Deep Learning Assisted Product Grouping for Shoe Size Recommendation

Eugene BULOG [1]*, Calli BATES [1], Naomi NORTH [2], Tsuyoshi IETA [2], Bo LI [1]
[1] ZOZO New Zealand Ltd., Auckland, New Zealand;
[2] ZOZO., Tokyo, Japan

https://doi.org/10.15221/22.63

Abstract

Shoe size recommendation tailored to specific products and users is a complex problem influenced by many factors. These include not only user-based attributes such as individual 3D foot shape and preferences, but also the sizing properties unique to each model of shoe. Large scale data collection and grouping of shoes based on the way they fit users is a crucial step towards being able to recommend to a user their perfect size in a specific item of footwear, down to the brand and product level.

This work presents a scalable and robust platform to facilitate AI-assisted grouping of footwear SKUs, allowing businesses to rapidly aggregate shoe products into groups containing similar items across multiple retailers with the exact same fitting properties, which can then be used to train a family of bespoke size recommendation models. These recommendation models use a combination of learned properties of each shoe and 3D foot scan data from users to predict a personalized ideal fitting size.

The platform leverages “human-in-the-loop” machine learning, by presenting highly accurate grouping predictions (generated by a deep learning triplet loss model) to human supervisors for quick confirmation. This provides a much faster alternative to humans combing an enormous list of products and manually cross checking each product against all existing groups.

Use of this platform has greatly accelerated the ability of our shoe size recommendation product (ZOZOMAT) to support new models of shoes - by automating the most time-intensive and error-prone aspect of grouping shoes for training and prediction. This results in more accurate and granular shoe size recommendations for users, and lower customer return rates in purchased shoes.

Keywords: shoe size prediction, recommendation, deep learning, metric learning, triplet loss, human-in-the-loop learning, 3d foot scanning, automation, data collection, data cleaning

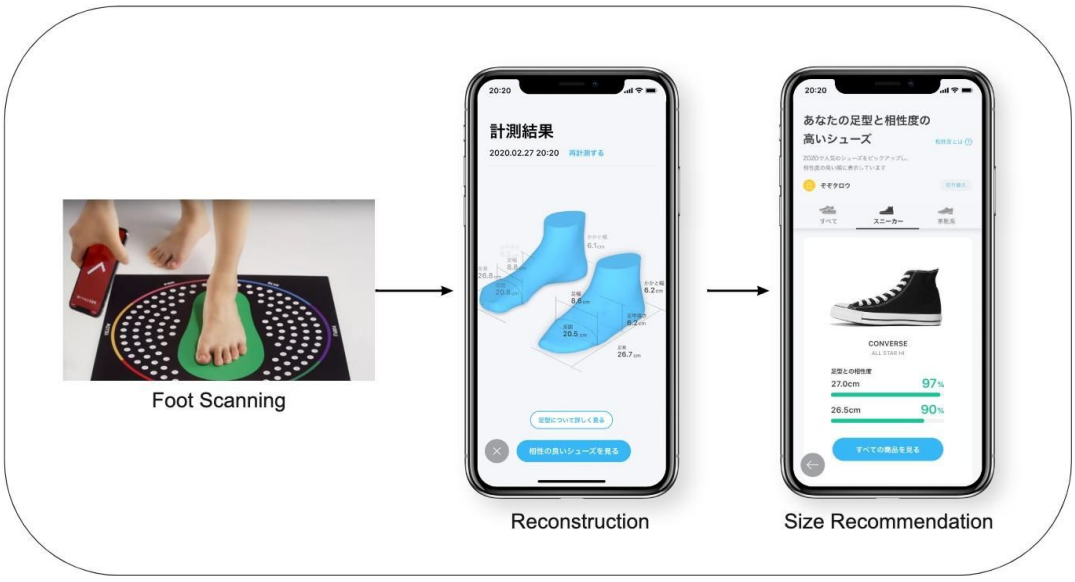

Fig. 1. The ZOZOMAT scanning and recommendation process.

* eugene.bulog@zozo.com, www.zozonz.com

1. Introduction

As an online apparel storefront, ZOZOTOWN has a keen focus on improving the ability of customers to buy clothing in their ideal size, without losing the convenience of wholly online shopping. This has led to the development of various software-based body measurement tools, including the ZOZOSUIT, ZOZOGLASS, and ZOZOMAT products. In particular, the ZOZOMAT aims to provide shoe size recommendations to users, in order to increase customer satisfaction, and reduce product return rates [1, 2].

