với một lớp *softmax*, mô hình DNN 3 lớp ẩn và mô hình CNN để tính xác suất của từng nhãn âm thanh đầu vào và cuối cùng xuất ra nhãn dự đoán.

2. NGHIÊN CỨU LIÊN QUAN

Máy học đã được chứng minh là có khả năng mạnh mẽ cho bài toán phân loại. Một kỹ thuật thường được sử dụng cho nhiệm vụ phát hiện từ khóa (KWS) là Hide Markov Model Key-Word/Filler (HMM) (Rohlicek et al., 1989; Rose & Paul, 1990; Silaghi & Bourlard, 1999; Silaghi, 2005). Trong cách tiếp cận tổng quát này, mỗi từ khóa sẽ được huấn luyện với mô hình HMM chính, và các phân đoạn không phải từ khóa của tín hiệu tiếng nói sẽ được huấn luyện với mô hình HMM phụ (Chen et al., 2014). Các nghiên cứu gần đây đã đưa ra một số mô hình khác dựa trên công thức biên độ lớn (Keshet & Bengio, 2009; Tabibianet al., 2011) hoặc mô hình Recurrent Neural Network (RNN) (Li et al., 1992; Fernández et al., 2007). Các kỹ thuật này cho thấy một số cải tiến so với cách tiếp cận HMM, nhưng có độ trễ tương đối dài, vì chúng yêu cầu xử lý toàn bộ tiếng nói để tìm vùng của từ khóa hoặc lấy đầu vào từ một khoảng thời gian dài để dự đoán từ khóa. Hệ thống KWS hiện tại của Google (Chen et al., 2014) sử dụng DNN, hệ thống này tốt hơn hệ thống HMM truyền thống và cũng rất đơn giản, yêu cầu tính toán tương đối thấp hơn. Tuy nhiên, mô hình CNN có thể cung cấp một số cải tiến hơn so với mô hình DNN trong các bài toán về từ vựng lớn và nhỏ (Abdel-Hamid et al., 2012; Sainath et al., 2013). CNN tốt hơn DNN cho bài toán KeyWords (KWS) chủ yếu vì 2 lý do. Đầu tiên, DNN chỉ cần bỏ qua cấu trúc liên kết đầu vào và thay đổi kích thước của nó thành các vector cột. Tuy nhiên, đối với tín hiệu âm thanh thì các biểu diễn phổ cho thấy mối liên quan rất chặt chẽ về thời gian và tần số. Vì vậy, với CNN việc mô hình hóa các mối liên kết cục bộ sẽ có lợi và có hiệu suất tốt hơn nhiều so với DNN. Thứ hai là về chất lượng chia sẻ tham số, trên cùng một tác vụ thì CNN có thể có ít tham số hơn nhiều so với DNN, do đó làm giảm dung lượng bộ nhớ và yêu cầu tính toán. Vì vậy, CNN sẽ có hiệu suất được cải thiện và giảm kích thước mô hình so với DNN trong bài toán KWS.

3. NỘI DUNG

3.1. Tập dữ liệu và đặc trưng

3.1.1. Tập dữ liệu

Trong bài báo sử dụng bộ dữ liệu tiếng nói điều khiển của Google, bao gồm 65.000 tệp âm thanh tiếng nói với 35 từ khác nhau, mỗi tệp kéo dài trong một giây. Tập dữ liệu đã được chia thành các danh mục khác nhau như số, động vật, chỉ đường hoặc tên người. Bằng cách đó, hệ thống có thể được huấn luyện với các mục đích cụ thể hơn. Tập dữ liệu này được chia thành ba phần, bao gồm 80% tập huấn luyện, 10% tập đánh giá và 10% thử nghiệm, và mỗi tập hợp con của âm thanh tiếng nói được phân loại là khoảng lặng, từ không xác định hoặc từ khóa xác định trước được gắn các nhãn khác nhau tương ứng.

Từ khóa: Lớp từ khóa được gắn nhãn riêng, chứa một tập hợp các từ có liên quan. Trong bài báo này các từ như *"up", "down", "left", "right"* được chọn để thực hiện nhận dạng tiếng nói.

Từ không xác định: đó là những đoạn âm thanh ghi lại các từ phát âm không rõ có thể thúc đẩy mạng

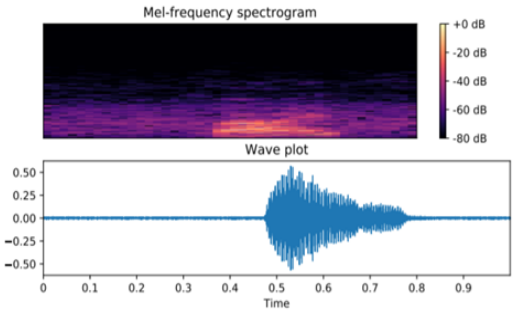
học những tiếng nói mà có thể được bỏ qua hoặc được ghi lại.

