Master's Thesis of Engineering

Phonetic Similarity Detection on Trademark Name using CNN-Siamese Networks

CNN-Siamese 네트워크를 활용한 문자 상표 발음 유사성 탐지

August 2022

Graduate School of Engineering
Seoul National University

Industrial Engineering Major

Gi Jung Kim

Phonetic Similarity Detection on Trademark Name using CNN-Siamese Networks

CNN-Siamese 네트워크를 활용한 문자 상표 발음 유사성 탐지

Professor Sungzoon Cho

Submitting a master's thesis of Engineering

August 2022

Graduate School of Engineering
Seoul National University

Industrial Engineering Major

Gi Jung Kim

Confirming the master's thesis written by
Gi Jung Kim

August 2022

Chair	Myunghwan Yun	(Seal)
Vice Chair	Sungzoon Cho	(Seal)
Examiner	Jaewook Lee	(Seal)

Abstract

Phonetic Similarity Detection on Trademark Name using CNN-Siamese Networks

Gi Jung Kim
Department of Industrial Engineering
The Graduate School
Seoul National University

Recently, as the number of registered trademarks has rapidly increased, research to determine trademark similarity based on machine learning has been actively conducted. Similarity of trademarks is judged based on shapes, meaning, and pronunciation. In the case of pronunciation, there is a limit in judging similarity because the standards for similarity are ambiguous and spellings do not correspond to pronunciation in many cases. On the other hand, the performance of converting text into speech has been remarkably improved due to the recent development of speech synthesis technology. In this paper, we propose a deep learning framework that automatically determines the pronunciation similarity of trademarks using speech data converted using speech synthesis technology. First, after synthesizing the trademark text into speech, it is converted into a log Mel spectrogram, and feature learning is performed through a convolutional neural network with a triplet loss. To compare the proposed method with previous studies, the trademark text dataset provided by AIhub was used, and our proposed method showed superior performance than the

previous studies..

Keywords: CNN, Siamese Network, Trademark Similarity, Trademark Pronunciation, Speech Synthesis

Student Number: 2019-23889

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A trademark refers to a legally registered or established symbol, word or mark used to represent a company or product, and must be protected from theft or infringement as part of intellectual property rights. However, as the number of trademarks applied for recently is rapidly increasing, the difficulty in examining trademark registration is increasing. Figure 1.1 shows the number of trademark applications and trademark registration examinations by year in Korea. In Figure 1.1, it can be seen that the number of trademark applications sharply increased between 2017 and 2020, while the number of trademark examinations fell short of that.

Due to the above limitations, attempts to determine the similarity of trademarks based on machine learning have been actively studied in recent years. The similarity of trademarks is judged based on shapes, meaning, and pronunciation, and most of the machine learning based attempts have been developed around logo images. With the recent development of Deep Convolution Neural Network (DCNN), visual data can be better represented, and many achievements have been achieved in the field of trademark image similarity detection.

On the other hand, machine learning research based on pronunciation showed relatively slow development. The pronunciation similarity detection problem of trade-

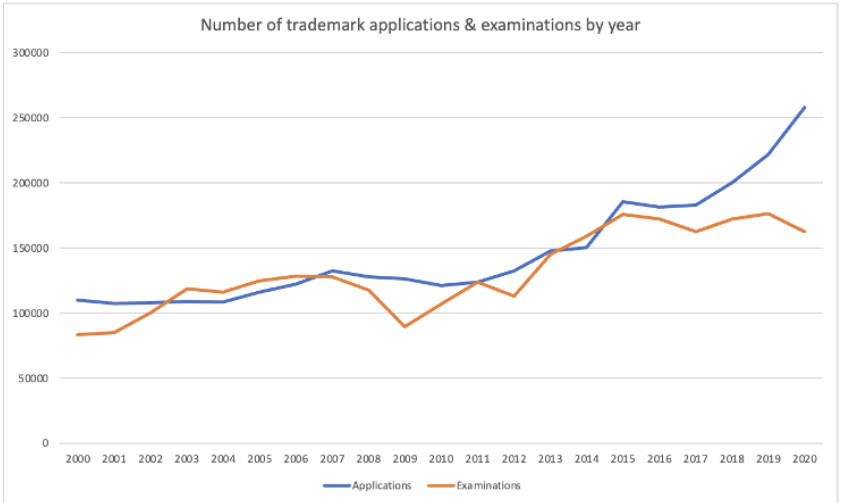

Figure 1.1: Number of trademark applications examinations by year

marks was mainly made using text data, and it was limited to calculating the distance between two texts through an algorithm ([2], [13])