The ZOZOMAT allows users to scan their 3D foot geometry, using their own smartphone camera (with the ZOZOMAT application installed) and a printed paper or plastic mat to aid in camera calibration and computer vision. Using a combination of the user's foot scans, and predictors trained on product information, including size reviews, measurements, and other data, the ZOZOMAT platform can provide sizing recommendations for each individual shoe supported by the platform, unique to that user's foot geometry. These recommendations are tailored not only to the user's foot shape, but also to the fitting properties of individual shoes, as ideal shoe size for an individual user can vary across different shoes. The scanning and recommendation process is illustrated in figure 1. A key goal of continued development on ZOZOMAT is increasing the number of supported shoes.

2. Grouping system platform

2.1. Product tag grouping

Due to the massive number (roughly 800,000) of individual footwear SKUs (referred to as "G_IDs") sold on the ZOZOTOWN store, training and maintaining separate fitting predictors for each item quickly becomes a scalability problem. Furthermore, due to ZOZOTOWN stocking products from a variety of suppliers and retailers, there is significant overlap between suppliers listing the same product lines under different G_IDs, as well as variants such as different colorways or visual patterns - all of which will have unique G_IDs but from the physical fitting perspective should be treated as the same item. A solution to these issues is to aggregate G_IDs into groups (referred to as ZOZO_tags), within which all items should have exactly the same physical fitting properties. This allows one fitting predictor per ZOZO_tag to be used across all variants of a product across different suppliers/retailers. Aggregating in this way also increases the amount of predictor training data available for training each ZOZO_tag specific fitting predictor.

Unfortunately, because each G_ID's details are entered on the supplier side, data such as the product name, brand name, price, etc. can vary drastically between G_IDs representing the same product, when supplied by different sources. This means that aggregating G_IDs into ZOZO_tag groups becomes a very manual process, requiring human interpretation of limited and noisy data in order to one-by-one assign incoming products to an existing ZOZO_tag (where possible). Generally, this process involved using a large, multi-page spreadsheet containing shoe data for all shoes sold on ZOZOTOWN. Staff would browse the spreadsheets to look at shoes one by one, inspect details such as their names, type of shoe, material, images, price, description and brand. They would then perform searches across the spreadsheet in an attempt to find matching shoes. If a match was found, this information was added into another spreadsheet. Due to the complexity of the task, this would take a long time to complete for even just a few shoes. As a result, grouping shoes into tag IDs was quickly identified as a bottleneck for rolling out support.

2.2. Human-in-the-loop suggested groupings

Due to the high amount of noise in the data used to determine tag groupings, fully automating the grouping task with a high level of confidence is challenging. This means that an ideal solution would incorporate some element of human supervision to ensure that generated groups are valid, while automating as much of the manual process as possible. To this end, we utilize the concept of a "human-in-the-loop" training and inference cycle.

Human-in-the-loop is a method of bootstrapping machine learning training, by exploiting the rich prior knowledge held by humans during the training process in order to overcome sparse or noisy ground truth data [3]. In the data labeling/cleaning case, this involves iteratively training a model on the available data, then using human annotations of the model predictions to retrain the model, until it is capable of accurately labeling unlabeled inputs. This process is much faster than humans manually labeling every single datapoint, as it instead only requires labeling the incorrect (or low confidence) model predictions, which will gradually reduce in quantity after every iteration.

