GOLDEN OWL SOLUTIONS HO CHI MINH CITY

INTERNSHIP PROGRAM 2025

BÁO CÁO

BÀI KIỂM TRA ĐÁNH GIÁ NĂNG LỰC AI/ML INTERN

Người thực hiện: Hà Đức Huy

Thành phố Hồ Chí Minh, năm 2025

MỤC LỤC

LỜI MỞ ĐẦU

Báo cáo này trình bày kết quả bài kiểm tra đánh giá năng lực do công ty **Golden Owl Solutions** gửi nhằm đánh giá năng lực chuyên môn của em trước khi chính thức tham gia chương trình thực tập. Bài kiểm tra được thiết kế để kiểm tra kiến thức lập trình, khả năng giải quyết vấn đề thực tế, từ đó giúp công ty đánh giá mức độ phù hợp cũng như tiềm năng phát triển của ứng viên.

Trong quá trình thực hiện, em đã có cơ hội vận dụng những kiến thức đã học tại **Khoa Khoa học Máy tính và Khoa học Dữ liệu, Trường Đại học Khoa học Tự nhiên – Đại học Quốc gia TP. Hồ Chí Minh**, đồng thời nâng cao kỹ năng tư duy logic và rèn luyện cách trình bày giải pháp một cách rõ ràng, mạch lạc.

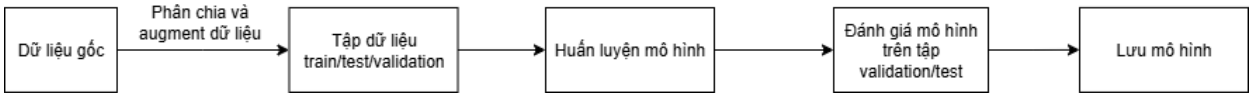
Em xin chân thành cảm ơn công ty **Golden Owl Solutions** đã tạo cơ hội thử sức và đánh giá năng lực, đồng thời cảm ơn các anh/chị đã hướng dẫn, hỗ trợ em trong suốt quá trình thực hiện.

Hà Đức Huy
Sinh viên Khoa Khoa học Máy tính & Khoa học Dữ liệu
Trường Đại học Khoa học Tự nhiên, ĐHQG TP. Hồ Chí Minh
TP. Hồ Chí Minh, 09/2025

1 Exercise 1 – Image classification

1.1 Huấn luyện mô hình phân loại ảnh chó mèo

Phần này trình bày quy trình xây dựng và huấn luyện mô hình phân loại hình ảnh chó/mèo sử dụng TensorFlow/Keras. Dữ liệu được lưu trữ trên Google Drive và xử lý trên Google Colab.



Hình 1: Sơ đồ huấn luyện mô hình

1.1.1 Chuẩn bị dữ liệu

- Dữ liệu gốc sử dụng được lấy từ link sau: Link Kaggle Dataset
- Bộ dữ liệu có 24,998 ảnh, gồm 12,499 ảnh con mèo và 12,499 ảnh con chó.
- Tạo tập tin create_data.py để thực hiện phân chia dữ liệu. Dữ liệu chia thành 3 tập train, validation và test với. Mỗi tập gồm 2 lớp: `cat` và `dog`.
- Tập train được augment với mỗi ảnh gốc sẽ tạo thêm 1 ảnh xoay theo góc bất kì và 1 ảnh được thêm nhiễu Gaussian. Dữ liệu tập validation và tập test được giữ nguyên.
- Thống kê số lượng ảnh từng lớp ở mỗi tập:
  - **Train**: 26247 ảnh mèo, 26247 ảnh chó
  - **Validation**: 2499 ảnh mèo, 2499 ảnh chó
  - **Test**: 1251 ảnh mèo, 1251 ảnh chó
- Tải dữ liệu lên Google Drive. Sau đó khi train sẽ kết nối Google Drive để truy cập dữ liệu.

1.1.2 Tiền xử lý và augmentation

- Sử dụng `ImageDataGenerator` để chuẩn hóa ảnh về [0,1], thực hiện dịch chuyển, lật ngang trên tập train.
- Tập validation và test chỉ chuẩn hóa.

1.1.3 Kiến trúc mô hình

Mô hình CNN được xây dựng với 4 khối convolution lần lượt có số lượng bộ lọc (filters) là 32, 64, 128 và 128. Mỗi khối bao gồm các lớp: `Conv2D`, `BatchNormalization`, `MaxPooling2D`, và `Dropout`.

Sau bốn khối convolution, mô hình được làm phẳng (`Flatten`) và đưa qua lớp `Dense` với 512 nút ẩn, kết hợp `BatchNormalization` và `Dropout` để giảm overfitting.