In the case of judging the similarity of pronunciation of trademarks, it is judged based on the sound generated when the brand name is pronounced, not on the spelling of the brand name. However, due to the absence of audio data that correctly pronounces the trademark name and the fact that trademark examination is performed without using audio data, most of the preceding studies tried to detect trademarks with similar pronunciations using text data. However, similar spellings between texts are not necessarily similar in pronunciation, and vice versa. Figure 1.2 shows that the edit distance between the registered trademark and examining trademark is calculated through the Levenshtein Distance algorithm, which is used in previous studies. Levenshtein Distance judges similarity based on the number of edits of two letters, and there are cases where the distance of the registered trademark is closer than the distance of the rejected trademark. Through this, it can be

confirmed that spelling alone does not capture the similarity of actual pronunciation well.

Therefore, various studies have used a method of converting the spelling closer to pronunciation, such as Soundex and Double Metaphone (Fall & Giraud-Carrier [2]), to resolve the inconsistency between pronunciation and text. However, this method is also an indirect method of processing the spelling to be similar to the pronunciation, and there is a possibility that information may be lost in the process of preprocessing the data.

On the other hand, in recent years, text-to-speech performance has been remarkably improved due to the development of speech synthesis technology. An index for evaluating the quality of synthesized speech is represented by the Mean Opinion Score (MOS), and a recent study, Tan et al. [10] performance reached 4.58 based on LJSpeech Dataset [4]. A score between 4.5 and 4.8 is believed to be difficult to distinguish from human pronunciation.

Therefore, in this paper, we assumed that similarly pronounced trademark search performance can be improved by adopting a direct method of converting text data into speech data instead of an indirect method of pre-processing the text data.

This paper proposes a deep learning framework that automatically search trademarks with similar pronunciation. This paper does not attempt to judge similarity by using the text-based model that has been mainly used until now, but to determine the similarity after converting text data into speech data using recently developed speech synthesis technology. The proposed model is a structure using CNN and Siamese Network. It receives speech data converted to log Mel Spectrogram, calculates a feature vector using CNN model, and compares the calculated feature vectors

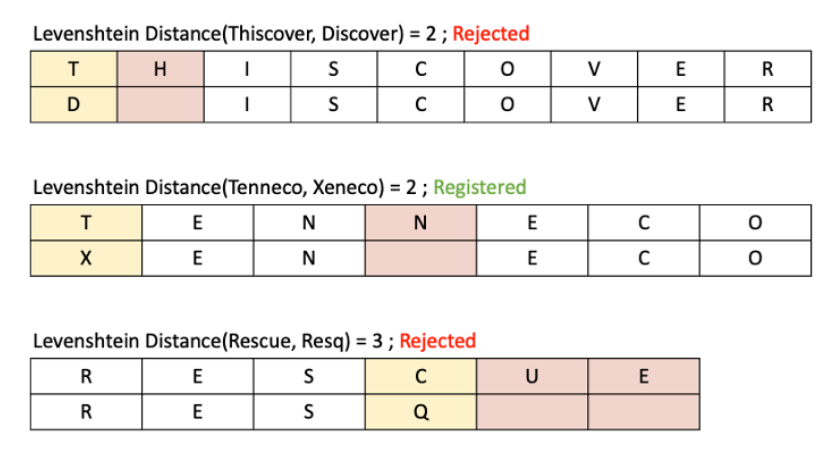

Figure 1.2: Comparison of registration and rejection of similar pronunciation trademarks using Levenshtein Distance

with each other to determine the similarity of pronunciation.

To verify the validity of the method proposed in this paper, various signal processing methods such as log Mel spectrogram, spectrogram, log spectrogram, and Mel spectrogram were tested. In addition, pre-processing methods such as zero padding, conversion to RGB image, and various backbone networks were also tested. As a result, it was shown that the proposed method is superior to the method proposed in previous studies.

Chapter 2

Related Work

Similar trademark search task has been actively studied mainly in the direction of searching for similar logo images, and research results of higher performance are being published due to the development of Convolutional Neural Network (CNN). The first study using CNN for similar trademark search was Tursun et al. [14]. In this study, AlexNet, GoogLeNet, and VGGNet16 were used to search for similar trademark images, and it was confirmed that the performance was superior to that of the traditional method that does not use deep neural network. Since then, the methodology has shifted from using a fully connected layer to only using the output of the pooling layer as a feature vector ([15], [17]). Recently, studies using object detection models such as Faster R-CNN and Yolo v4 ([12], [18]) or reinforcement learning have also been conducted ([16]).