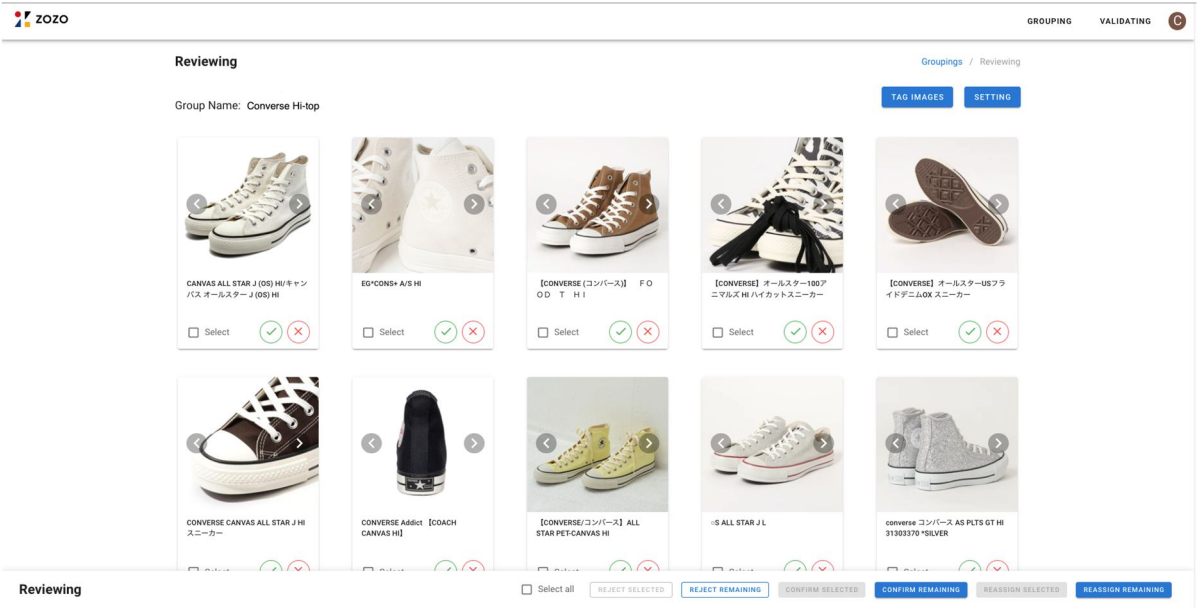

Fig. 2. The grouping suggestion review interface. Human reviewers are shown the deep learning predicted item groupings, and the name of the associated ZOZO_tag group, and can rapidly confirm or reject individual or bulk suggestions

Applying this concept, the grouping system uses a deep learning model to suggest the most likely ZOZO_tag grouping to each ungrouped G_ID, taking as input various sources including product images, metadata such as product name, brand name, pricing etc. These generated “tag suggestions” are then displayed to human supervisors via the Grouping System’s user interface, such that supervisors can quickly and efficiently accept or reject suggestions at a rapid pace. The decisions on whether suggestions are accepted or rejected are then fed back into the training process for the underlying deep learning model, creating a loop in which the model becomes more accurate over time as it receives feedback from human supervisors with continued use. This approach also has the benefit of massively reducing the complexity of decisions required by human annotators, from a large space (selecting an appropriate ZOZO_tag from thousands of options) to a simple binary decision (accept or reject the predicted group). Figure 2 illustrates the interface shown to human reviewers for validating grouping predictions.

3. Methods

3.1. Triplet loss

The deep learning model used to generate tag group suggestions is based on the “Triplet Loss” training technique [4]. Triplet loss is a form of deep metric learning [5] that aims to minimize the intra-class distance (defined by some learned abstract difference/similarity metric) and maximize the inter-class distance between samples. Triplet loss has been used for a wide range of tasks including facial recognition [6, 7], keyword spotting [8], and speaker verification [9]. This loss function excels at training models for tasks involving a large number of classes, with few examples per class - the opposite of the typical coarse classification problem, involving a handful of classes with many examples per class. An added advantage of triplet loss is its ability to label classes it hasn’t seen during training time, provided examples are given during inference.

Using triplet loss, we implemented and trained two separate embedding networks, one taking product images as input, and another taking textual and/or numerical product data as input. In both cases, the models output an n-dimensional embedding vector for each G_ID, normalized such that all embeddings occur on the surface of an n-dimensional unit hypersphere. In order to suggest a ZOZO_tag for an incoming untagged G_ID, we can calculate its embedding, and simply find the nearest neighbor which has a known ZOZO_tag (using a metric such as cosine similarity), and suggest that the incoming G_ID belongs to this same ZOZO_tag. Due to the metric-learning nature of triplet loss, we can also list the top m most likely ZOZO_tags in order of most to least likely, for any arbitrary m. Both networks used the Selectively Contrastive Triplet loss variant [10], combined with online batch mining [6].

3.2. Image-based suggestions

For image-based tag suggestions, we used a standard ResNet34 architecture [11] (pretrained on the ImageNet task [12]), modified to output a 512-dimensional vector normalized to a magnitude of 1.0. This model was then trained using triplet loss, wherein each triplet "class" consisted of images belonging to the same ZOZO_tag. Each training minibatch was composed of up to 12 (where available) images from each of 10 randomly selected ZOZO_tags, meaning each minibatch was composed of ≤ 120 224x224 RGB images overall.