Cuối cùng, lớp đầu ra là `Dense(1)` với hàm kích hoạt `sigmoid` nhằm thực hiện phân loại nhị phân (chó/mèo).

1.1.4 Huấn luyện mô hình

Mô hình được biên dịch với:

- **Loss function**: `binary_crossentropy` (phù hợp cho phân loại nhị phân).
- **Optimizer**: Adam.
- **Metric**: Accuracy.

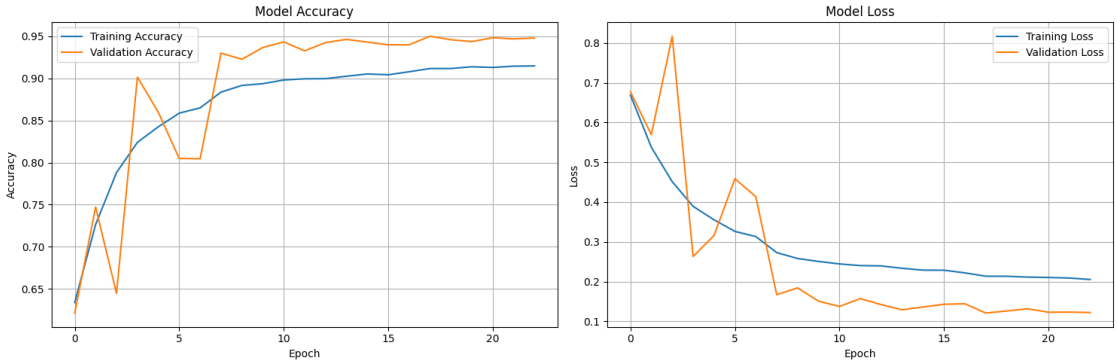
Trong quá trình huấn luyện, các `callback` được sử dụng nhằm tối ưu hiệu quả:

- `EarlyStopping`: Dừng sớm khi `val_loss` không cải thiện trong 5 epoch, đồng thời khôi phục trọng số tốt nhất.
- `ReduceLROnPlateau`: Giảm learning rate theo hệ số 0.2 nếu `val_loss` không giảm trong 3 epoch liên tiếp (tối thiểu 1×10^{-7}).
- `ModelCheckpoint`: Lưu mô hình tốt nhất dựa trên `val_accuracy` với tên gọi là `best_model.h5`.

Mô hình được huấn luyện trong **25 epoch**, với **batch size = 32**, sử dụng dữ liệu train và validation đã chuẩn bị từ trước. Sau khi huấn luyện xong, kết quả huấn luyện như sau:

- Độ chính xác trên tập huấn luyện: **0.9147**
- Độ chính xác trên tập validation: **0.9478**
- Loss trên tập huấn luyện: **0.2054**
- Loss trên tập validation: **0.1223**

Biểu đồ trực quan sự thay đổi accurancy và loss trong quá trình huấn luyện được thể hiện trong hình dưới:

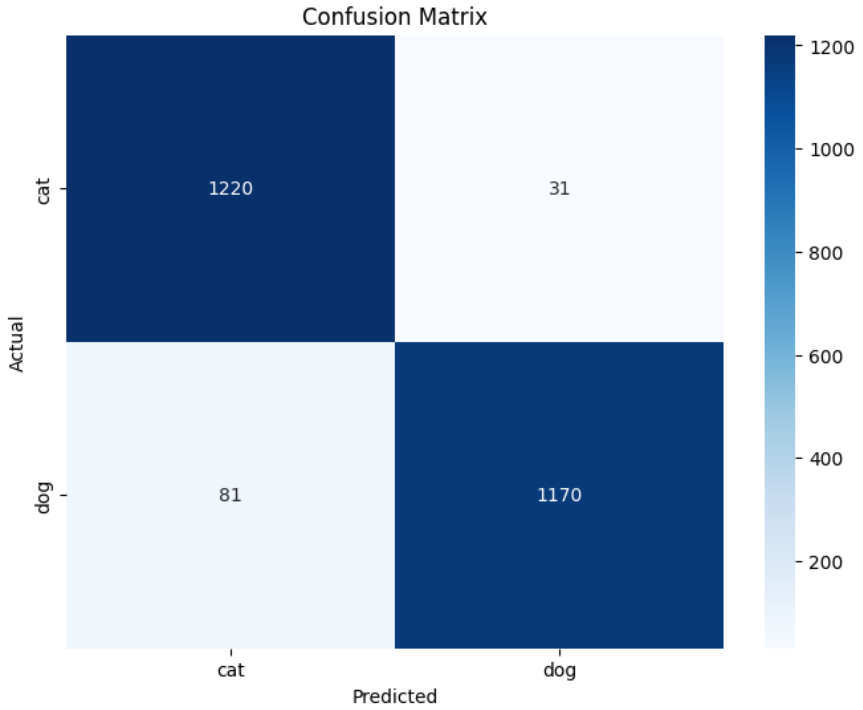


Hình 2: Biểu đồ trực quan sự thay đổi accurancy và loss trong quá trình huấn luyện

