Handshape Classification in a Reverse Dictionary of Sign Languages for the Deaf

Alikhan Abutalipov[1], Aigerim Janaliyeva[1], Medet Mukushev[2], Antonio Cerone[1], and Anara Sandygulova[2](✉)

[1] Department of Computer Science,
Nazarbayev University, Nur-Sultan, Kazakhstan
{alikhan.abutalipov,aigerim.janaliyeva,antonio.cerone}@nu.edu.kz

[2] Department of Robotics and Mechatronics,
Nazarbayev University, Nur-Sultan, Kazakhstan
{mmukushev,anara.sandygulova}@nu.edu.kz

Abstract. This paper showcases the work that aims at building a user-friendly mobile application of a reverse dictionary to translate sign languages to spoken languages. The concept behind the reverse dictionary is the ability to perform a video-based search by demonstrating a handshape in front of a mobile phone's camera. The user would be able to use this feature in two ways. Firstly, the user would be able to search for a word by showing a handshape for the application to provide a list of signs that contain that handshape. Secondly, the user could fingerspell the word letter by letter in front of the camera for the application to return the sign that corresponds to that word. The user can then look through the suggested videos and see their written translations. To offer other functionalities, the application also has Search by Category and Search by Word options. Currently, the reverse dictionary supports translations from Russian Sign Language (RSL) to Russian language.

Keywords: Reverse dictionary · Sign language dictionary · Fingerspelling recognition · Video-based search interface · Human-computer interaction · iOS application · Russian Sign Language (RSL)

1 Introduction

Deaf communities around the world use sign languages for everyday communication. Each country or region has its own sign language. Contrary to popular belief, Russian Sign Language (RSL) does not share structure or grammar with the Russian language. In addition, people native to RSL do not necessarily know how to read and write Russian and have to learn it as a foreign language.

Most online sign language (SL) dictionaries are alphabet-based which are convenient for people who are fluent in spoken languages. When searching for a sign, they need to know the written translation of it and search by its first

J. Bowles et al. (Eds.): DataMod 2020, LNCS 12611, pp. 217–226, 2021.
https://doi.org/10.1007/978-3-030-70650-0_14

letter. However, such functionality is useful for people who want to learn SL and cannot provide a reverse option - searching for meaning of unfamiliar signs.

There exists only a few reserve dictionaries where searching by sign is performed by one of its components, such as handshapes. Nonetheless, this is still not user-friendly as each handshape is described in a written form. Usually these descriptions are compiled by professional SL linguists, which makes it hard for a non-expert user to understand the description. Sometimes the pictorial representations of the handshapes are provided too, but then the creation of such dictionaries for every sign is time-consuming.

Therefore, this work aims to build an automatic reverse dictionary where a search is performed in the most natural way - searching by demonstration. Since each sign in a sign language consists of one or several handshapes, searching by handshape demonstration would yield the most intuitive method for people native to sign languages.

2 Related Work

2.1 Sign Language Dictionaries

Computer-based sign language dictionaries could be divided into two categories: search by textual description of the sign and demonstration of the sign.

Search by demonstration systems became popular with the introduction of Microsoft Kinect, which supports skeletal joints tracking [6]. Another approach is the use of systems that accept video demonstration of the sign as an input to find the list of similar signs [4,20]. However, such systems may have poor performance when tested on different users and users are required to perfectly demonstrate the sign in order to find its match.

Feature-based sign language dictionaries overcome the problems of computer-based sign language dictionaries by focusing only on features or components describing a sign. Bragg *et al.* [3] proposed a feature-based dictionary system that enables users to lookup unknown signs by selecting from features such as handshape, orientation, or location. Increase in the computational power of smartphones opens new opportunities for the development of sign language dictionaries. Some functional prototypes were built both for text-to-sign [8] or handshape-to-sign systems [14]. Alonzo *et al.* [1] highlight the difficulty of searching for an unfamiliar sign in dictionaries. Furthermore, they showed that the placement of the searched sign in the list (its position) and the similarity of the shown items affect user's opinion regarding the quality of the search results. In general, researchers agree that there are few resources available that are robust enough to overcome all existing limitations [2].