An issue encountered with image-based suggestions is the extremely noisy nature of storefront product images. An ideal image contains the footwear item centered in the frame, completely visible, on a plain background, and from an informative camera angle (e.g. not an image of the shoe's sole). Training with all available images resulted in a low accuracy due to many images not being informative enough by these standards. To this end, we trained a pair of simple multi-label classifiers to classify images into a camera angle class (e.g. side, front, rear, top, bottom, unclear) and an "accept/reject" class (rejecting images where the shoe is too small in the frame, e.g. being worn by a model such that the shoe only occupies 10% of the image). Using these classifiers to clean the image data, as well as image augmentation during training resulted in greatly improved performance in the "sneaker" category of shoes.

3.3. Text-based suggestions

While the above method performed well on shoes in the "sneaker" category, other categories such as "pumps" or "dress shoes" did not perform very well with the image model. A possible reason for this is that shoes within these categories are much less visually distinct than sneakers, which often have a wide variation in styles and colors, whereas categories more suited for formal wear are more likely to feature muted colors and visually subtle variations in style, making it difficult to distinguish examples purely from one or two images to the high degree of granularity required for this task. As a solution to this problem, we developed a text-based model as an alternative for these categories.

While a standard solution for natural language processing tasks in deep learning is to leverage pre-trained word embeddings, we faced a unique problem in that the textual data available to this model (product name and brand name) could contain mixtures of English and Japanese characters, misspellings, brand/product names that were wholly original (i.e. wouldn't exist in a pre-trained vocabulary), and terms without specific semantic meaning (such as product names that are simply a unique model number). To solve this, we opted to build word embeddings from scratch, beginning at the per-character level to learn a unique vocabulary suited to our specific data domain.

This was accomplished by creating a character vocabulary of English letters and numbers, and a set of common Japanese characters, where each character in the vocabulary maps to a unique learned embedding vector. Because the set of all possible Japanese characters is very large, training a well-behaved embedding for every single one is infeasible (especially for those which don't appear in the training data). To this end, we also create an "overflow" index mapping for characters outside of the trained range, into which untrained characters will be hash-mapped, to allow a pseudo-unique embedding for unseen characters. While these overflow characters won't have an embedding trained based on their semantic meaning, they still map to unique embeddings, which contributes to distinguishing a string containing them from a similar string containing a different unseen character.

The overall text-based model first tokenizes each input (product name and brand name) by splitting them into sequences of words, and then splits each word into ordered sequences of individual characters. The characters are mapped into their embedding vectors via the vocabulary process mentioned above, and then fed into a recurrent neural net (composed of LSTM layers [13]) to build up a word-level embedding out of each ordered sequence of character-level embeddings. The word-level embeddings are then aggregated into a per-element mean, min, and max, across the entire input (e.g. item name or brand name), to create a "summary" vector at the "sentence" level. The sentence-level embeddings do not take into account the order of individual words within the "sentence", as we found that in product names this varies significantly, and does not provide any exploitable information - for example, one G_ID may have the product name in the format of "[brand] [model] [color]", while another G_ID in the same tag group may be in the format of "[color] [model] [brand]", which should both result in the same or similar embeddings as they refer to the same overall product. The "sentence"-level embeddings for product name and brand name (and those of any additional textual/numerical inputs) are then concatenated and passed to several point-wise feed forward layers, followed by a normalization, outputting the final embedding vector representing the G_ID as a whole. An overview of this design is shown in figure 3.