Khoảng lặng: Lớp này chứa âm thanh trên môi trường yên tĩnh.

Ngoài ra, để đa dạng hóa bộ thông số, tiếng ồn được trộn theo tỷ lệ vào tập dữ liệu huấn luyện. Cụ thể, âm thanh tiếng ồn từ máy móc và các âm thanh hoạt động sinh hoạt thường ngày được chia thành các phân đoạn nhỏ và trộn ngẫu nhiên vào các mẫu âm thanh huấn luyện với âm lượng thấp hơn âm lượng nền là 0,1 và tần số là 0,8

3.1.2. Rút trích đặc trưng MFCC

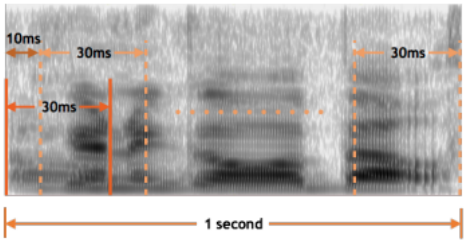
Trong bất kỳ hệ thống nhận dạng tiếng nói tự động trích các đặc trưng tốt là bước quan trọng để xác định nội dung của ngôn ngữ và loại bỏ thông tin không liên quan như tiếng ồn xung quanh. Mel-Frequency Cepstral Coefficients (MFCC) được giới thiệu bởi Davis vào những năm 1980, được sử dụng rộng rãi trong nhận dạng tiếng nói và đã trở thành công nghệ tiên tiến kể từ đó (Fernández et al., 2007). Mel-frequency cepstrum (MFC) đại diện cho năng lượng phổ ngắn hạn của âm thanh, dựa trên các biến đổi tuyến tính cosin của phổ công suất log trên thang tần số mel phi tuyến. Bước đầu tiên trong nghiên cứu này là chuyển đổi tập tin âm thanh sang các đặc trưng MFCC bằng cách sử dụng thư viện API librosa (Abdel-Hamid et al., 2012), và sau đó sử dụng các đặc trưng hai chiều này để huấn luyện mô hình. Hình 1 thể hiện dạng phổ tần số Mel và dạng sóng của một tập tin âm thanh.



Hình 1. Dạng phổ và dạng sóng của tập tin âm thanh

Hình 2. Sơ đồ rút trích đặc trưng MFCC

Sơ đồ rút trích đặc trưng được thể hiện trong Hình 2. Trước hết một cửa sổ phân tích 30 ms được xác định và chia tín hiệu tiếng nói thành các frames thời gian khác nhau bằng các cửa sổ trượt với bước dịch chuyển 10 ms. Vì mỗi mẫu tín hiệu âm thanh là 1 giây, do đó sẽ có (1000 - 30)/10 + 1 = 98 frames thời gian, như thể hiện trong Hình 3. Sau khi trượt cửa sổ, Fast Fourier Transformation (FFT) được tính toán cho mỗi frame để có được các đặc trưng tần số và ngân hàng bộ lọc Mel-Scales được áp dụng để chuyển đổi các frames Fourier. Bước cuối cùng là tính toán Discrete Cosine Transformation (DTC) để được vectơ hệ số 40 chiều. Kết quả của quá trình này sẽ thu được một ma trận 2D [98 x 40] làm đầu vào cho mạng neural.

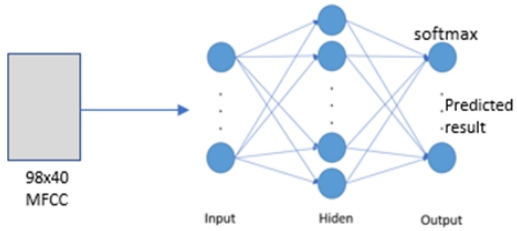


Hình 3. Đặc trưng cửa sổ Excation

3.2. Phương pháp

3.2.1. Mô hình Vanilla với một lớp Softmax

Đầu tiên, một mô hình Vanilla được xây dựng (Hình 4) với duy nhất một lớp ẩn được kết nối đầy đủ và một lớp *softmax* đầu ra, đầu vào của mô hình này là một ma trận đặc trưng MFCC với kích thước 98x40. Mô hình đơn giản này chỉ có một phép nhân ma trận và bias, số lượng các nút đầu ra cũng giống như các nhãn, tổng số lượng tham số và kiến trúc các lớp của mô hình này được thể hiện trong Hình 5. Mô hình Vanilla không thể đưa ra kết quả chính xác nhưng có thể hoạt động rất nhanh.