1.1.5 Đánh giá mô hình trên tập test

- Độ chính xác trên tập test: **0.9552**
- Loss trên tập test: **0.1124**

- Confusion matrix trên tập test như sau:



Hình 3: Confusion matrix trên tập test

- Báo cáo phân loại (classification report):

```
Classification Report:
              precision    recall  f1-score   support

         cat       0.94      0.98      0.96      1251
         dog       0.97      0.94      0.95      1251

    accuracy                           0.96      2502
   macro avg       0.96      0.96      0.96      2502
weighted avg       0.96      0.96      0.96      2502
```

Hình 4: Classification report trên tập test

1.1.6 Ví dụ dự đoán trên ảnh test

Mô hình được kiểm tra trên một số ảnh mẫu từ tập test. Kết quả dự đoán gồm tên lớp và độ tin cậy:

Hình 5: Một vài dự đoán thực tế trên ảnh test

1.1.7 Lưu mô hình

- Lưu mô hình dưới dạng `dog_cat_classifier_final.h5` (mô hình lưu đầy đủ) và `dog_cat_classifier _final_no_opt.h5` (mô hình không lưu optimizatiers parameters).
- Lưu model summary ra file `model_summary.txt`.
- Dữ liệu, mô hình và file notebook train mô hình được lưu ở: Link Drive

Mô hình CNN đã được xây dựng và huấn luyện thành công cho bài toán phân loại chó/mèo. Kết quả trên tập test cho thấy mô hình có độ chính xác tốt. Có thể cải thiện thêm bằng cách thử các kiến trúc sâu hơn hoặc sử dụng transfer learning.

1.2 Luồng xử lý trang web phân loại ảnh chó mèo

Mô hình sau khi được huấn luyện trên Google Colab sẽ được tích hợp vào ứng dụng để phục vụ dự đoán.

1.2.1 Công nghệ sử dụng và Source code

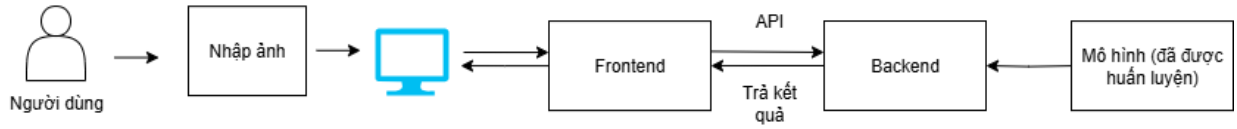
Em đã sử dụng các công nghệ sau để tạo trang web:

- **Frontend:** React + Vite
- **Backend:** FastAPI
- **Deploy frontend:** Netlify
- **Deploy backend:** Render

Source code của trang web được lưu tại: Link GitHub

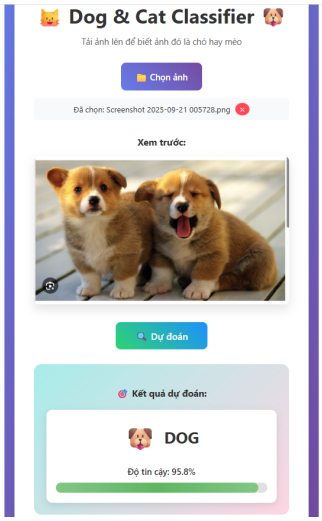
1.2.2 Quy trình hoạt động của hệ thống

Toàn bộ hệ thống hoạt động theo các bước chính sau:



Hình 6: Sơ đồ luồng xử lý trang web phân loại ảnh chó mèo

- Người dùng lựa chọn ảnh đầu vào trên giao diện frontend (web hoặc ứng dụng). Đây là dữ liệu hình ảnh thô mà hệ thống cần phân loại.
- Ảnh được gửi đến backend thông qua API `/predict`. Tại đây, hệ thống tiếp nhận dữ liệu và chuẩn bị cho bước xử lý tiếp theo.
- **Tiền xử lý dữ liệu hình ảnh:** Backend thực hiện các thao tác chuẩn hóa dữ liệu, bao gồm resize ảnh về kích thước chuẩn 224x224 pixel và chuẩn hóa giá trị điểm ảnh về khoảng [0,1]. Việc này đảm bảo dữ liệu đầu vào phù hợp với kiến trúc mô hình CNN đã huấn luyện.
- **Dự đoán bằng mô hình CNN:** Ảnh sau khi tiền xử lý sẽ được đưa vào mô hình học sâu (Convolutional Neural Network) để thực hiện suy luận. Mô hình này đã được huấn luyện trước trên tập dữ liệu chó/mèo, do đó có khả năng trích xuất đặc trưng và phân loại chính xác.
- **Trả về kết quả dự đoán:** Backend gửi lại phản hồi cho frontend dưới dạng nhãn phân loại (chó hoặc mèo), kèm theo xác suất dự đoán (confidence score) thể hiện độ tin cậy của kết quả.
- **Hiển thị kết quả cho người dùng:** Frontend nhận dữ liệu phản hồi, hiển thị trực quan trên giao diện để người dùng có thể dễ dàng xem và kiểm chứng.



