딥보이스 음성 탐지 프로젝트

프로젝트 개요

최근 인공지능 기술의 발전으로 인해 딥페이크(Deepfake) 음성 생성 기술이 빠르게 발전하고 있다. 이 기술은 가짜 음성을 진짜 음성처럼 들리게 만들어 다양한 사회적 문제를 야기할 수 있다. 이를 악용할 경우, 사기와 허위 정보 유포 등 다양한 악의적 목적을 가진 범죄와 명예 훼손 문제 등으로 사회에 심각한 위협이 될 수 있다. 이러한 문제를 해결하기 위해, 진짜 목소리와 가짜 목소리를 효과적으로 식별할 수 있는 기술 개발이 매우 중요해졌다.

본 프로젝트는 가짜 목소리와 진짜 목소리를 분류하는 알고리즘을 개발하는 것을 목표로 한다. 이를 위해 다양한 음성 데이터셋을 구축하고, 음성 데이터에서 유의미한 특징을 추출한 후, 머신러닝 및 딥러닝 모델을 학습시켜 가짜 목소리를 식별하는 모델을 개발하고자 한다. 최종적으로는 개발된 모델의 정확도와 실시간 처리 능력을 평가하여 신뢰성 높은 음성 식별 시스템을 구축하고자 한다. 특히, 기존에 한국어로 진위 여부를 판별할 수 있는 모델링이 부족한 상황에서, 본 연구가 한국어 가짜 음성 분류 모델 연구 분야의 출발점이 되고자 한다.

이 프로젝트는 다음과 같은 기대 효과를 가지고 있다. 가짜 목소리를 효과적으로 식별함으로써 음성 사기 및 악의적인 사용을 방지할 수 있으며, 정확한 음성 식별을 통해 사회적으로 신뢰성 높은 음성 정보를 제공할 수 있다. 또한, 본 프로젝트에서 개발한 기술은 음성 인식 및 처리 분야의 발전에 기여하고 다양한 응용 분야에서 활용될 수 있다.

프로젝트 배경**(Background)**

AI 기술로 가족의 목소리를 활용해 가짜 목소리를 이용한 보이스피싱이 기승을 부리면서, 이러한 사기 피해를 막기 위한 기술이 필요하게 되었다.
(https://www.news1.kr/articles/5415166 - 딥보이스를 이용한 보이스피싱 피해 사례)
(https://www.digitaltoday.co.kr/news/articleView.html?idxno=518520 - AI 관련 보이스피싱에 정부 대비 관련 기사)

음성 생성 AI 또는 딥보이스를 이용한 피싱 범죄는 꾸준히 증가중이며 정부에서도 2024년부터 신종범죄로 떠오르는 딥보이스 탐지에 초점을 맞춰 기술 개발이 필요하다는 것을 알렸다.
(https://www.mk.co.kr/news/economy/10785582 - 보이스피싱 관련 기사)
(https://news.mt.co.kr/mtview.php?no=2023020913433930492 - 딥보이스 사례 및 정부의견 기사)

유사한 연구**(Related Work)**

AI 보이스 탐지 시스템 (정수환 숭실대 정보통신전자공학부 교수)
https://www.aisrc.net/ 사이트에 음성 파일을 넣으면 어떤 목소리가 가짜인지 구분할 수 있다. 다양한 AI 생성기 모델 각각의 특징을 이용해 트레이닝 시켜 진짜 목소리와 가짜

목소리를 구분하도록 했고, 특히 억양과 감정 등 비언어적 요소에 주목했다. 진짜 목소리 중 감정 벡터 부분을 끄집어내어 AI 목소리와 구분하는 법을 집중적으로 연구했다.
(관련 기사 https://news.mt.co.kr/mtview.php?no=2024040413531816140)

딥보이스 탐지 기술 (딥브레인AI)
기존 딥보이스 탐지에 MFCC 기법을 주로 사용했는데 딥브레인AI에서는 특허 기술과 정보 추출 모델, 위변조 유무 판별 모델을 통합해 딥러닝 모델 탐지 성능을 개선했다. 또한 데이터 전처리 과정부터 변조 유무 판별, 결과 값의 후처리까지 전담하는 하나의 서버 파이프라인으로 구성해 단순 탐지를 넘어 솔루션으로 활용 가능하도록 했다. 모델의 객관적 탐지 성능을 정확도, ROC, EER(동일 오류율) 등 구체적 지표로 제시해 측정하기도 한다.
(관련 기사 https://www.aitimes.com/news/articleView.html?idxno=156974)