In text-based similar trademark search studies, similarity in meaning, spelling, or pronunciation have been studied. Anuar et al. [1] conducted an experiment focusing on the semantic similarity between trademark names. This study points out that the existing key-word-based methodology has limitations in synonyms and suggests a method to extract text features using WordNet, which organizes English words into a set of synonyms called synsets.

There were also studies focusing on similarity in pronunciation between trademark names ([2], [5]). Fall & Giraud-Carrier [2] introduces and analyzes a pronunciation similarity search algorithm applicable to a trademark search system. Algorithms covered in this study include Edit distance (or Levenshtein algorithm), Modified edit, N-grams, Soundex, and Double metaphone. First, Edit distance defines the distance between two texts as the cost of replacing one text with another. Costs fall in three cases: insertion, deletion, and replacement. Modified edit is a modification of the Edit distance, where cost is applied to 1 instead of 2 when replacing neighboring spellings. N-grams is an algorithm that defines the similarity between two texts as the number of overlapping n-grams. Soundex is an algorithm that converts text by allocating codes between 0-6 to alphabets with similar pronunciation and determines that texts corresponding to the same Soundex are similar. The Double metaphone also converts alphabets into chords, like Soundex. The algorithm maps the alphabet to 12 consonants and an A (where the vowel is first). For example, in the case of MOBIGEL, since the pronunciation of G is similar to J and K, it can be converted into MPJL and MPKL. Both Soundex and Double metaphone have the disadvantage that the degree of similarity cannot be determined.

Ko et al. [5] mapped the trademark name converted to the international phonetic alphabet (IPA) into a two-dimensional image using 2-gram. The images were trained using CNN, and an accuracy of 92.7% was obtained when tested through Korea Trademark dataset.

Trappey et al. [13] proposes a new method for comprehensively judging the similarity of shapes, meaning, and pronunciations of trademarks using deep learning. For the similarity of image trademarks, CNN was used, and for the similarity of mean-

ing, Word2Vec was used. For pronunciation, Modified Edit, Soundex, and Double Metaphon proposed by Fall & Giraud-Carrier [2] were used to achieve about 95% performance. However, the number of data used for the validation was less than 100, and in the case of pronunciation similarity, there was a limit to using a spelling-based algorithm instead of a deep learning technique.

On the other hand, a study was conducted to detect a similarity in pronunciation using voice data (Zeghidour et al. [20]). In this study, a model was proposed to predict whether two voices were uttered by the same speaker and whether the pronunciations of the two voices were similar. Voice data was processed using Mel Fiterbanks Spectral Coefficient (MFSC), and Siamese network and Triamese network structures were used as the model structure.

Chapter 3

Proposed Method

In this paper, we propose a framework for learning the feature vector of log Mel spectrogram (or image) through Siamese network structure with CNN-based backbone and triplet loss.

3.1 Model Architecture

The Siamese network is described by Koch et al. [6] and is mainly used for the verification task to determine the similarity of two images. The Siamese network is a model that learns by receiving two images, outputting two feature embeddings, and adjusting the similarity of the two images, rather than predicting the class of each image. The name Siamese network was given because the structure of calculating two embeddings is like using two identical models. In this paper, since the log Mel spectrogram images are used for similarity detection, we believed it is appropriate to use the Siamese network structure.

Meanwhile, in this paper, the triplet marginal loss is used as the loss function, not the contrastive loss used by [6]. Looking at the contrastive loss in Equation 3.1, it trains to minimize the distance when two receiving image classes are the same, and in other cases, it trains to move further away. However, since the two images are

randomly selected, there are more probabilistic cases in which the two images are of different classes than the case of the same class. Therefore, this imbalance problem can negatively affect the training (Moon et al. [7]).

Triplet margin loss is a loss function that is first introduced by Schroff et al. [8] It calculates three input images, Anchor, Positive, and Negative rather than two images. Therefore, unlike contrastive loss, it has the advantage of more stable learning because the ratio of positive and negative cases can be kept the same.

$$\text{Contrastive Loss: } \mathcal{L}(x_i, x_j) = 1[y_i = y_j]||f(x_i)-f(x_j)||_2^2+1[y_i \neq y_j]\max(0, \lambda-||f(x_i)-f(x_j)||_2^2) \tag{3.1}$$