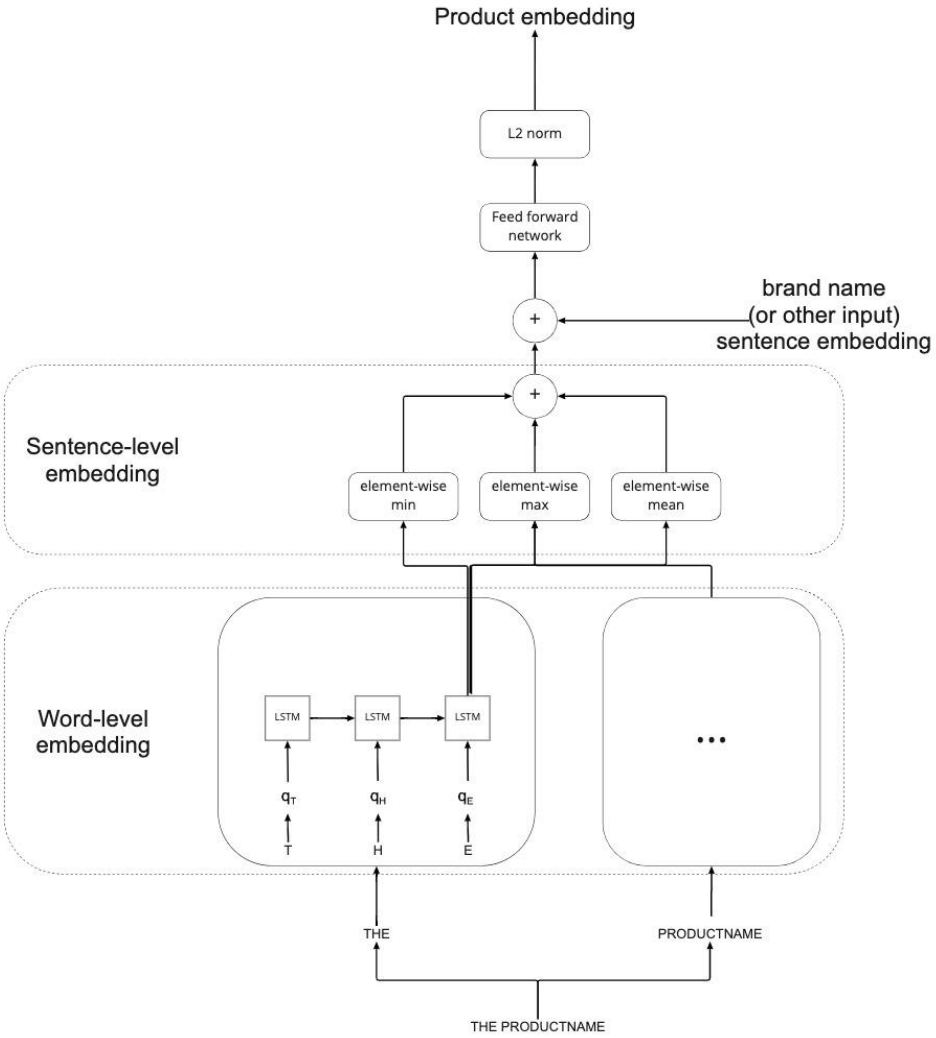

Fig. 3. The architecture of the textual embedding model.

While this per-character approach means the semantic meaning of words is heavily dependent on the "meanings" of individual characters, and therefore wouldn't be suitable for usage on tasks in which grammar is important, we found that it performs very well in this task, as it essentially functions as an advanced form of a string similarity search, which is tolerant to misspellings and mixtures of English and Japanese, and can recognize different terms with similar meanings, within this domain. This text-based model also overcomes the limitations of the image-based model in that it yields high accuracy across all tested categories of footwear, not just "sneakers".

4. Results

Accuracy experiments were performed using a 90-10 train-test split on items with known ZOZO_tags (split via ZOZO_tag, to ensure no overlap in ZOZO_tag between training and testing data). The training set was composed of 1748 ZOZO_tags, containing a total of 35114 G_IDs, and 108771 images (after filtering). The testing set was composed of 195 ZOZO_tags, containing a total of 8050 G_IDs, and 24525 images (after filtering). Note that for the image model, each individual image is a single datapoint (as G_IDs typically have multiple different images), whereas in the case of the text model, each G_ID itself is a single datapoint.

Testing was performed by running model training on the training set ZOZO_tags, then running inference on the test set datapoints, and counting the proportion of test set query points whose nearest neighbor was in the same ZOZO_tag group as the query point. The results of these experiments are shown in Table 1.

Table 1. Experimental accuracy results.

Model Type	Accuracy rate (sneakers)	Accuracy rate (dress shoes)	Accuracy rate (pumps)
Image model (no filtering or image augmentation	54.2%	-	-
Image model (with filtering, image augmentation)	92.4%	68.7%	70.9%
Text model	95.8%	93.2%	94.4%

As mentioned previously, filtering images to remove uninformative examples, and utilizing image augmentation (primarily color, rotation, and scale transformations) significantly improves accuracy of the image model on the sneaker set. However, performance on the dress shoes and pumps categories was still poor by comparison. We speculate that this is due to the smaller amount of visual variation in these two categories compared to sneakers. The text model performed well across all three categories. Other categories on ZOZOTOWN were not tested due to lack of training data.