2.2 Sign Language Fingerspelling Recognition

Automatic sign language recognition has been an active field of research in the past couple of decades and fingerspelling recognition is one part of sign language recognition. Fingerspelling is used to express words that have no specific sign in the vocabulary of sign languages. Many approaches were used to solve this task such as hand crafted features, Convolutional Neural Networks, and depth features. The field largely benefits from the advances in computer vision.

The Australian Sign Language fingerspelling recognizer uses a combination of features extracted from skin detection which are later used to extract geometric features. For classification it applies Hidden Markov Models (HMM) to get the output probabilities for the given sequence of features. At word level this model achieves 88.61% recognition accuracy [7]. For the American Sign Language (ASL) a semi-Markov conditional model approach was developed. It achieves an 11.6% letter error rate compared to the HMM baseline with 16.3% [9].

Microsoft Kinect depth cameras showed good results when applied to fingerspelling recognition. Pugeault and Bowden [16] proposed a real-time ASL fingerspelling recognition system based on Microsoft Kinect. Their approach focuses on detecting user's hands and extracting handshape features and is based on Gabor filtering of the intensity and depth images. The classification part was performed with multi-class random forest and achieved 75% accuracy. Dong *et al.* [5] proposed a model for recognizing 24 static ASL alphabet signs with 90% accuracy. Their model first extracted hand segments based on depth contrast features which were then used to localize hand joint positions. For the classification part, a Random Forest algorithm was applied. Another interesting approach based on classification tree and machine learning was developed for the Japanese Sign Language. It supports the classification of 41 characters without movement with 86% accuracy [11]. Point cloud descriptors recorded with Microsoft Kinect were used to recognize static letters of the Polish Sign Language. The classification part was performed using HMM and achieved an accuracy of 78.8% [21].

Some specific hardware is required for the systems mentioned above which are not convenient for the end users. In contrast, vision-based approaches can be implemented using only web-cameras or mobile phone cameras. Shi et al. [17] introduced the largest dataset for ASL fingerspelling recognition used to detect fingerspelling "in the wild" in realistic conditions. Most of the previous works performed experiments on more controlled data with a limited number of participants. The proposed system has two parts: hand detector and sequence recognizer. The best letter accuracy was achieved with a CTC-based recognizer and was around 42%. Another approach for detecting fingerspelling in realistic conditions used end-to-end model with an iterative attention mechanism. In contrast to previous work, this approach is not using explicit hand detection or segmentation. The best accuracy was 61.2% on ChicagoFSWild dataset [18].

3 System Design and Architecture

Our system is based on a database where each sign video has a list of handshapes corresponding to that sign. We used a publicly available dictionary of RSL from the Spread the Sign dictionary[1]. Thus, every frame of the sign video

[1] www.spreadthesign.com.

was cropped to contain only the hand region using the “Hand-CNN” pre-trained hand detection model [13]. Then we utilized the “Deep Hand” pre-trained handshape recognition model [10] to classify handshapes in each sign video. Once the database was ready, we built a system consisting of two main components, an iOS mobile application and a server that runs the “Hand-CNN” and “Deep Hand” models. When a user takes a photo of a handshape, the application sends the image over HTTP as a request to the server, which in turn classifies the handshape in the photo and return the result to the application via an HTTP request. The application then shows the user the signs that contain the user’s handshape by searching the database. When a user takes a photo of another handshape, the just-described process repeats itself, but this time the application shows the signs that contain both handshapes. The more handshapes are shown, the narrower the search is. Overall architecture of the application is presented in Fig. 1.