Generalization Of Audio Deepfake Detection (Tianxiang Chen, Avrosh Kumar, Parav Nagarsheth, Ganesh Sivaraman, Elie Khoury)
음성 딥페이크를 감지하는 연구. 가짜 음성을 구분하기 위해 딥러닝 및 음성 처리 기술(MFCCs) 을 사용한다. CNNs, RNNs 등을 이용한 모델로 음성을 구별한다. 학습 데이터를 기반으로 실제 음성과 가짜 음성간의 특징을 학습해 가짜 음성을 감지하는 패턴을 파악한다.
(https://www.researchgate.net/profile/Avrosh-Kumar/publication/345141913_Generalization_of_Audio_Deepfake_Detection/links/600cb38945851553a0678e07/Generalization-of-Audio-Deepfake-Detection.pdf)

Deepfake audio detection by speaker verification (Alessandro Pianese, Davide Cozzolino, Giovanni Poggi and Luisa Verdoliva)
훈련 단계에서 임의적으로 생성된 가짜 음성에 대해 일반화하기 어렵다는 문제점 발견. 그래서 특정한 가짜 음성 생성 방법에 의존하지 않고 실제 음성 데이터만을 사용해 훈련해 일반화가 자동으로 보장되도록 했다. 그 결과 좋은 성능과 높은 일반화 능력, 오디오 손상에 대한 높은 강건성을 얻었다.
https://arxiv.org/pdf/2209.14098

우리가 사용한 방법**(Methods)**

1. 데이터 수집

✔ Real voice data 수집
aihub의 "감정 분류를 위한 대화 음성 데이터셋"을 사용하였다. 해당 파일은 실제로 사람이 발화한 음성 데이터와 해당 음성의 대본을 포함하고 있다.

✔ Fake voice data 수집
네이버 클로바 보이스, openAI TTS, kt voice studio 등의 TTS 프로그램을 이용하여 가짜음성 데이터를 생성하였다. 특히 훈련 데이터와 음성을 맞추기 위해서 "감정 분류를 위한 대화 음성 데이터셋"에서 제공하는 스크립트를 TTS가 읽도록 하였다.

2. 데이터 전처리

✔ 모델의 인풋 데이터를 맞추기 위해서 전체 음성 데이터들을 모두 4초 단위로 분할하였다. 또한 각 음성의 sampling rate가 다르기 때문에 음성을 분할하는 과정에서 sampling rate를 모두 22050로 조절했다. (음성 데이터 분할 예시 코드)

✔ 실제 발화 음성에는 주변 소음이나 잡음이 존재하지만, TTS로 생성한 음성은 주변 잡음이 없기 때문에 잡음 제거를 고려하였고, meta에서 공개한 denoiser 모델을 이용하여 주변 잡음을 제거하였다. (잡음 제거 예시 코드)

3. 모델링

✔ 데이터 규모는 real/fake 데이터를 각각 3000개를 이용하였고, train : test = 8 : 2로 분할하여 훈련하였다. 4초로 분할을 진행했음에도 데이터의 차원이 다른 경우를 방지하기 위해서 데이터 차원을 고정하였다.

✔ Neural Net과 Convolutional Neural Net 두가지 모델로 모델링 진행하였다.

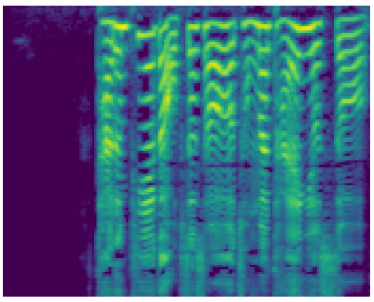
✔ Neural Net의 경우 sr(=22050) * 4로 88200 차원 벡터를 입력 받도록 모델링하였다.

✔ Convolutional Neural Net의 경우 mel spectrogram으로 변환 후 모델링을 진행하였는데, mel spectrogram이란 음성 데이터를 사람이 듣는 소리에 가깝게 시각적으로 표현할 수 있는 방법 중 하나이다. 음성 데이터를 아주 짧은 시간초 단위로 Fourier Transform을 진행하고 이때의 frequency domain 그래프에 mel transformation을 가한다. 이를 4초 전체에 대해서 반복하면 시간 축에 대한 mel-transformed-FFT 결과를 얻을 수 있다.
* mel transformation은 사람이 사람의 귀가 고주파의 신호보다 저주파의 신호를 더 잘 인식한다는 점을 고려한 변환이다.