Hình 4. Mô hình Vanilla kết nối đầy đủ

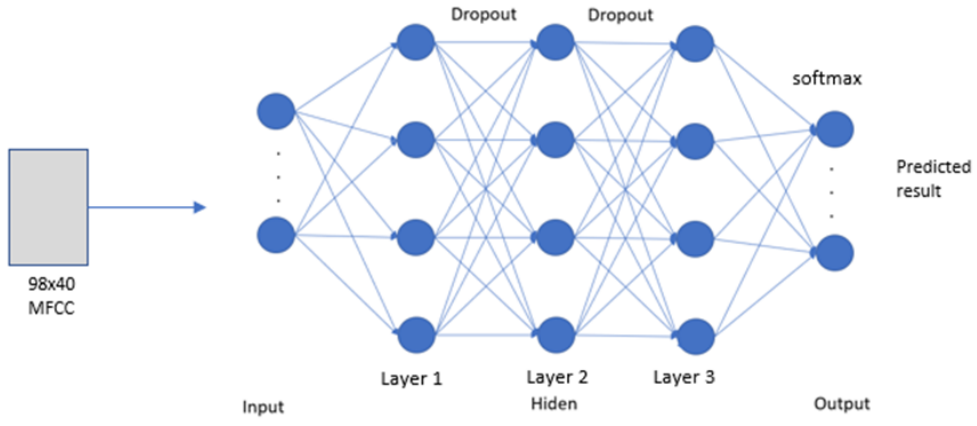
Layer (type)	Output Shape	Param #
intput (InputLayer)	[(None, 98, 40, 1)]	0
flatten (Flatten)	(None, 3920)	0
dense1 (Dense)	(None, 512)	2007552
pred (Dense)	(None, 4)	2052

Total params: 2,009,604
Trainable params: 2,009,604
Non-trainable params: 0

Hình 5. Kiến trúc mạng Vanilla

3.2.2. Mô hình Deep Neural Network - DNN

Mô hình thứ hai được xây dựng là mạng DNN 3 lớp ẩn với 128 nút ẩn trên mỗi lớp và một lớp *softmax* đầu ra được thể hiện như trong Hình 6, mỗi lớp ẩn sử dụng một đơn vị hiệu chỉnh tuyến tính (ReLU). Lớp *softmax* chứa nhãn đầu ra cho mỗi từ khóa được nhận dạng. Mô hình đã sử dụng 3 lớp ẩn vì trong thực tế mạng neural 3 lớp ẩn kết nối đầy đủ thường hoạt động tốt hơn DNN có 1 hoặc 2 lớp ẩn, nhưng chỉ kém hơn một chút so với các DNN có 4 lớp ẩn trở lên. Một kinh nghiệm thực nghiệm khác là sử dụng nhiều nút ẩn hơn trên mỗi lớp sẽ đạt được độ chính xác cao hơn, mặc dù nó có thể dẫn đến tình trạng *overfitting*. Trong bài báo này kỹ thuật *dropout* được sử dụng để ngăn chặn tình trạng *overfitting*. Đối với các lớp ẩn, đơn vị tuyến tính hiệu chỉnh (ReLU) được sử dụng như hàm kích hoạt để giảm việc tính toán tổng trọng số đầu ra từ lớp trước đó. Kiến trúc lớp và tổng tham số của mô hình DNN được thể hiện như trong Hình 7. So với mô hình Vanilla một lớp ẩn, mô hình DNN này sẽ cho kết quả chính xác hơn với chi phí bộ nhớ nhiều hơn và chi phí tính toán cao hơn.