Each training batch consists of triplet (Anchor, Positive, and Negative). In this experiment, Anchor is a rejected trademark name and Positive is a similarly pronounced pre-registered trademark name. Negative is a trademark name randomly sampled from the dataset excluding Anchor and Positive. This training batch is fed to Siamese Network structure, a model with equal weights. It then outputs a feature vector each for Anchor, Positive, and Negative and is used to compute triplet margin loss (Equation 3.2). In triplet margin loss the distance from the Anchor to Positive is minimized, and the distance from the Anchor to Negative is maximized. (Figure 3.1)

$$\text{Triplet Margin Loss: } \mathcal{L}(A, P, N) = \max(||f(A) - f(P)||_2^2 - ||f(A) - f(N)||_2^2 + \alpha, 0) \tag{3.2}$$

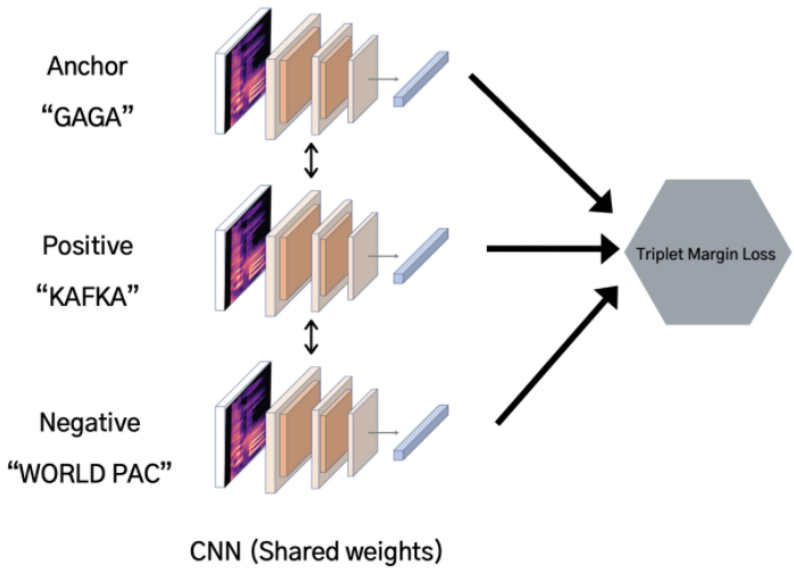

Figure 3.1: Visualization of Siamese Network structure

A typical method for extracting feature vectors in CNN is to obtain a global representation by passing through several convolution layers and Fully Connected Layers (FC Layers). In a general classification task, the dimension can be controlled relative easily by using the FC layer. However, in the FC layer, location-related information is lost because the Convolutional Feature Map including location information is spread out densely. In this paper, an embedding vector that well represents image and location information is needed, so Maximum Activations of Convolution (MAC) is used instead of the FC Layer.

MAC (Tolias et al., [11]) was used as a method of extracting feature vectors in CNN. MAC is one of the feature extraction methods frequently used in image retrieval or metric learning. For each feature map $X(C \times H \times W)$ of the last convolution layer, maximum value is extracted from each channel X_C and output a $1 \times C$

dimension feature vector. (Equation 3.3)

$$\text{MAC: } f = [f_1 ... f_c ... f_C]^T, f_c = \max_{x \in \mathcal{X}} x \tag{3.3}$$

In the proposed framework, the CNN pre-trained with the ImageNet dataset is used as the backbone network. Backbone network is a network that calculates the last convolution feature map before applying MAC operation. In this paper, ResNet [3], ResNeXt [19], EfficientNet [9] was used for the experiment. ResNet [3] is a neural network using skip connection which allows the gradient signal to skip layers while still propagating backward through the network. We used this to solve the gradient vanishing problem that occurs as the layer gets deeper. ResNeXt [19] is a network that improves performance by adding a cardinality dimension representing the size of a transformation set to the existing ResNet structure. EfficientNet [9] introduced a scaling coefficient that uniformly adjusts the number of channels, width, and image resolution of a model to explore an efficient model structure that provides the best performance relative to the size of the model. In this paper, the optimal model structure was searched by comparing the performance of three model structures used as backbone networks. In this paper, a database is built to evaluate the search performance. When constructing a database, first, a set of pre-registered trademark names are synthesized into speech, and then converted into log Mel spectrogram. The converted log Mel spectrogram is input to the trained model, output as a feature vector, and stored in the database.

When a new trademark name is applied for registration, it is converted into a log Mel spectrogram after undergoing speech synthesis. It is then fed to the same trained model used to build a database that stores feature vectors of pre-registered

trademark names.

After that, the cosine similarity between feature vector for the newly applied trademark name and the feature vector stored in the database is calculated, and the database is sorted in the order of the highest similarity. Finally, we use the top n trademark names in the sorted database as the search results.

3.2 Evaluation Metric

In this paper, accuracy and recall@K are used as evaluation metric to check the performance of the proposed framework.