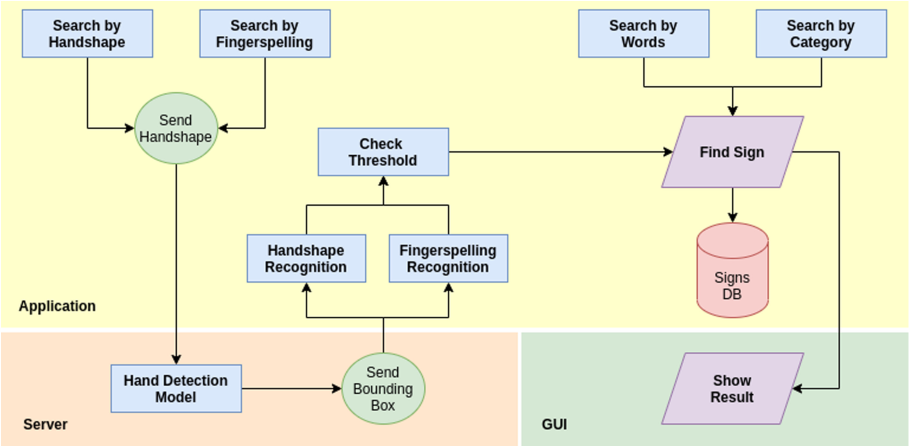


Fig. 1. System components

3.1 Datasets

In order to adapt the handshape classification to support RSL and fingerspelling in RSL, we utilized a manually labeled dataset of RSL handshapes [12] as well as a previously collected Cyrillic fingerspelling dataset [19] to perform transfer learning of the “Deep Hand” model to make two models for RSL handshape recognition and Cyrillic fingerspelling. In the end, the number of classes was 29 for 33 letters in the Russian alphabet as some cases were combined due to being different only in the movement. This is the case for the signs for the letters И, Й, Ш, Щ.

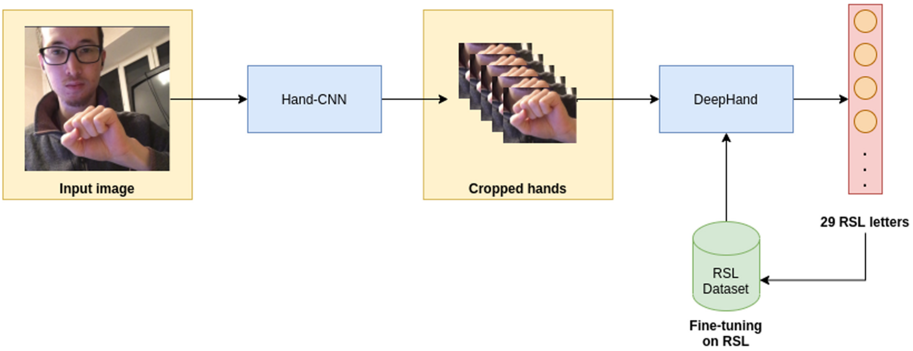


Fig. 2. Training process

3.2 Implementation Details

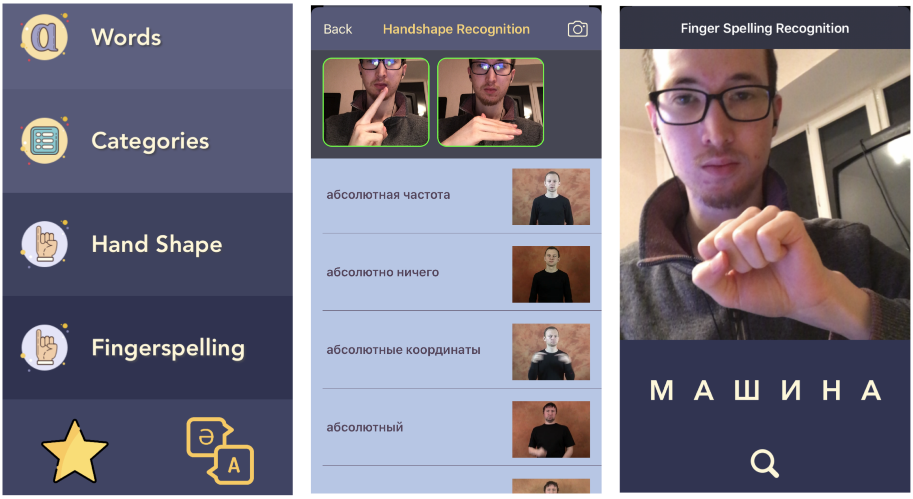
For the transfer learning, we decreased the overall learning rate from 0.0005 to 0.0002 while increasing the learning rate of the final layer by 2 and used the RSL datasets [12,19] to re-train the pre-trained "Deep Hand" model's weights. The results were: Top-1 results refer to the output deemed most probable by a model, while Top-5 results refer to the 5 most probable outputs of models. The reason for transfer learning was two-folds: first, the "Deep Hand" model already showed rather good results on their dataset: 85% for Top-1 results and 94.8% for Top-5 results [10] (see Table 1). It was beneficial to use the model's "knowledge". Secondly, the size of our datasets used for transfer learning was much smaller than the dataset used to train "Deep Hand" in [10]: 3201 images for 36 classes of the Handshapes model and 1587 images for the Fingerspelling model versus over 1 million handshape images in [10]. The training process is shown in Fig. 2.