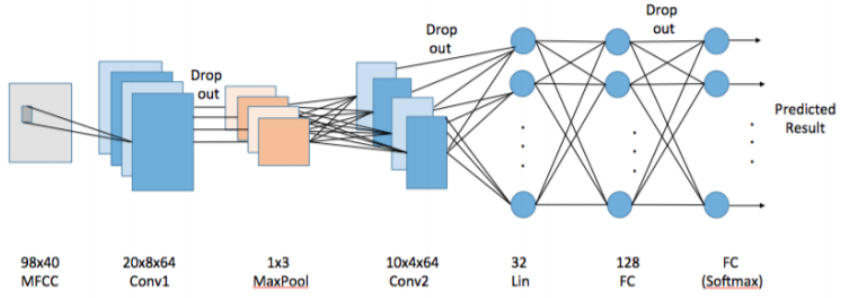
Hình 6. Mô hình DNN

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input (InputLayer)           [(None, 98, 40, 1)]       0
_________________________________________________________________
flatten (Flatten)            (None, 3920)              0
_________________________________________________________________
dense1 (Dense)               (None, 512)               2007552
_________________________________________________________________
dense2 (Dense)               (None, 512)               262656
_________________________________________________________________
dense3 (Dense)               (None, 512)               262656
_________________________________________________________________
pred (Dense)                 (None, 4)                 2052
=================================================================
Total params: 2,534,916
Trainable params: 2,534,916
Non-trainable params: 0
```

Hình 7. Kiến trúc mạng DNN

3.2.3. Mô hình Convolution Neural Network - CNN

Mô hình CNN rất phù hợp đối với bài toán phát hiện từ khóa (KWS) (Sainath & Parada, 2015), vì vậy bài báo đã triển khai một kiến trúc mạng CNN với hai lớp convolution, kiến trúc của mô hình được thể hiện trong Hình 8. Trong đó, một số lớp lớp Convolutions (Conv) sử dụng nhiều bộ lọc Conv để có được các đặc trưng khác nhau, lớp MaxPool giảm tỷ lệ lấy mẫu bằng cách thực hiện thao tác giảm số lượng tham số và tính toán trong mạng do đó kiểm soát được tình trạng overfitting, lớp Dropout chỉ giữ một neural kích hoạt với một xác suất *p* hoặc đặt nó bằng 0 để kiểm soát tình trạng overfitting, Lớp tuyến tính thấp (Lin) thực hiện phép nhân và cộng tuyến tính để chuyển đầu ra của lớp Conv đến các nút rời rạc, giảm các tham số và tính toán, kiểm soát overfitting, và lớp Kết nối đầy đủ (FC) lưu giữ thông tin đầy đủ hoặc đưa ra dự đoán *softmax* cuối cùng. Mô hình CNN được xếp các lớp như trong Hình 8. Trong mô hình chỉ áp dụng 2 lớp Conv thay vì áp dụng các lớp CNN rất sâu và lớn là để hạn chế số lượng tham số. Mô hình này đã đảm bảo được số lượng tham số dưới 250K, điều này khả thi đối với bài toán KWS có kích thước nhỏ trên thiết bị di động có dung lượng bộ nhớ bị hạn chế. Số lượng các tham số trong mô hình này được hiển thị như trong Bảng 1.



Hình 8. Kiến trúc mô hình CNN

Bảng 1. Tham số của kiến trúc mạng CNN

Loại	Ht.	Wd.	Độ sâu	Bước Ht.	Bước Wd.	Tham số
Conv1	20	8	64	1	3	10.2k
Conv2	10	4	64	1	1	164.8k
Lin	-	-	32	-	-	65.5k
DNN	-	-	128	-	-	4.1k
Softmax	-	-	6	-	-	0.7k
Total	-	-	-	-	-	244.4k

4. THỰC NGHIỆM VÀ KẾT QUẢ

4.1. Quá trình huấn luyện

4.1.1. Khởi tạo

Các trọng số được khởi tạo ngẫu nhiên với giá trị trung bình bằng 0 và độ lệch chuẩn được chỉ định để không bị mất cân bằng. Vì giá trị cuối cùng của trọng số trong mạng được huấn luyện là chưa biết, nhưng với việc chuẩn hóa dữ liệu có thể giả định rằng khoảng một nửa trọng số sẽ là dương và một nửa trong số đó là âm. Do đó, các trọng số được điều chỉnh gần bằng 0, bởi vì nếu mọi nueral trong mạng tính toán cùng một đầu ra, thì tất cả chúng cũng sẽ tính toán cùng một đạo hàm trong quá trình lan truyền ngược và thực hiện cập nhật thông số giống hệt nhau.