Accuracy is used as a criterion to evaluate the test dataset performance. The test dataset consists of a set of two similar or dissimilar trademark names. Accuracy is defined as the ratio of correctly predicted labels to the total number of predicted labels (Equation 3.4). Accuracy is used to compare performance with baseline models such as Edit Distance [2], Ko et al. [5]

Recall@K is a metric that measures the ability of a model to return relevant results from K retrieved results (Equation 3.5). A true positive is 1 if there is a correct label among the top K retrieved results and 0 otherwise.

$$\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP=True Positive, FP=False Positive, TN=True Negative, FN=False Negative

(3.4)

$$\text{Recall@K} \ = \frac{1}{N} \sum_{i=1}^{N} \text{TP@K}_i$$
$$\text{where TP@K} = \begin{cases} 1 \text{ if there is a correct label among the top K retrieval results} \\ 0 \text{ otherwise} \end{cases} \tag{3.5}$$

Chapter 4

Datasets

4.1 Train dataset

The dataset used in this paper is the trademark text dataset provided by Korea Intellectual Property Rights Information Service (KIPRIS) and is divided into a training dataset and a validation dataset. It consists of the application number, trademark name in Korean or English, and specific reasons for rejection. In this paper, a total of 214,984 cases of rejection due to pronunciation similarity were used, 191,839 were the train dataset and 24,145 were the validation dataset. An example of a dataset is shown in Table 4.1.

Table 4.1: Example of dataset

"text1" represents the rejected trademark name and "text2"represents the pre-registered trademark name.

text1	text2	label
WORLD PARK	WORLD PAC	116
GAGA	KAFKA	117
천리안 캐피	천리안 모뎀	118
MYKIDS	MYKID	119
RPG 메이커	RPG400	120
...	...	...
두레초당	두레마을	234

4.2 Test dataset

Test dataset was also constructed to compare the results of previous studies. A previous study conducted by Ko et al. [5] tested based on the KIPRIS trademark dataset from 2010 to 2016 and consists of a total of 12,553 pronunciation similar cases and 34,020 dissimilar cases. We sampled 12,553 similar cases and 34,020 dissimilar cases from the validation dataset to construct a test dataset.

4.3 Speech dataset

Amazon Polly service was used to convert trademark text data into speech data. In the case of speaker, only the voice of the female Korean speaker "Seoyeon" was used to control the experimental conditions. The converted audio was saved in waveform audio file format (.wav)

4.4 Preprocessing

In this paper, log Mel spectrogram was used to convert audio files into values. log Mel spectrogram is a signal processing method using Mel filter-bank and logarithms to reflect human auditory characteristics.

The process of converting speech data into log Mel spectrogram is as follows. Since speech data is continuous, sampling is performed to convert it to discrete data. In this case, the sampling rate means the number of samples per second and the unit is Hz.

Since the sampled signal has time-dependent characteristics, the signal is divided into very short frame sections to satisfy the time stationary condition. This process

is called windowing. In this case, the hamming window function is used to correct the discontinuity of the section boundary, which is expressed by the following Equation 4.1.

$$\omega(n) = 0.54 - 0.46\cos(\frac{2n\pi}{N-1}) \tag{4.1}$$

The signal divided by frame is combined with waveforms of several frequencies. To obtain frequency information, the spectrum is obtained by dividing waveforms by frequency through Fourier transform. At this time, Fourier transform is performed for a very short section in order to maintain time information, which is called Short Time Fourier Transform (STFT). In addition, the spectrum extracted through STFT combined in chronological order is called a Spectrogram and log Mel Spectrogram is a value obtained by applying logarithm to Mel Spectrogram.

In this paper, two methods of preprocessing were used, either directly using the log Mel spectrogram value with zero padding or using the RGB image converted from the spectrogram. In the former case, the dimensions are unified using zero padding based on the maximum sequence length, and an output dimension of 1 X feature size X sequence length is calculated. In the latter case, the display function provided by the librosa package was used, and as a result of preprocessing, it has an output dimension of 3 x width x height. More detailed transformation hyperparameters are shown in Table 4.2.

Table 4.2: log Mel Spectrogram transformation hyperparameter

Hyperparameter	Value
sampling rate	22,050
hop length	1,024
n fft	2,048
n mels	128

Chapter 5

Experimental Results

In this paper, the same experimental environment was applied to all experiments. All experiments were performed using a Tesla V100 32GB, batch size of 64, initial learning rate of $5X10^{-5}$ and max epoch of 10. AdamW was used as the optimizer and Cosine Annealing was used as learning rate scheduler.