Hình 7: Ví dụ kết quả đầu ra của trang web

1.3 Hạn chế và hướng phát triển

Mặc dù mô hình đạt độ chính xác cao trên tập test, vẫn còn một số hạn chế:

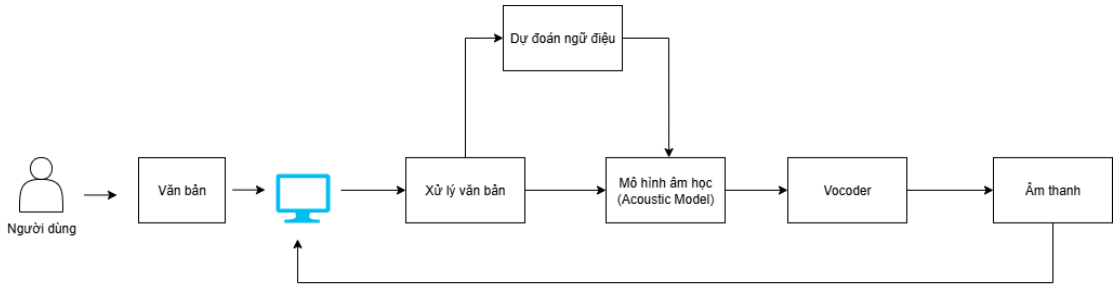
- Mô hình CNN tự xây dựng chưa khai thác được sức mạnh của các kiến trúc hiện đại như VGG16, ResNet hoặc EfficientNet.
- Dữ liệu huấn luyện chỉ giới hạn ở chó và mèo, nên khó mở rộng trực tiếp cho nhiều loài động vật khác.

Trong tương lai có thể cải thiện mô hình bằng cách:

- Thu thập thêm hình ảnh các loài khác để mô hình phân loại nhiều loài hơn.
- Ứng dụng **transfer learning** từ các mô hình pretrained.
- Tăng cường dữ liệu (augmentation) đa dạng hơn như thay đổi độ sáng, cắt ngẫu nhiên, zoom.

2 Exercise 2 – Text to Speech

Text-to-Speech (TTS) là công nghệ chuyển đổi văn bản thành giọng nói tự nhiên, giúp người dùng dễ dàng tiếp cận nội dung dưới dạng âm thanh. Nhờ trí tuệ nhân tạo, TTS ngày càng được ứng dụng rộng rãi trong giáo dục, y tế, trợ lý ảo và giải trí, góp phần nâng cao khả năng tiếp cận, xóa bỏ rào cản giao tiếp và tối ưu hóa trải nghiệm trong đời sống số hiện đại.



Hình 8: Sơ đồ của mô hình Text-to-Speech (TTS)

2.1 Xử lý văn bản

Trong hệ thống Text-to-Speech (TTS), xử lý văn bản là một bước quan trọng giúp chuyển đổi văn bản thô thành dạng chuẩn, dễ hiểu và sẵn sàng để đọc thành giọng nói tự nhiên. Quá trình này gồm nhiều giai đoạn kế tiếp nhau, từ phân tích cấu trúc đến chuẩn hóa và tách ngữ âm.

2.1.1 Phân tích cú pháp và cấu trúc văn bản:

Trước hết, hệ thống cần nắm được ý nghĩa và cấu trúc ngữ pháp của văn bản. Văn bản đầu vào sẽ được chia thành các câu nhỏ (phân đoạn câu), nhờ đó giữ được ngữ điệu phù hợp khi đọc. Ví dụ, văn bản "Hôm nay trời đẹp. Chúng ta đi dã ngoại nhé." sẽ được tách thành:

- Câu 1: "Hôm nay trời đẹp."
- Câu 2: "Chúng ta đi dã ngoại nhé."

Tiếp đó, mỗi từ trong câu được gắn nhãn từ loại (POS tagging) để xác định vai trò của chúng như danh từ, động từ hay đại từ. Chẳng hạn, trong câu "Chúng ta đi dã ngoại nhé." thì "Chúng, ta" là đại từ, "đi" là động từ và "dã ngoại" là danh từ.