4.1.2. Kích thước lô (Batch size)

Kích thước lô cho đạo hàm bằng 100. Đối với mỗi bước được huấn luyện ngẫu nhiên 100 mẫu, điều này có thể sẽ làm mất mối tương quan giữa chúng và làm cho mạng học hiệu quả hơn. Trong mô hình kích thước lô được chọn là khác 1 để tránh overfitting. Tuy nhiên, kích thước lô cũng không được quá lớn, vì việc huấn luyện một mạng nueral có thể tốn nhiều chi phí tính toán. Vì vậy, giá trị này là sự đánh đổi giữa hiệu suất và giới hạn phần cứng.

4.1.3. Tốc độ học (learning rate)

Tốc độ học là 0.001 cho tổng số 5/6 bước đầu tiên, tiếp theo là 0.0001 cho đến cuối. Tốc độ học cho bước sau tương đối nhỏ hơn để tinh chỉnh mô hình cho các bước sau đó.

4.1.4. Phương pháp cập nhật

Một nhược điểm của phương pháp giảm đạo hàm ngẫu nhiên là hướng cập nhật phụ thuộc hoàn toàn vào lô hiện tại. Vì vậy trong bài báo đã áp dụng phương pháp bản cập nhật Nesterov Momentum hoạt động tốt hơn. Đầu tiên là cập nhật một bước theo hướng ban đầu, sau đó tính toán đạo hàm sửa hướng cập nhật cuối cùng, được biểu thị bằng công thức như sau:

$$x_{t+1} = x_t + \Delta x_t \qquad (4.1)$$

$$\Delta x_t = p x_{t-1} - \eta \Delta f(x_{t-1} + p x_{t-1}) \qquad (4.2)$$

Trong đó, η là tỷ lệ học, và p giá trị xung lượng. Trong giai đoạn đầu, đạo hàm tương đối lớn với giá trị ban đầu p là 0,5 trong khi đó chọn p lớn để có đạo hàm nhỏ. Trong bài báo này giá trị xung lượng p được gán tăng dần qua các bước với giá trị là 0,5, 0,9, 0,95, 0,99 (http://cs231n.github.io/neural-networks-3/)

4.1.5. Cross Entropy

Đối với mỗi mô hình, lớp cuối cùng sử dụng một bộ phân loại *softmax* với cross entropy *L* để ước tính cho mỗi nhãn đầu ra phía sau được thể hiện bằng công thức sau:

$$L_i = -log\left(\frac{e^{f_{yi}}}{\sum_j e^{f_j}}\right) \qquad (4.3)$$

Trong đó $f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$ là hàm *softmax*

4.2. Kết quả và thảo luận

Bài báo sử dụng accuracy, precision, recall and loss làm thước đo hiệu suất của mô hình, đồng thời cũng vẽ được hai loại đồ thị đường cong ROC/AUV và đồ thị đường cong precision-recall để chứng minh hiệu suất của 3 mô hình: Vanilla 1 lớp ẩn, Deep Neural Network và Convolution Neural Network. Các số liệu được tính toán dựa theo công thức sau:

$$Precision = \frac{TP}{TP+FP} \qquad (1)$$

$$Recall = \frac{TP}{TP + FN} \qquad (2)$$

Trong một bài toán phân loại không cân bằng với nhiều hơn hai lớp, Precision được tính bằng tổng số true positives (TP) trên tất cả các lớp chia cho tổng số true positives và false positives (FP) trên tất cả các lớp:

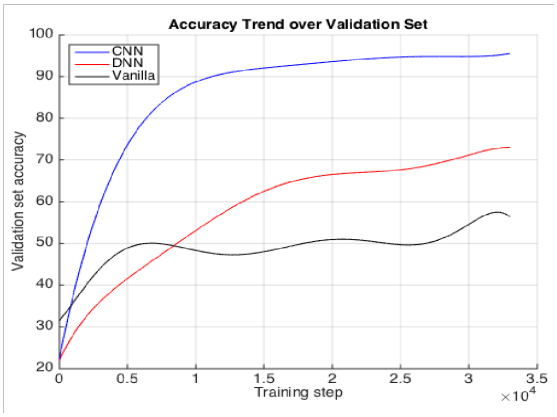
Precision = Sum c in C TruePositives_c / Sum c in C (TruePositives_c + FalsePositives_c) (3)

Tương tự như trên Recall được tính bằng tổng số true positives (TP) trên tất cả các lớp chia cho tổng số true positives và false nagatives (FN) trên tất cả các lớp:

Recall = Sum c in C TruePositives_c / Sum c in C (TruePositives_c + FalseNegatives_c) (4)

4.2.1. Độ chính xác và Mất mát