5.1 Experiment 1: Compare different input type

The first experiment compares the performance of the two preprocessing methods: log Mel spectrogram with zero padding, which uses zero padding method based on the length of the longest sequence and image of log Mel spectrogram, which is a method to convert log Mel spectrogram to RGB image.

First, in the case of log Mel spectrogram with zero padding, the longest sequence in the training dataset is 124, so log Mel spectrogram with a sequence length less than 124 are zero-padded until the sequence length is 124. In the case of log Mel spectrogram image, it was converted into an RGB image with a resolution of 224 X 224 using the display function of the Python library called Librosa.

The models trained in both methods are evaluated through the validation dataset. The performance was evaluated through Recall@K and ResNet50 was used as the

backbone networks.

As a result of the experiment, it was confirmed that the performance of the model using log Mel converted into RGB image as input was higher than the method using zero padding (Table 5.1).

Table 5.1: Performance comparison between different input type

Input type	Recall@1	Recall@2	Recall@4	Recall@8
raw log Mel with zero padding	0.31	0.44	0.55	0.68
log Mel image	**0.35**	**0.47**	**0.61**	**0.73**

.

5.2 Experiment 2: Compare signal processing methods

In the second experiment, the performance of the Mel function and the log function was compared using spectrogram, log spectrogram, Mel spectrogram, log Mel spectrogram.

As in Experiment 1, the search performance in the validation dataset was used as an evaluation metric, and ResNet50 was used as the backbone networks. In addition, a preprocessing method was used to convert the spectrogram into an RGB image of 224 X 224 resolution.

As a result of the experiment, it was confirmed that the signal processing method using the Mel function showed higher performance than the signal processing method that does not. The effect of log function was weaker than the effect of the Mel function, but it contributed to the performance improvement (Table 5.2).

Table 5.2: Performance comparison between four signal processing methods

Signal Processing Method	Recall@1	Recall@2	Recall@4	Recall@8
spectrogram	0.27	0.36	0.54	0.66
log spectrogram	0.29	0.38	0.55	0.67
Mel spectrogram	0.34	0.45	0.60	**0.73**
log Mel spectrogram	**0.35**	**0.47**	**0.61**	**0.73**

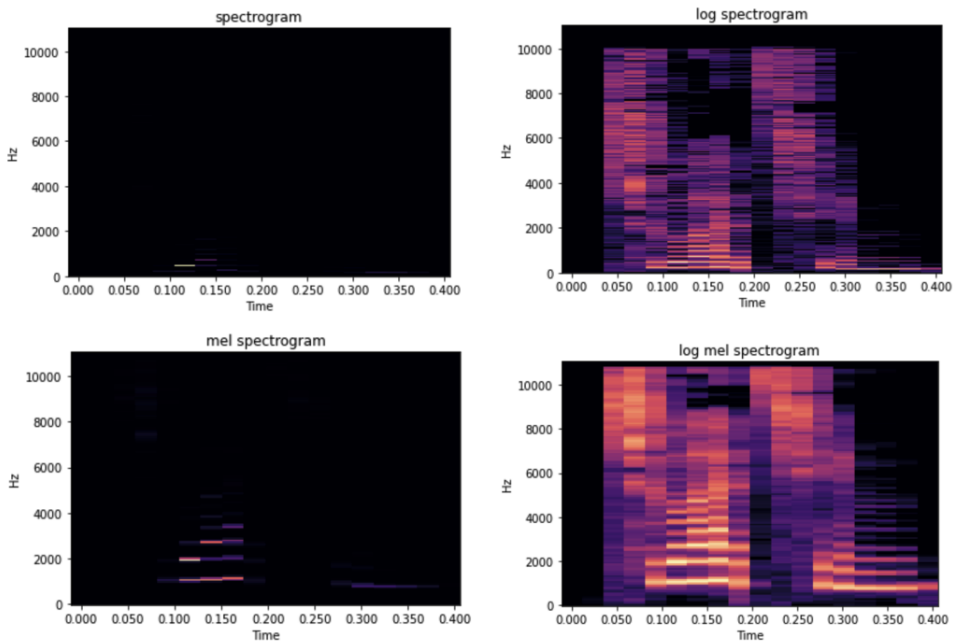

Figure 5.1: Comparison between different signal processing methods

5.3 Experiment 3: Compare backbone networks

In this paper, we compared the search performance according to the types of backbone networks. We used ResNet [3], ResNeXt [19], and EfficientNet [9] to compare neural network structure and ResNet50, ResNet101, ResNeXt50, ResNeXt101, EfficientNetb3, and EfficientNetb4 were used to compare backbone networks. RGB image data converted from log Mel spectrogram was used as input data, and search

performance was evaluated based on the validation dataset. As a result of the experiment, EfficientNet showed better performance than ResNet and ResNeXt (Table 5.3).