Table 1. Transfer-learned models' accuracy

Model	Number of classes	Top-1 accuracy [%]	Top-5 accuracy [%]
Deep Hand [10]	45	85	94.8
Fingerspelling	29	88	97
Handshapes	35	74	94.6

4 System Functionality

The application is a reverse dictionary that supports Russian and Russian Sign Language. It has the following components: "Search by Words", "Search by Categories", "Search by Fingerspelling" and "Search by Handshapes", which are described below in more details. Its screenshots of various views are presented in Fig. 3 and 4.



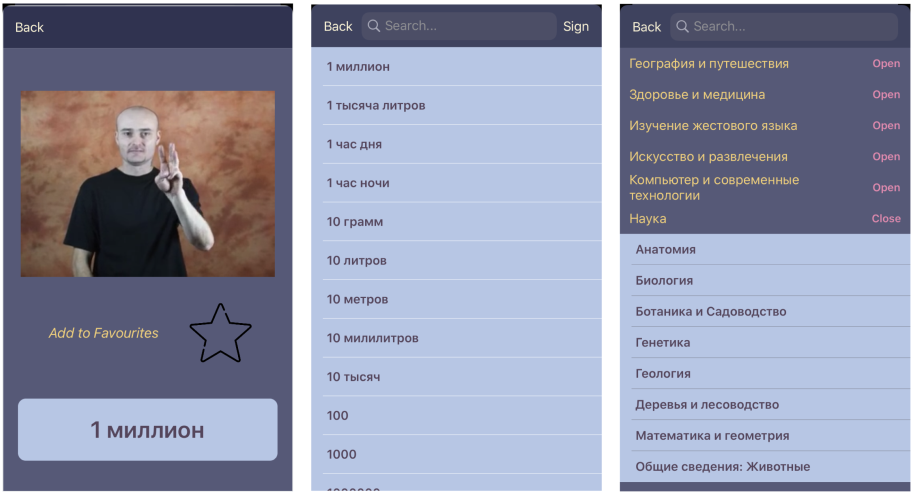
(a) Home view (b) Search by Handshapes (c) Search by Fingerspelling

Fig. 3. Application views

4.1 Search by Handshapes

The main functionality of the application is the ability to search for signs by the handshapes that are used to form them. The “Handshape” option from the home view launches the camera view for the user to take a photo of their hand. After taking a photo the “Search by Handshape” view is shown, where a top one-third part of the view shows the photos of the handshapes that the user uses to search for a sign. The rest of the view shows the list of signs that contain the user-provided handshapes. The signs are shown as videos in the loop. We assume that because deaf people are proficient in recognizing signs, they will not be confused by simultaneously playing videos of different signs.

The taking photo is on the top part of the view. If the application successfully classifies the handshape, the border around the photo of the handshape turns green. However, if the handshape is not classified or the application cannot reach the server, the image disappears. The user can also add other handshapes. To do so, the user taps on the “camera” button in the top right corner of the view, which presents the camera view, where the user can take a photo of another handshape. Moreover, the user can delete a handshape from the search by long pressing on the photo of the handshape and tapping on the “delete” button that will be shown as the result. The list of signs updates every time a new handshape is added or an existing one is deleted to reflect the most current state of the search. Finally, the user can tap on a sign, which will result in the “Sign” view to be shown.