Cuối cùng là phân tích cú pháp, giúp xác định quan hệ giữa các thành phần trong câu. Ví dụ, câu "Tôi thích đọc sách." được phân tích thành:

- Chủ ngữ: "Tôi"
- Động từ: "thích"
- Tân ngữ: "đọc sách."

2.1.2 Chuẩn hóa văn bản (Text Normalization):

Sau khi đã hiểu cấu trúc, văn bản sẽ được chuẩn hóa để hệ thống có thể phát âm chính xác. Công đoạn này bao gồm:

- **Xử lý từ viết tắt:** Chẳng hạn, ta sẽ chuyển "P.T.HCM" thành "Phố Thị Hồ Chí Minh"
- **Chuyển đổi số và ký hiệu:** Ví dụ "2025" thành "Hai nghìn không trăm hai mươi lăm", hoặc "12%" thành "Mười hai phần trăm"
- **Xử lý từ đồng âm, đồng nghĩa theo ngữ cảnh:** Sử dụng ngữ cảnh để phân biệt "quả táo" (danh từ) với "táo bạo" (tính từ).
- **Xử lý dấu câu:** Điều này quyết định ngữ điệu khi đọc, chẳng hạn dấu chấm tạo ngữ điệu xuống, dấu hỏi làm giọng lên cao, và dấu cảm thán giúp tăng cảm xúc.

Việc chuẩn hóa văn bản là bước quan trọng giúp hệ thống TTS hiểu chính xác nội dung cần đọc và truyền tải đúng sắc thái ngôn ngữ. Quá trình chuẩn hóa này đảm bảo rằng giọng đọc AI không chỉ chính xác về mặt nội dung mà còn tự nhiên và giàu sắc thái như lời nói của con người.

2.1.3 Phân đoạn ngữ âm:

Bước cuối cùng là phân đoạn ngữ âm, tức chuyển đổi chữ viết sang các đơn vị âm vị. Đây là nền tảng để biến văn bản thành tín hiệu âm thanh.

- **Chuyển đổi chữ sang các âm vị (G2P):** Ví dụ, "Xin chào" sẽ thành [x, i, n] [ch, a, o].
- Áp dụng quy tắc phát âm theo ngữ cảnh, nghĩa là cùng một từ có thể phát âm khác nhau tùy trường hợp. Chẳng hạn từ "đọc":
  - "Tôi đọc sách." → phát âm [dọc].
  - "Sách này để đọc." → phát âm [đọc].

Nhờ bước này, hệ thống có thể đảm bảo rằng giọng đọc AI không chỉ chính xác về chữ nghĩa mà còn phản ánh đúng cách phát âm tự nhiên của tiếng Việt.

2.2 Dự đoán ngữ điệu

Trong hệ thống tổng hợp tiếng nói, dự đoán ngữ điệu (prosody prediction) giữ vai trò quyết định giúp giọng đọc trở nên tự nhiên, mạch lạc và giàu cảm xúc. Quá trình này tập trung vào việc tạo ra sự nhấn nhá phù hợp, điều chỉnh cao độ, trường độ và cường độ của giọng nói, giúp tránh tình trạng đơn điệu khi đọc văn bản.

2.2.1 Các yếu tố chính của ngữ điệu

- **Cao độ (Pitch):** Đây là yếu tố quyết định sự lên xuống của giọng nói trong câu. Ví dụ, với câu hỏi "Bạn khỏe không?", giọng thường lên ở cuối câu để tạo cảm giác thắc mắc, trong khi câu khẳng định như "Tôi rất khỏe." sẽ giữ giọng bằng và dứt khoát ở cuối.
- **Trường độ (Duration):** Xác định độ dài khi phát âm các âm vị. Những từ khóa, mang tính nhấn mạnh sẽ thường được kéo dài hơn. Ví dụ, trong câu "Rất vui được gặp bạn.", các từ "rất" và "gặp" thường có độ ngân dài để nhấn ý.
- **Cường độ (Intensity):** Liên quan đến âm lượng và mức độ biểu cảm. Chẳng hạn, câu cảm thán như "Tuyệt vời!" sẽ có âm lượng lớn, mạnh mẽ và đầy cảm xúc, trong khi câu thường "Chúng ta đi thôi." lại được đọc với âm lượng vừa phải và đều đặn.