실험 결과**(Experiments)**

Colab 노트북 실행 결과 붙여넣기

1. wav 데이터를 주파수 데이터로 변경
2. 주파수 데이터를 시각화

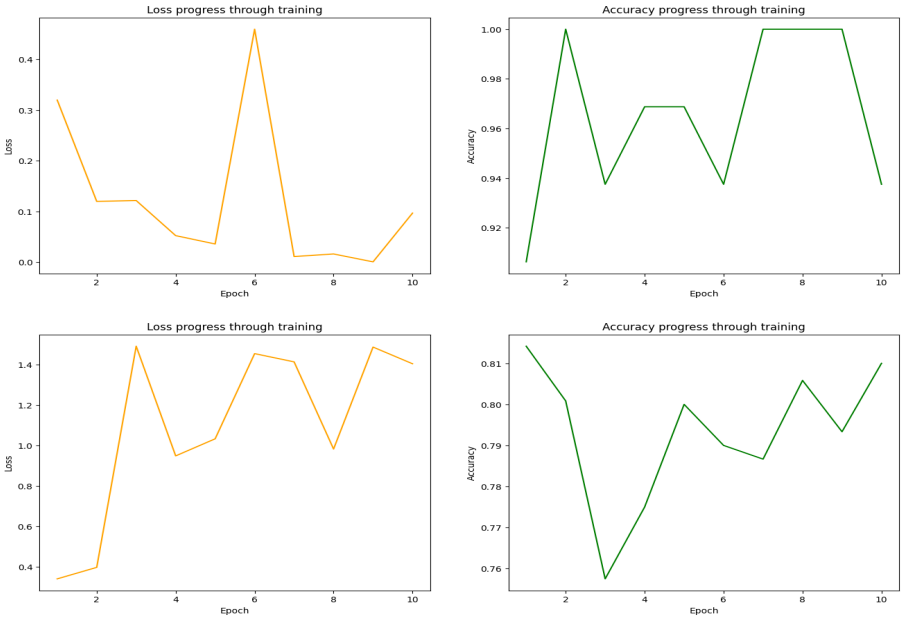


3. 뉴럴 네트워크를 이용한 오디오 분류 모델 구현

3-1. 모델 학습

```
Epoch 1:
1 150     loss: 0.7029    acc: 0.3750
2 150     loss: 0.6817    acc: 0.4375
3 150     loss: 0.6375    acc: 0.5312
4 150     loss: 0.5608    acc: 0.5625
5 150     loss: 0.6761    acc: 0.5000
6 150     loss: 0.6332    acc: 0.5938
7 150     loss: 0.4601    acc: 0.8750
8 150     loss: 0.5298    acc: 0.7188
9 150     loss: 0.4819    acc: 0.7500
10 150    loss: 0.5592    acc: 0.8125
11 150    loss: 0.5552    acc: 0.7812
12 150    loss: 0.5183    acc: 0.7500
13 150    loss: 0.5112    acc: 0.8125
14 150    loss: 0.6033    acc: 0.7188
15 150    loss: 0.5403    acc: 0.8750
16 150    loss: 0.5867    acc: 0.7500
17 150    loss: 0.6159    acc: 0.7500
18 150    loss: 0.5346    acc: 0.7812
19 150    loss: 0.6821    acc: 0.6875
20 150    loss: 0.6160    acc: 0.7500
...
130 150   loss: 0.0592    acc: 0.9688
131 150   loss: 0.0240    acc: 1.0000
132 150   loss: 0.0159    acc: 1.0000
133 150   loss: 0.1016    acc: 0.9688
134 150   loss: 0.0396    acc: 1.0000
135 150   loss: 0.0336    acc: 1.0000
136 150   loss: 0.0179    acc: 1.0000
137 150   loss: 0.0465    acc: 1.0000
138 150   loss: 0.0158    acc: 1.0000
139 150   loss: 0.0295    acc: 1.0000
140 150   loss: 0.0033    acc: 1.0000
141 150   loss: 0.0557    acc: 0.9688
142 150   loss: 0.0697    acc: 0.9688
143 150   loss: 0.0078    acc: 1.0000
144 150   loss: 0.0131    acc: 1.0000
145 150   loss: 0.0063    acc: 1.0000
146 150   loss: 0.0377    acc: 0.9688
147 150   loss: 0.0350    acc: 0.9688
148 150   loss: 0.0546    acc: 0.9688
149 150   loss: 0.4158    acc: 0.9375
150 150   loss: 0.0966    acc: 0.9375
```