(a) Sign view (b) Search by Words (c) Search by Category

Fig. 4. Application views

4.2 Search by Fingerspelling

Another important feature of the application is "Search by Fingerspelling". Similarly to the "Search by Handshapes" it sends the handshape image shown during fingerspelling to the server, which returns back the bounding boxes that bound the hands in the image. The application classifies the image using the locally run "Fingerspelling" model. Here, however, the distinction between these two features is evident. The application does not search for signs immediately, but rather sends another image of handshape to the server and waits for the bounding boxes coordinates. It does so for a few dozen images, after which it checks whether there is a particular sign that corresponds to a minimum 80% percent of classified images of handshapes. If so, the application builds a word by adding the letter that is represented by the handshape that reached the 80% threshold. If the threshold is not met, the application discards the oldest frame, sends the latest frame to the server again and tests for the threshold again. After the word is built, the application sends a query to the database, fetches signs that relate to the built word and shows the result to the user.

4.3 Other Search Methods

In the "Search by Words" the user sees the list of all words and phrases that the application has in its vocabulary. Users can use the search bar at the top of the view to search for the word or phrase that they want the sign translation for. After the user taps on a specific word or phrase, the video of the signs that correspond to the selected word or phrase is shown in a loop. In addition to that, the word or phrase is shown at the bottom of the screen. Moreover, the user can tap on the star image to mark or unmark the sign as favorite. All favorite signs can be accessed quickly by clicking on the "Favorites" option in the home view. This method of searching will be mostly useful for the people who are learning the sign language. However, deaf people might also find this method useful, as it would allow them to translate unknown Russian words that they encounter.

In the home view, when the user taps on the "Category" option, "Search by Category" has multiple categories, which are presented in a way similar to the "Search by Words" view. By tapping a word or phrase in this list, the list of sign videos are presented.

5 Evaluation

We conducted a user study with two non-deaf people and conducted interviews with 6 non-deaf people after showing them a video of the application. After the result of the user study and interviews we come up with three broad comments. First of all, the non-deaf users might want to have additional capabilities in the app to help them in the learning of the Kazakh/Russian sign language. The application might have a list of handshapes that are used in the Kazakh/Russian sign language and in fingerspelling. Further, the app might have educational parts, where the user could learn by practicing fingerspelling and sign words/phrases. Second, the "Search by Categories" was deemed the least useful feature of the app by both groups. Thirdly, the app might need to include a tutorial of how to use the app. The tutorial might consist of a video that demonstrates and explains all the features when the app is first launched.

There were some other useful points from the user study. One participant said that looking for a word in "Search by Categories" was not very easy. In response, we included the A-Z index list shown in Fig. 4. Users can slide their fingers along the index list, and the app would "snap" or move to the words that begin with a particular letter in the index list. In addition, during the user study, searching by handshapes took a long time for both participants, in the range from 1 min to almost 2 min. The fingerspelling model gave reasonable results, for example, when the users were instructed to fingerspell the word "ЗАЯЦ", although the timing was not great, about 2 min for both participants. When the participants were instructed to fingerspell the word "НОЗДРЯ" the model could not recognize the handshape for the letter "Р" for both users.

6 Conclusion and Future Work

In this work we presented the prototype of an automatic reverse dictionary based on the video-based handshape configuration search. Handshape is the basic component of a sign. Thus, searching by demonstration provides the most natural way for the users. We also presented how transfer learning could be applied to sign language recognition. As most of the sign languages are considered as low-resource languages, such approach could be beneficial when annotated data is limited.

Future work will include training a hand detection model compatible with Core ML. This will allow us to run all models on the device and will allow the users to use the application without the need to be connected to the internet. Moreover, a user study with deaf people should be conducted and the feedback should be incorporated in the future version of the application. We plan to conduct a two-step usability study. Firstly, we plan to conduct a pilot study with approximately five hearing users and later conduct a usability study with approximately five deaf users. According to Nielsen and Landauer [15] five users should be enough to find most of the usability issues during the testing.

Acknowledgment. This work was supported by the Nazarbayev University Faculty Development Competitive Research Grant Program 2019–2021 "Kazakh Sign Language Automatic Recognition System (K-SLARS)". Award number is 110119FD4545.