A loose time comparison was also performed between the previous manual spreadsheet based system and the deep learning suggestion based system on the grouping platform. Due to the variable nature of how complex determining the correct tag group for a given item is, an exact comparison is difficult, however it was found that on average, grouping items using the deep learning suggestion based system took between 1 to 3 minutes per G_ID (depending on accuracy), compared to the manual system which could take from 1 minute up to 10-15 minutes per G_ID. Note that this comparison is intended to only give a rough indication of speed improvements, as it varies significantly based on the experience of the human performing reviewing, and their familiarity with the brands and products being grouped. It is expected that use of the Grouping System will further speed up as the human reviewers become more comfortable with the process, and as the algorithm continues to improve in accuracy due to the “human-in-the-loop” cycle.

5. Conclusion, Limitations, and Future Work

This work introduces a platform for quickly and efficiently grouping similar products together via a human-in-the-loop platform, powered by deep triplet loss grouping suggestions based on product images or textual data. This suggestion-based approach and the overarching centralized platform has greatly improved the speed and efficiency of grouping shoe products together for better training of the ZOZOMAT size prediction tool, allowing a wider range of supported products, in turn leading to a reduced product return rate.

The grouping suggestion model can also be applied to other areas, including an advanced store search interface, product recommendation algorithms, and other forms of data cleaning, such as grouping products based on their manufacturer when the “brand name” field is noisy or erroneous.

There are three main areas of this current approach which could benefit from further research. Firstly, due to the nature of triplet loss, predicting a measure of confidence on a suggested grouping is difficult, as the similarity metric learned can vary in acceptable range between different clusters - i.e. the algorithm is good at “ranking” products in order of similarity to a query, but not as good at predicting whether the “most similar” product is actually of the same type. Secondly, the possibility of clustering ungrouped items to create *new* ZOZO_tags (as opposed to only assigning incoming items to existing groups) requires further investigation. Finally, a cross-domain attention-based model [14], which exploits information from both the image and textual domains simultaneously (instead of the current two separate models) would likely improve the accuracy and robustness of the model to incomplete data in either the image or product text inputs. We are continuing to work on these areas, and hope that future work will yield further improvements.

References

[1] ZOZO Inc, “ZOZOMAT,” https://zozo.jp/zozomat/ [Accessed: 15- Aug- 2022].

[2] B. Li, "ZOZOSUIT & ZOZOMAT: Solutions for Online Shopping in Japan", Proc. of 3DBODY.TECH 2020 - 11th Int. Conf. and Exh. on 3D Body Scanning and Processing Technologies, Online/Virtual, 17-18 Nov. 2020, #47

[3] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He, “A survey of human-in-the-loop for machine learning,” Future Generation Computer Systems, 2022, https://doi.org/10.1016/j.future.2022.05.014.

[4] J. Wang et al., “Learning fine-grained image similarity with deep ranking,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1386–1393, https://doi.org/10.48550/arXiv.1404.4661.

[5] J. Lu, J. Hu, and J. Zhou, “Deep metric learning for visual understanding: An overview of recent advances,” IEEE Signal Processing Magazine, vol. 34, no. 6, pp. 76–84, 2017, https://doi.org/10.1109/MSP.2017.2732900.

[6] B. Amos, B. Ludwiczuk, M. Satyanarayanan, and others, “Openface: A general-purpose face recognition library with mobile applications,” CMU School of Computer Science, vol. 6, no. 2, p. 20, 2016.

[7] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823, https://doi.org/10.48550/arXiv.1503.03832.

[8] R. Vygon and N. Mikhaylovskiy, “Learning Efficient Representations for Keyword Spotting with Triplet Loss,” in Speech and Computer, 2021, pp. 773–785, https://doi.org/10.1007/978-3-030-87802-3_69.

[9] Z. Ren, Z. Chen, and S. Xu, “Triplet based embedding distance and similarity learning for text-independent speaker verification,” in 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2019, pp. 558–562, https://doi.org/10.48550/arXiv.1908.02283.

[10] H. Xuan, A. Stylianou, X. Liu, and R. Pless, “Hard negative examples are hard, but useful,” in European Conference on Computer Vision, 2020, pp. 126–142, https://doi.org/10.48550/arXiv.2007.12749.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778, https://doi.org/10.48550/arXiv.1512.03385.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255, https://doi.org/10.1109/CVPR.2009.5206848.

[13] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997, https://doi.org/10.1162/neco.1997.9.8.1735.

[14] A. Vaswani et al., “Attention is all you need,” Advances in neural information processing systems, vol. 30, 2017, https://doi.org/10.48550/arXiv.1706.03762.