2.2.2 Tích hợp ngữ cảnh trong ngữ điệu

Dự đoán ngữ điệu không chỉ dừng lại ở mức phát âm mà còn cần tích hợp ngữ cảnh để đảm bảo tính chính xác và tự nhiên. Một số thách thức chính là:

- **Xử lý từ đa nghĩa:** Cùng một từ có thể mang ý nghĩa và cách phát âm khác nhau tùy vào vị trí trong câu. Ví dụ, từ "đọc" trong câu "Tôi thích đọc sách." mang nghĩa động từ, trong khi ở câu "Sách này để đọc." lại được hiểu là danh từ.
- **Ngữ cảnh văn bản:** Giọng đọc phải phản ánh cảm xúc phù hợp với nội dung. Chẳng hạn, câu "Tôi thật sự thất vọng." cần được đọc chậm rãi với giọng buồn, trong khi câu "Chúng ta thắng rồi!" lại cần sự hào hứng, phấn khởi.

Hai bước **dự đoán ngữ điệu** và **tích hợp ngữ cảnh** bổ trợ chặt chẽ cho nhau. Ngữ điệu mang lại sự sống động cho từng câu nói, trong khi ngữ cảnh đảm bảo rằng cảm xúc và ý nghĩa được thể hiện đúng chỗ. Khi kết hợp lại, chúng giúp giọng đọc AI không chỉ chính xác về nội dung mà còn truyền tải được sắc thái, cảm xúc và sự tự nhiên, từ đó nâng cao trải nghiệm người nghe và mở rộng khả năng ứng dụng của công nghệ TTS trong thực tế.

2.3 Mô hình âm học (Acoustic Model)

Mô hình âm học là một thành phần trung tâm trong hệ thống Text-to-Speech (TTS), chịu trách nhiệm chuyển đổi văn bản đã được xử lý sang các quang phổ âm thanh. Đây là bước trung gian quan trọng để từ văn bản có thể tạo ra giọng nói tự nhiên và giàu cảm xúc. Các mô hình học sâu hiện đại như Tacotron, Tacotron2, FastSpeech, FastSpeech2 hay FastPitch được thiết kế nhằm tối ưu quá trình này, đảm bảo đầu ra vừa chính xác vừa hiệu quả.

2.3.1 Vectơ hóa âm vị (Phoneme Vectorization)

Sau khi văn bản được xử lý và phân tách thành âm vị, hệ thống cần biểu diễn các âm vị đó dưới dạng số để mô hình học sâu có thể xử lý. Mỗi âm vị được ánh xạ thành một số

nguyên, rồi chuyển thành vectơ nhiều chiều chứa thông tin về ngữ điệu, âm sắc và tần số cơ bản.

- **Ví dụ:** Âm vị "k" có thể được mã hóa thành số 5, rồi biểu diễn thành một vectơ như [0.2, 0.8, 0.3, 0.7].
- Việc vectơ hóa giúp giữ lại đặc trưng ngữ nghĩa và ngữ âm của từng đơn vị, đồng thời chuẩn bị dữ liệu đầu vào cho các mô hình học sâu.

2.3.2 Dự đoán quang phổ âm thanh

Quang phổ âm thanh (spectrogram) là biểu đồ thể hiện sự thay đổi tần số theo thời gian, phản ánh cường độ của từng tần số tại mỗi thời điểm. Đây là dữ liệu trung gian quan trọng để tổng hợp giọng nói. Trong TTS, các mô hình học sâu sẽ nhận đầu vào là các vectơ âm vị đã được mã hóa, rồi dự đoán ra quang phổ. Các đặc trưng âm thanh quang phổ biểu diễn bao gồm:

- **Tần số cơ bản (Pitch Frequency):** Quyết định cao độ của giọng nói.
- **Cường độ (Amplitude):** Biểu diễn độ mạnh hay nhẹ của âm thanh, ảnh hưởng tới cảm xúc và nhấn nhá.
- **Thời lượng (Duration):** Khoảng thời gian phát âm của từng âm vị.

Ví dụ: Với câu "Xin chào", quang phổ sẽ biểu diễn cách phát âm từng âm vị theo thời gian, như cao độ tăng hoặc giảm, thời gian kéo dài của từ "chào".

2.3.3 Các mô hình âm học tiêu biểu

- **Tacotron2:** Một trong những mô hình tiên phong, sử dụng kiến trúc seq2seq với attention. Nó trực tiếp dự đoán Mel-spectrogram từ chuỗi văn bản hoặc âm vị. Kết quả là quang phổ có chất lượng cao, nhưng tốc độ xử lý còn hạn chế vì mang tính tự hồi quy (autoregressive).
- **FastSpeech2:** Được thiết kế theo hướng phi hồi quy (non-autoregressive), nên tốc độ sinh quang phổ nhanh gấp nhiều lần so với Tacotron 2. Mô hình này còn dự đoán trực tiếp các yếu tố như pitch và duration, giúp tăng độ tự nhiên và khả năng kiểm soát giọng nói.
- **FastPitch:** Tập trung mạnh vào việc kiểm soát cao độ. Điều này giúp giọng tổng hợp giàu cảm xúc hơn, tránh tình trạng đơn điệu thường gặp trong các hệ thống TTS cũ.