References

1. Alonzo, O., Glasser, A., Huenerfauth, M.: Effect of automatic sign recognition performance on the usability of video-based search interfaces for sign language dictionaries. In: The 21st International ACM SIGACCESS Conference on Computers and Accessibility, pp. 56–67 (2019)
2. Bragg, D., et al.: Sign language recognition, generation, and translation: an interdisciplinary perspective. In The 21st International ACM SIGACCESS Conference on Computers and Accessibility, pp. 16–31 (2019)
3. Bragg, D., Rector, K., Ladner, R.E.: A user-powered American Sign Language dictionary. In: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, pp. 1837–1848 (2015)
4. Cooper, H., Pugeault, N., Bowden, R.: Reading the signs: a video based sign dictionary. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 914–919. IEEE (2011)
5. Dong, C., Leu, M.C., Yin, Z.: American sign language alphabet recognition using Microsoft Kinect. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 44–52 (2015)
6. Elliott, R., Cooper, H., Ong, E.-J., Glauert, J., Bowden, R., Lefebvre-Albaret, F.: Search-by-example in multilingual sign language databases. In: Proceedings of the Sign Language Translation and Avatar Technologies Workshops (2011)
7. Goh, P., Holden, E.-J.: Dynamic fingerspelling recognition using geometric and motion features. In: 2006 International Conference on Image Processing, pp. 2741–2744. IEEE (2006)
8. Jones, M.D., Hamilton, H., Petmecky, J.: Mobile phone access to a sign language dictionary. In: Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility, pp. 331–332 (2015)
9. Kim, T., Shakhnarovich, G., Livescu, K.: Fingerspelling recognition with semi-Markov conditional random fields. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1521–1528 (2013)
10. Koller, O., Ney, H., Bowden, R.: Deep hand: how to train a CNN on 1 million hand images when your data is continuous and weakly labelled. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3793–3802 (2016)
11. Mukai, N., Harada, N., Chang, Y.: Japanese fingerspelling recognition based on classification tree and machine learning. In: 2017 NICOGRAPH International (NICOInt), pp. 19–24. IEEE (2017)

12. Mukushev, M., Imashev, A., Kimmelman, V., Sandygulova, A.: Automatic classification of handshapes in Russian Sign Language. In: Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives, Marseille, France, pp. 165–170. European Language Resources Association (ELRA), May 2020
13. Narasimhaswamy, S., Wei, Z., Wang, Y., Zhang, J., Hoai, M.: Contextual attention for hand detection in the wild (2019)
14. Nelson, A., Price, K., Multari, R.: ASL reverse dictionary-ASL translation using deep learning. SMU Data Sci. Rev. **2**(1), 21 (2019)
15. Nielsen, J., Landauer, T.K.: A mathematical model of the finding of usability problems. In: Proceedings of the INTERACT 1993 and CHI 1993 Conference on Human Factors in Computing Systems, CHI 1993, New York, NY, USA, pp. 206–213. Association for Computing Machinery (1993)
16. Pugeault, N., Bowden, R.: Spelling it out: real-time ASL fingerspelling recognition. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1114–1119. IEEE (2011)
17. Shi, B., et al.: American sign language fingerspelling recognition in the wild. In: 2018 IEEE Spoken Language Technology Workshop (SLT), pp. 145–152. IEEE (2018)
18. Shi, B., Rio, A.M.D., Keane, J., Brentari, D., Shakhnarovich, G., Livescu, K.: Fingerspelling recognition in the wild with iterative visual attention. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5400–5409 (2019)
19. Tazhigaliyeva, N., et al.: Cyrillic manual alphabet recognition in RGB and RGB-D data for sign language interpreting robotic system (SLIRS). In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 4531–4536. IEEE (2017)
20. Wang, H., Stefan, A., Moradi, S., Athitsos, V., Neidle, C., Kamangar, F.: A system for large vocabulary sign search. In: Kutulakos, K.N. (ed.) ECCV 2010. LNCS, vol. 6553, pp. 342–353. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35749-7_27
21. Warchoł, D., Kapuściński, T., Wysocki, M.: Recognition of fingerspelling sequences in polish sign language using point clouds obtained from depth images. Sensors **19**(5), 1078 (2019)