Table 5.3: Performance comparison between 6 backbone networks

Backbone	Recall@1	Recall@2	Recall@4	Recall@8
ResNet50	0.35	0.47	0.61	0.73
ResNet101	0.33	0.51	0.66	0.74
ResNeXt50	0.35	0.42	0.59	0.68
ResNeXt101	0.39	0.43	0.63	0.71
EfficientNetb3	0.41	**0.59**	**0.74**	**0.83**
EfficientNetb4	**0.44**	0.58	0.72	0.79

5.4 Experiment 4: Compare baseline models

To compare the performance with the baseline method, the performance was evaluated based on the test dataset. Edit distance [2], commonly known as Levenshtein distance and 2-gram Phonetic Feature Generation Using CNN (Ko et al. [5]), were used as baseline methods. In the case of Edit distance, the threshold was set to 5, and if the distance was less than the threshold, the label was set to 1 and if it was greater than the threshold, the label was set to 0. And in the case of the proposed framework, if the cosine similarity was greater than 0.4, it was set to 1, otherwise it was set to 0.

As a result of the experiment, proposed framework showed better performance than the baseline methods. Edit Distance showed the lowest accuracy because there is a clear limitation in judging the similarity of pronunciation only by spelling. Also, Ko et al. [5] significantly outperformed the Edit Distance, but performed worse than the proposed framework.

Table 5.4: Performance comparison between baseline methods

Model	Accuracy
Edit Distance [2]	79.61
Ko et al. [5]	92.7
ours (EfficientNetb3)	98.16
ours (EfficientNetb4)	**98.74**

Chapter 6

Conclusion

In this paper, we propose a deep learning framework that automatically determines the pronunciation similarity of trademark names using speech data converted using speech synthesis technology. The specific framework proposed in this paper is as follows. First, the trademark names are synthesized into speech data, and then the speech data is converted into RGB images of log Mel spectrogram. A Siamese network with an EfficientNetb4 backbone network receives the converted image and outputs a feature vector, which is learned through triplet margin loss.

The proposed framework showed 98.74% accuracy in the test dataset, which outperforms baseline methods such as Edit distance [2] and Ko et al. [5]. With the result, it can be assumed that the method which encodes the pronunciation directly produces a better result than the 2-gram Phonetic Feature Generation Using CNN proposed in Ko et al. [5].

In this study, we minimized the inconsistency between text and pronunciation by converting the trademark name into speech data, and we improved the accuracy of detecting similarity by using the converted speech data.

However, this study also has the following limitations.

First, the data used in this paper is limited to Korean text data, and speech

synthesis was performed using only the pronunciation of Korean speakers. Therefore, it is difficult to determine whether the proposed method is applicable to other languages. In the case of Korean, there is little discrepancy between pronunciation and spelling, and to evaluate its versatility, evaluation should be made based on foreign data and the voices of foreign speakers.

Second, there is a possibility that the speech synthesis method used in this paper is also inconsistent with the actual pronunciation. It is often difficult to pronounce a trademark name accurately only with text. Therefore, it is necessary to utilize speech data that is closer to actual pronunciation than text.

The following are suggestions for future research directions to improve the above limitations.

First, it is necessary to build an additional dataset based on different languages and pronunciation of various speakers, and train and evaluate the proposed model based on this. It is expected that this will ensure the versatility of the proposed model.

Second, it is necessary to construct data that is closer to actual pronunciation than text by using the International Pronunciation Sign (IPA) or speech data as it is. To synthesize international phonetic symbols into speech, an API can be used, and there is also a method of training a speech synthesis model using speech data labeled with international phonetic symbols.

The improved model is expected to be widely used in similar trademark name detection and search.

Bibliography

[1] F. M. Anuar, R. Setchi, and Y.-K. Lai, *Semantic retrieval of trademarks based on conceptual similarity*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 46 (2016), pp. 220–233.

[2] C. J. Fall and C. G. Giraud-Carrier, *Searching trademark databases for verbal similarities*, World Patent Information, 27 (2005), pp. 135–143.

[3] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015.

[4] K. Ito and L. Johnson, *The lj speech dataset.* `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[5] K. P. Ko, K. H. Lee, M. S. Jang, and G. H. Park, *2-gram-based phonetic feature generation for convolutional neural network in assessment of trademark similarity*, 2018.