2.3.4 Vai trò của mô hình âm học

Mô hình âm học đóng vai trò cầu nối then chốt giữa ngôn ngữ và âm thanh. Nếu dự đoán quang phổ chính xác, giọng nói sinh ra sẽ mượt mà, có nhịp điệu và sắc thái cảm xúc giống giọng người thật. Đây chính là nền tảng để các bước tiếp theo, như Vocoder, có thể tái tạo giọng nói chất lượng cao từ dữ liệu trung gian này.

2.4 Vocoder

Vocoder là mô hình chuyển đổi đặc biệt, được thiết kế để biến quang phổ âm thanh (Mel-spectrogram) thành tín hiệu sóng âm nghe được. Đây là cầu nối cuối cùng trong hệ thống Text-to-Speech (TTS), đảm bảo quang phổ được tái tạo thành giọng nói tự nhiên, mượt mà và có đầy đủ sắc thái.

Các Vocoder hiện đại không chỉ tái tạo cao độ, thời lượng, cường độ mà còn có khả năng biểu đạt cảm xúc, nhấn nhá, giúp giọng nói gần như không thể phân biệt với người thật.

2.4.1 Quy trình hoạt động của Vocoder

1. **Bước 1:** Nhận đầu vào là quang phổ âm thanh
   - Quang phổ Mel-spectrogram được đưa vào Vocoder dưới dạng ma trận số.
   - Ví dụ: Quang phổ biểu diễn câu "Xin chào" cho biết tại từng thời điểm, các tần số và cường độ xuất hiện như thế nào.
2. **Bước 2:** Phân tích và tổng hợp âm thanh
   - Vocoder xử lý các đặc trưng như cao độ (pitch), độ lớn (amplitude), nhịp điệu và thời lượng (duration).
   - Quá trình này biến dữ liệu rời rạc trong quang phổ thành tín hiệu âm thanh liên tục.
3. **Bước 3:** Sinh tín hiệu âm thanh cuối cùng
   - Kết quả là sóng âm kỹ thuật số có thể phát qua loa hoặc tai nghe.
   - Ví dụ: Vocoder tái tạo giọng người đọc "Xin chào" với cao độ đi lên ở cuối từ "chào", mang lại cảm giác thân thiện.

2.4.2 Các mô hình Vocoder tiêu biểu

- **WaveNet:** Vocoder tiên phong dựa trên mạng nơ-ron hồi tiếp (RNN), tái tạo sóng âm ở mức mẫu (sample-level) nên cho âm thanh chi tiết và tự nhiên, nhưng tính toán nặng và tốc độ chậm, khó đáp ứng ứng dụng thời gian thực.
- **HiFi-GAN:** Vocoder dựa trên kiến trúc GAN, sinh âm thanh trực tiếp từ Mel-spectrogram với tốc độ rất nhanh, âm thanh mượt và chất lượng cao, hiện là chuẩn mực cho nhiều hệ thống TTS hiện đại.

2.4.3 Vai trò của Vocoder

Vocoder đóng vai trò quyết định trong việc biến quang phổ âm thanh (mel-spectrogram) thành dạng sóng hoàn chỉnh, từ đó nâng cao toàn bộ chất lượng giọng nói. Các vocoder hiện đại giảm méo tiếng, tái tạo ngữ điệu và cảm xúc, giúp giọng tổng hợp gần như giống giọng người thật. Với khả năng tạo ra âm thanh rõ ràng, giàu sắc thái, vocoder được ứng dụng rộng rãi trong nhiều lĩnh vực như trợ lý ảo, giáo dục, sách nói, giải trí, lồng tiếng phim hay dịch giọng nói theo thời gian thực.

2.5 Các vấn đề của TTS tiếng Việt và hướng giải quyết

Khi xây dựng mô hình Text-to-Speech (TTS) cho tiếng Việt, chúng ta có thể đối mặt với hai nhóm vấn đề chính: những khó khăn đặc thù của tiếng Việt và những thách thức kỹ thuật chung của các hệ thống TTS hiện đại.