Trong bài báo sử dụng 6 nhãn (label) *"up"/"down"/"left"/"right"/ "silence"/"unknown"* cho bài toán KWS, được thực hiện 33.000 bước huấn luyện cho 3 mô hình và cứ sau 400 lượt sẽ thực hiện kiểm tra đánh giá. Sau quá trình huấn luyện sẽ thực hiện dự đoán trên tập dữ liệu test để có được độ chính xác của mô hình trên tập dữ liệu thử nghiệm kết quả được thể hiện trên Hình 9 và Bảng 2.



Hình 9. Độ chính xác của 3 mô hình trên tập đánh giá

106	0	0	0	0	0
1	76	7	15	5	2
0	6	264	0	1	1
0	2	2	246	3	0
0	1	2	3	260	1
1	11	2	2	1	242

Hình 10. Ma trận hỗn loạn cho CNN

Bảng 2. So sánh độ chính xác

Mô hình	Độ chính xác trên tập Validation	Độ chính xác trên tập Test	Loss
CNN	95.1%	94.5%	0.190
DNN	72.5%	71.9%	1.048
Vanilla	57.3%	56.7%	3.640

Trong ma trận hỗn loạn ở Hình 10 cho mô hình CNN, các cột đại diện cho một tập hợp các mẫu được dự đoán là *"silence", "unknown", "up", "down", "left", "right"* trong đó tất cả các giá trị tại các vị trí đều rất nhỏ ngoại trừ đường chéo trung tâm, điều này cho thấy rằng mô hình CNN rất ít bị sai lầm.

4.2.2. Thu hồi (Recall), Độ chính xác (Precision), ROC và AUC

Các giá trị Recall, Precision được tính theo công thức *(1, 2, 3, 4)* với ma trận hỗn loạn trên bộ dữ liệu thử nghiệm đạt kết quả như thể hiện trong Bảng 3. Có thể thấy rằng hai đại lượng Precision và Recall bằng nhau là các số không âm và nhỏ hơn hoặc bằng một.

Bảng 3. Giá trị Recall và Precision cho 3 mô hình

	Cnn	Dnn	Vani
Giá trị Precision	0.9454	0.7866	0.5416
Giá trị Recall	0.9454	0.7866	0.5416

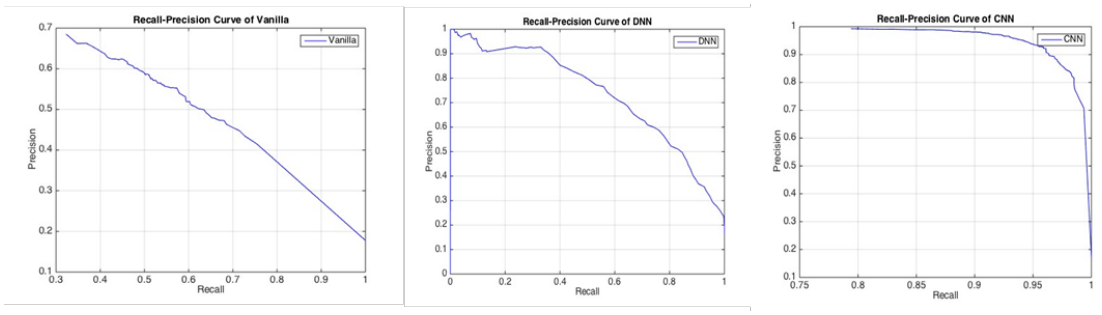
Đồng thời, hiệu quả của mô hình được đánh giá dựa trên việc thay đổi ngưỡng (threshold) từ 0 tới 1 và quan sát giá trị của Recall và Precision. Các đường cong ROC/AUC được vẽ dựa vào cách tính sau:

Giả sử có N ngưỡng để tính precision và reccall, với mỗi ngưỡng cho một cặp giá trị precision, recall là $P_n, R_n, n = 1, 2, \dots, N$. ROC/AUC được vẽ bằng cách vẽ từng điểm có tọa độ (R_n, P_n) trên trục tọa độ và nối chúng lại với nhau. AUC được xác định bằng:

$$AUC = \sum_n (R_n - R_{n-1}) P_n$$

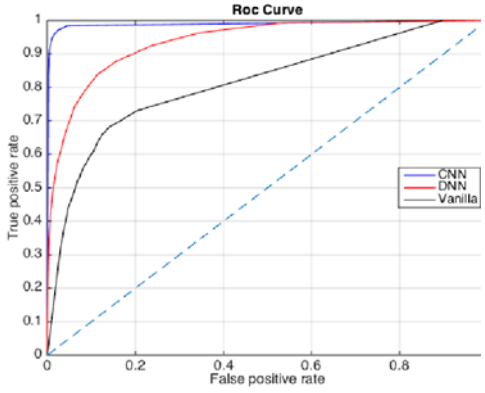
Trong đó $(R_n - R_{n-1})P_n$ chính là diện tích hình chữ nhật có có chiều rộng $(R_n - R_{n-1})$ và chiều cao P_n, đay cũng gần với cách tính tích phân dựa trên cách tính diện tích của từng hình chữ nhật nhỏ.

Các đường cong ROC/AUC được minh họa trong Hình 11, đồng thời so sách đường cong của 3 mô hình Valli, DNN và CNN như thể hiện trong Hình 12.



Hình 11. Recall và Precision cho Vanilla, DNN, CNN

Những kết quả này cho thấy CNN hiệu quả hơn DNN và Vanilla, với 18,6% cải tiến tương đối về giá trị Precision so với DNN và 72,3% so với Vanilla, 37,4% so với DNN và 61,8% so với Vanilla về giá trị Recall, trong đó Precision cao liên quan đến tỷ lệ dương thấp và Recall cao liên quan đến tỷ lệ âm thấp.

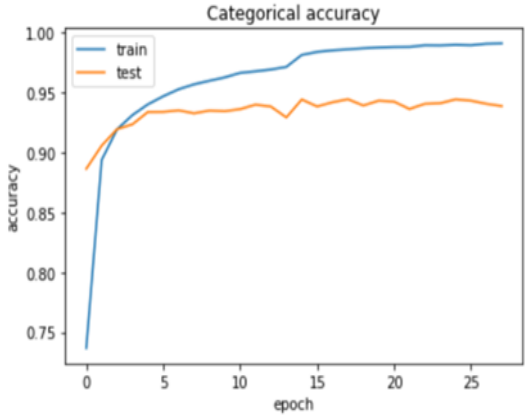


Hình 12. So sánh đường cong của 3 mô hình

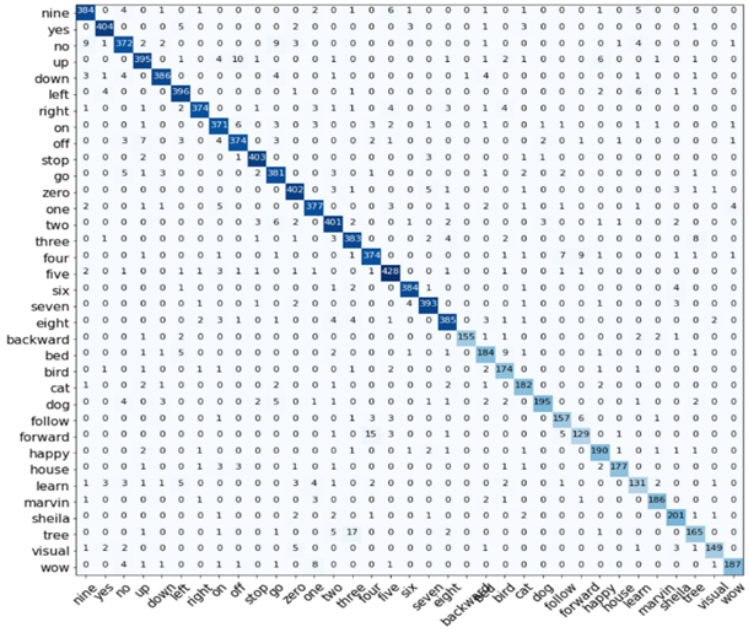
Đường precision-recall cho thấy sự đánh đổi giữa Precision và Recall cho các ngưỡng khác nhau. CNN có diện tích đường cong bao phủ cao nhất thể hiện cho cả recall cao và precision cao. Điều này có nghĩa là mô hình CNN đang trả về kết quả chính xác (độ chính xác cao), cũng như phần lớn các kết quả tích cực (recall cao).