[6] G. R. Koch, *Siamese neural networks for one-shot image recognition*, 2015.

[7] J. Moon, M.-J. Kim, S.-O. Lee, and Y. Yu, *A deep learning model based on triplet losses for a similar child drawing selection algorithm*, Journal of the Korea Industrial Information Systems Research, 27 (2022).

[8] F. Schroff, D. Kalenichenko, and J. Philbin, *FaceNet: A unified embedding for face recognition and clustering*, in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, jun 2015.

[9] M. Tan and Q. V. Le, *Efficientnet: Rethinking model scaling for convolutional neural networks*, (2019).

[10] X. Tan, J. Chen, H. Liu, J. Cong, C. Zhang, Y. Liu, X. Wang, Y. Leng, Y. Yi, L. He, F. Soong, T. Qin, S. Zhao, and T.-Y. Liu, *Naturalspeech: End-to-end text to speech synthesis with human-level quality*, (2022).

[11] G. Tolias, R. Sicre, and H. Jégou, *Particular object retrieval with integral max-pooling of cnn activations*, 2015.

[12] A. J. Trappey, C. V. Trappey, and E. Lin, *Intelligent trademark recognition and similarity analysis using a two-stage transfer learning approach*, Advanced Engineering Informatics, 52 (2022), p. 101567.

[13] C. V. Trappey, A. J. Trappey, and S. C.-C. Lin, *Intelligent trademark similarity analysis of image, spelling, and phonetic features using machine learning methodologies*, Advanced Engineering Informatics, 45 (2020), p. 101120.

[14] O. Tursun, C. Aker, and S. Kalkan, *A large-scale dataset and benchmark for similar trademark retrieval*, (2017).

[15] O. Tursun, S. Denman, S. Sivapalan, S. Sridharan, C. Fookes, and S. Mau, *Component-based attention for large-scale trademark retrieval*, IEEE Transactions on Information Forensics and Security, 17 (2022), pp. 2350–2363.

[16] O. Tursun, S. Denman, S. Sridharan, and C. Fookes, *Learning test-time augmentation for content-based image retrieval*, 2020.

[17] O. Tursun, S. Denman, S. Sridharan, and C. Fookes, *Learning regional attention over multi-resolution deep convolutional features for trademark retrieval*, in 2021 IEEE International Conference on Image Processing (ICIP), IEEE, sep 2021.

[18] W. Wang, X. Xu, J. Zhang, L. Yang, G. Song, and X. Huang, *Trademark image retrieval based on faster r-CNN*, Journal of Physics: Conference Series, 1237 (2019), p. 032042.

[19] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, *Aggregated residual transformations for deep neural networks*, 2016.

[20] N. Zeghidour, G. Synnaeve, N. Usunier, and E. Dupoux, *Joint learning of speaker and phonetic similarities with siamese networks*, in INTERSPEECH, 2016.

국문초록

최근 등록되는 상표의 수가 빠르게 증가함에 따라 기계학습을 기반으로 상표 유사성을 판단하려는 연구가 활발히 진행되어 왔다. 상표의 유사성은 도형, 관념, 발음을 기준으로 판단되는데, 발음의 경우 유사함의 기준이 모호하며 철자가 발음에 대응되지 않는 경우가 많기 때문에 유사성을 판단하는데 한계가 존재한다. 한편, 최근 음성 합성 기술의 발달로 인해 텍스트를 음성으로 변환하는 성능이 눈에 띄게 향상하였다. 본 논문은 음성합성기술을 활용하여 상표의 발음 유사성을 자동으로 판단하는 딥러닝 프레임워크를 제안한다. 먼저, 상표 텍스트를 음성으로 합성한 뒤, log Mel Spectrogram 으로 변환하고 합성곱 신경망과 삼중항 손실을 통해 feature 학습을 진행한다. 제안하는 방법과 선행 연구를 비교하기 위해 AIhub 에서 제공하는 상표 텍스트 데이터셋을 활용하였고, 제안하는 방식이 선행 연구를 앞서는 것을 확인하였다.

주요어: 합성곱 신경망, 샴 네트워크, Trademark Similarity, Trademark Pronunciation, 음성 합성

학번: 2019-23889

감사의 글

연구실 재학동안 제 의견을 전폭적으로 지지해주신 조성준 교수님께 감사의 말씀을 드립니다. 저와 함께 인공지능 세계를 탐구한 김주원, 안영훈 석사과정에게도 감사의 말씀을 전합니다. 서울대학교 산업공학과 데이터마이닝 연구실 모든 식구들께 감사드립니다.