2.5.1 Khó khăn đặc thù của tiếng Việt

- **Thanh điệu phức tạp:** Tiếng Việt có sáu thanh, chỉ cần phát âm sai cao độ hoặc thanh điệu là thay đổi nghĩa câu. Để giải quyết vấn đề này, cần xây dựng tập dữ liệu huấn luyện được gắn nhãn thanh điệu rõ ràng, đồng thời sử dụng các kỹ thuật chuyên biệt như pitch embedding hoặc tăng cường dữ liệu bằng cách biến đổi cao độ.
- **Khác biệt vùng miền:** Giọng Bắc, Trung, Nam có cách phát âm và ngữ điệu khác nhau. Giải pháp là thu thập dữ liệu đa vùng miền, đồng thời áp dụng kỹ thuật speaker embedding hoặc multi-speaker modeling để mô hình có thể tái hiện nhiều kiểu giọng đọc.
- **Nhiễu trong văn bản đầu vào:** Tiếng Việt trong văn bản đầu vào còn có nhiều yếu tố gây nhiễu như viết tắt, ký hiệu, số, hoặc từ ngoại lai. Nếu không xử lý chuẩn hóa trước khi đưa vào mô hình, đầu ra dễ bị sai sót hoặc đọc không tự nhiên. Vì vậy, cần xây dựng module chuẩn hóa văn bản (text normalization), chuyển đổi số và ký hiệu thành chữ, loại bỏ các ký tự không cần thiết, đảm bảo mô hình nhận đầu vào đã được chuẩn hóa.

2.5.2 Thách thức kỹ thuật chung của TTS hiện đại

- **Chất lượng giọng đầu ra:** Một trong những vấn đề quan trọng là chất lượng giọng nói đầu ra. Mặc dù các mô hình hiện đại như Tacotron 2 hay HiFi-GAN đã đạt chất lượng cao, nhưng đôi khi vẫn xảy ra lỗi như tiếng ồn nền, giọng bị "robotic", hoặc các đoạn âm thanh bị méo, thiếu mượt mà. Để cải thiện, cần sử dụng tập dữ liệu thu âm sạch, hậu xử lý lọc nhiễu, chuẩn hóa âm lượng, bổ sung hàm mất mát cảm nhận (perceptual loss).
- **Độ trễ và hiệu năng:** Một số vocoder (như WaveNet) cho chất lượng cao nhưng chậm và tốn tài nguyên. Giải pháp là ưu tiên các mô hình phi tự hồi quy (non-autoregressive) như FastSpeech 2 hoặc HiFi-GAN, vốn được tối ưu cho tốc độ. Đồng thời có thể áp dụng các kỹ thuật nén mô hình, lượng tử hóa hoặc distillation để giảm kích thước và tăng tốc độ suy luận.
- **Khả năng biểu cảm và ngữ điệu hạn chế:** Giọng nói nhân tạo đôi khi còn khô cứng, thiếu cảm xúc, khiến người nghe cảm thấy không tự nhiên. Để giải quyết, cần huấn luyện mô hình với dữ liệu giàu cảm xúc và đa dạng ngữ điệu, đồng thời khai thác các tham số điều khiển cao độ (pitch), thời lượng (duration) và cường độ (energy) để tinh chỉnh giọng nói đầu ra. Ngoài ra, việc áp dụng thêm các mô hình chuyên biệt cho prosody modeling cũng giúp tái hiện giọng đọc tự nhiên hơn.

- **Khả năng tổng quát hóa kém:** Khi gặp từ hiếm, tên riêng, hoặc cấu trúc ngữ pháp ít gặp, mô hình dễ phát âm sai hoặc tạo ra giọng nói bất thường. Để khắc phục, cần bổ sung tập dữ liệu phong phú và đa dạng hơn, đồng thời tận dụng cơ chế fine-tuning hoặc transfer learning để thích ứng với dữ liệu mới mà không phải huấn luyện lại toàn bộ mô hình từ đầu.
- **Tài nguyên và triển khai:** Các mô hình TTS hiện đại thường đòi hỏi GPU mạnh và bộ nhớ lớn, gây khó khăn khi triển khai trên thiết bị di động hoặc hệ thống nhúng. Giải pháp là sử dụng các mô hình nhẹ đã được tối ưu, áp dụng các framework tăng tốc suy luận như TensorRT hoặc ONNX Runtime, hoặc triển khai dạng dịch vụ đám mây.

Tài liệu tham khảo

[1] Bhavik Jikadara, "Cats and Dogs Classification Dataset". Available at: https://www.kaggle.com/datasets/bhavikjikadara/dog-and-cat-classification-dataset

[2] Duy Nguyen, "Cats vs Dogs Classification using CNN Keras". Available at: https://viblo.asia/p/cats-vs-dogs-classification-using-cnn-keras-1Je5EAx15nL

[3] Yiin AI, "TTS Là Gì? Cách Dùng Giọng AI Để Tạo Thu Nhập Online". Available at: https://www.yiin.ai/kb/tts-la-gi

[4] Yiin AI, "Nguyên lý hoạt động của công nghệ Text-to-Speech (TTS)". Available at: https://www.yiin.ai/kb/nguyen-ly-hoat-dong-cua-text-to-speech-tts