ROC/AUC cho thấy rằng mô hình CNN đạt được nhiều AUC hơn, và tăng đáng kể ở một tỷ lệ dương giả rất thấp, đó là một kết quả mong muốn cho hệ thống nhận dạng tiếng nói điều khiển.

Ngoài việc thực nghiệm với giới hạn 6 nhãn đầu ra như *"silence", "unknown", "up", "down", "left", "right"*, trong nghiên cứu này cũng đã tiến hành thực nghiệm trên toàn bộ tập dữ liệu tiếng nói điều khiển từ Google, bao gồm 65.000 tập tin âm thanh tiếng nói với 35 từ khác nhau. Mô hình được huấn luyện với 60 epochs, kết quả thực nghiệm cho thấy rằng mô hình CNN hoạt động rất tốt và cho kết quả độ chính xác trên tập dữ liệu thử nghiệm đạt 94.5%. Kết quả được thể hiện trong Hình 13 và ma trận hỗn loạn như Hình 14.



Hình 13. Kết quả huấn luyện mô hình CNN trên toàn bộ tập dữ liệu



Hình 14. Ma trận hỗn loạn CNN trên toàn bộ tập dữ liệu

4.2.3. Kết quả đối sánh

Trong bài báo này, đã sử dụng 3 mô hình Vanilla, DNN và CNN cho bài toán KWS. Cả 3 mô hình này đều khai thác bộ phân loại *softmax* và khai thác đặc trưng MFCC. Kết quả thử nghiệm cho thấy mô hình CNN vượt trội hơn hai mô hình còn lại và đạt được độ chính xác tương đối là 31,43% và 66,67% đối với DNN và Vanilla; 82% và 94,8% về loss; 18,6% và 72,3% về giá trị chính xác; và 37,4% và 61,8% về giá trị recall. Kết quả của mô hình cũng được so sánh với một số công trình khác thực hiện trên cùng một bộ dữ liệu tiếng nói điều kiển từ Google bao gồm 65.000 tệp âm thanh tiếng nói với 35 từ khác nhau được thể hiện trong Bảng 4, mô hình DenseNet-121 (McMahan & Rao, 2017). ConvNet on raw WAV (Jansson, 2018). Kết quả mô hình CNN trên tập dữ liệu tiếng nói điều khiển được thể hiện cuối cùng:

Bảng 4. Kết quả đối sánh độ chính xác trên tập dữ liệu tiếng nói điều khiển từ Google

Model	Accuracy (%) 35 Word
DenseNet-121 No pretrain, no multiscale	80.13
DenseNet-121 No pretrain, multiscale	82.11
ConvNet on raw WAV	89.4
CNN	94.5

5. KẾT LUẬN

Trong bài báo, bộ dữ liệu tiếng nói điều khiển của Google bao gồm 65.000 tệp âm thanh tiếng nói với 35 từ khác nhau được sử dụng, mỗi tệp kéo dài trong một giây để thực hiện các thực nghiệm nhận dạng tiếng nói trên 3 mô hình đó là Vanilla một lớp ẩn, DNN 3 lớp ẩn và mô hình CNN. Kết quả cho thấy mô hình CNN thực hiện tốt hơn các mô hình còn lại và đạt độ chính xác là 94,5%. Một hạn chế của mô hình CNN là số lượng nhân lớn trong ConvLayer thứ hai do các ánh xạ đầu vào 3 chiều trải dài trên miền thời gian, tần số và đặc trưng. Vì vậy, trong những nghiên cứu tiếp theo sẽ đi sâu vào một số kiến trúc CNN mới với số nhân ít hơn và kết quả trong tương lai có thể khả thi để thực hiện điều khiển bằng tiếng nói trên một số thiết bị hạn chế tài về nguyên phân cứng.

TÀI LIỆU THAM KHẢO

Abdel-Hamid, O., Mohamed, A. R., Jiang, H., & Penn, G. (2012). Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In *2012 IEEE international conference on Acoustics, speech and signal processing (ICASSP)* (pp. 4277-4280). IEEE.