3-2. 학습 결과 시각화



4. CNN을 이용한 오디오 분류 모델 구현

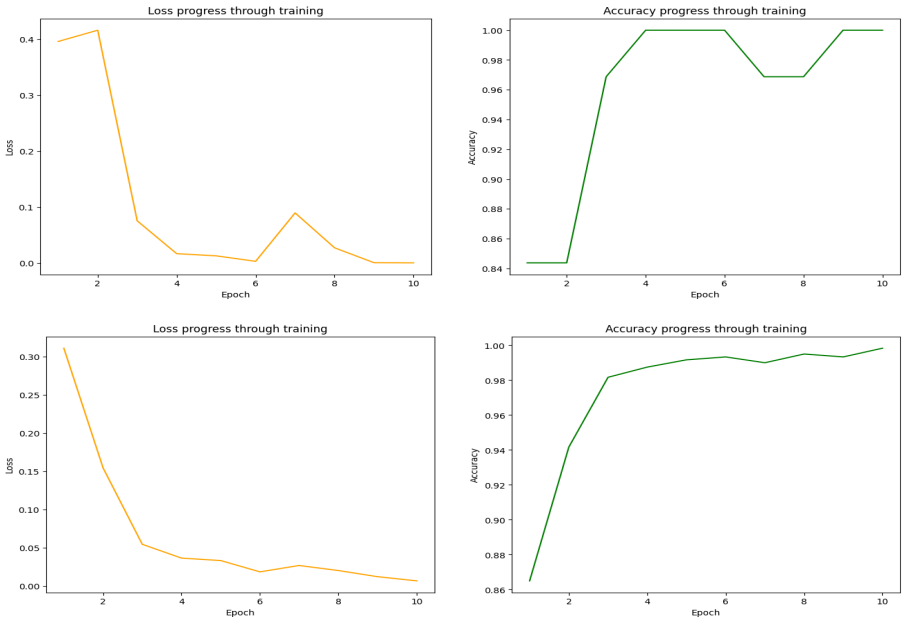
4-1. 모델 학습

```
Epoch 1:
1 150     loss: 0.7085    acc: 0.3438
2 150     loss: 0.7035    acc: 0.2188
3 150     loss: 0.7025    acc: 0.3438
4 150     loss: 0.6917    acc: 0.5312
5 150     loss: 0.6958    acc: 0.4375
6 150     loss: 0.6888    acc: 0.6250
7 150     loss: 0.6977    acc: 0.4062
8 150     loss: 0.6904    acc: 0.5625
9 150     loss: 0.6926    acc: 0.5000
10 150    loss: 0.6878    acc: 0.5938
11 150    loss: 0.6855    acc: 0.5938
12 150    loss: 0.6994    acc: 0.3750
13 150    loss: 0.6931    acc: 0.4375
14 150    loss: 0.6920    acc: 0.5625
15 150    loss: 0.6783    acc: 0.7188
16 150    loss: 0.6588    acc: 0.6875
17 150    loss: 0.6196    acc: 0.7188
18 150    loss: 0.7811    acc: 0.4375
19 150    loss: 0.6021    acc: 0.6562
20 150    loss: 0.6008    acc: 0.8125
```

...

```
130 150    loss: 0.0001    acc: 1.0000
131 150    loss: 0.0001    acc: 1.0000
132 150    loss: 0.0002    acc: 1.0000
133 150    loss: 0.0001    acc: 1.0000
134 150    loss: 0.0001    acc: 1.0000
135 150    loss: 0.0002    acc: 1.0000
136 150    loss: 0.0003    acc: 1.0000
137 150    loss: 0.0021    acc: 1.0000
138 150    loss: 0.0003    acc: 1.0000
139 150    loss: 0.0000    acc: 1.0000
140 150    loss: 0.0000    acc: 1.0000
141 150    loss: 0.0001    acc: 1.0000
142 150    loss: 0.0032    acc: 1.0000
143 150    loss: 0.0001    acc: 1.0000
144 150    loss: 0.0001    acc: 1.0000
145 150    loss: 0.0001    acc: 1.0000
146 150    loss: 0.0000    acc: 1.0000
147 150    loss: 0.0000    acc: 1.0000
148 150    loss: 0.0001    acc: 1.0000
149 150    loss: 0.0001    acc: 1.0000
150 150    loss: 0.0000    acc: 1.0000
```

4-2. 학습 결과 시각화



- 학습 결과
  - 뉴럴네트워크 모델과 CNN 모델 학습 결과 accuracy가 1과 가깝게 나오거나 1이 나왔는데, 데이터가 단조롭고 학습량이 작아 과적합이 의심된다.
  - 다양한 목소리의 fake 데이터와 잡음 제거를 통해 성능 개선이 필요하다.

결론**(Conclusion)**

본 프로젝트에서 우리는 AI를 활용하여 진짜(real) 및 가짜(fake) 목소리 데이터를 분류(classification)할 수 있는 모델을 개발하였다. 기존에 존재하는 한국말 진위 여부 판별 데이터 세트가 없다는 점에서, 우리 연구가 한국어 가짜 음성 분류 모델 연구 분야의 출발점이 되기를 희망한다.