Chen, G., Parada, C., & Heigold, G. (2014, May). Small-footprint keyword spotting using deep neural networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4087-4091). IEEE.

Fernández, S., Graves, A., & Schmidhuber, J. (2007, September). An application of recurrent neural networks to discriminative keyword spotting. In *International Conference on Artificial Neural Networks* (pp. 220-229). Springer, Berlin, Heidelberg.

Jansson, P. (2018). Single-word speech recognition with Convolutional Neural Networks on raw waveforms.

Keshet, J., & Bengio, S. (2009). *Automatic speech and speaker recognition*. Large Margin and Kernel Methods, John Willy and Sons.

Li, K. P., Naylor, J. A., & Rossen, M. L. (1992, March). A whole word recurrent neural network for keyword spotting. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on* (Vol. 2, pp. 81-84). IEEE Computer Society.

McMahan, B., & Rao, D. (2018, April). Listening to the world improves speech command recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 32, No. 1).

Rohlicek, J. R., Russell, W., Roukos, S., & Gish, H. (1989, May). Continuous hidden Markov modeling for speaker-independent word spotting. In *International Conference on Acoustics, Speech, and Signal Processing*, (pp. 627-630). IEEE.

Rose, R. C., & Paul, D. B. (1990, April). A hidden Markov model based keyword recognition system. In *International Conference on Acoustics, Speech, and Signal Processing* (pp. 129-132). IEEE.

Russell, S., Dewey, D., & Tegmark, M. (2015). Research priorities for robust and beneficial artificial intelligence. *Ai Magazine*, *36*(4), 105-114.

Silaghi, M. C., & Bourlard, H. (1999). Iterative posterior-based keyword spotting without filler models. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding (ASRU'99) Workshop (No. CONF).*

Silaghi, M. C. (2005, April). Spotting subsequences matching an HMM using the average observation probability criteria with application to keyword spotting. In *AAAI* (pp. 1118-1123).

Schalkwyk, J., Beeferman, D., Beaufays, F., Byrne, B., Chelba, C., Cohen, M., ... & Strope, B. (2010). "your word is my command": Google search by voice: A case study. In *Advances in speech recognition* (pp. 61-90). Springer, Boston, MA.

Sainath, T. N., Mohamed, A. R., Kingsbury, B., & Ramabhadran, B. (2013, May). Deep convolutional neural networks for LVCSR. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 8614-8618). IEEE.

Sainath, T. N., & Parada, C. (2015). Convolutional neural networks for small-footprint keyword spotting. In *Sixteenth Annual Conference of the International Speech Communication Association.*

Tabibian, S., Akbari, A., & Nasersharif, B. (2011, June). An evolutionary based discriminative system for keyword spotting. In *2011 International Symposium on Artificial Intelligence and Signal Processing (AISP)* (pp. 83-88). IEEE.

Tóth, L. (2014, May). Combining time-and frequency-domain convolution in convolutional neural network-based phone recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 190-194). IEEE.

Warden, P. (2017). Speech commands dataset. https://ai. googleblog. com/2017/08/launching-speech-commands-dataset