DeepSolfège: Recognizing Solfège Hand Signs Using Convolutional Neural Networks

Dominic Ferreira(✉) and Brandon Haworth

University of Victoria, Victoria, Canada
{dominicf,bhaworth}@uvic.ca

Abstract. Hand signs have long been a part of elementary music theory education systems through the use of Kodály-Curwen Solfège hand signs. This paper discusses a deep learning convolutional neural network model that can identify 12 hand signs and the absences of a hand sign directly from pixels both quickly and effectively. Such a model would be useful for automated Solfège assessment in educational environments, as well as, providing a novel human computer interface for musical expression. A dataset was designed for this study containing 16,900 RGB images. Additional domain-specific image augmentation procedures were designed for this application. The proposed CNN achieves a precision, recall, and F1 score of 94%. We demonstrate the model's capabilities by simulating a real-time environment.

Keywords: CNN · Image classification · Solfège

1 Introduction

Solfège is a musical education method that makes use of static hand signs to represent musical notes. The hand signs correspond to the 12 tone equal temperament system, common in Western Music, and act as a kinesthetic aid for learning singing [10,17]. This elementary system's popularity, history, and formalized mapping of hand signs make it a good candidate to automate the identification process so that it can be deployed in learning applications.

This paper identifies an effective convolutional neural network for classifying Solfège hand signs. During each stage of development, careful consideration was made to make this network useful in a wide variety of applications. Particular emphasis was placed on ensuring real world performance was responsive, adaptable, and accurate. The target platform during experimentation was a mid-range consumer PC using a single webcam. The two proposed use cases that would benefit from the CNN identified in this paper are that of applications in the educational and artistic domains. Many education systems globally are increasing their use of online resources and automated evaluations which facilitate larger audiences as well as distance learning. These educational tools often require tailored solutions to provide the most effective experience for the users. Using our

G. Bebis et al. (Eds.): ISVC 2021, LNCS 13017, pp. 39–50, 2021.
https://doi.org/10.1007/978-3-030-90439-5_4

CNN, an educational application could be built which teaches these signs to students and validates that they are correctly learning the techniques. Any curriculum that utilizes Solfège could deploy this application as a convenient way to track a student's progress.

Additionally, this CNN could provide a system to be used as an input device to control an instrument. A common technical standard for controlling instruments is MIDI. The predicted labels could be output as MIDI signals, allowing for seamless integration into many synthesizers and digital audio workstations. This would allow for live performances using only the Solfège hand signs as an input.

In this paper, we propose a CNN for real-time Solfège hand sign classifications which was trained on our augmented dataset. Our contributions include the CNN architecture, dataset curation, augmentation methodology, and a real world simulation. We evaluate our method extensively by analyzing and presenting the precision, recall, F1 scores, and confusion matrix for our proposed architecture. Our model is cross-validated and subject to an ablation study. We also compare the effects that input resolution has on accuracy and computation time. Finally, we simulate real world accuracy by feeding a video through the network to validate usability in applications.

2 Related Work

Barehanded image recognition has a long history in human computer interaction applications [19]. In recent years, Convolutional Neural Networks (CNN) have been a boon for image classification, recognition, segmentation, but also barehanded human computer interaction [5]. CNNs are capable of learning and extracting features directly from pixels, recognizing patterns, classifying images, and have been used to solve similar problems to the domain presented here [8]. Advances in that field which focused on human computer interaction have created many educational and accessibility tools, such as sign language recognition [2]. Sign language recognition is an interdisciplinary topic combining elements of computer vision, natural language processing, and machine learning. While some problems in this field require temporal information or linguistic considerations, static image classification does not rely on context external to the current frame. Some work has been done specifically to solve the temporal aspect of recognizing hand gestures in sign language by implementing the use of 3D CNNs [4]. In this paper, we propose single frame predictions on static hand signs which is similar to other static sign language detection research [20]. We account for real-time use on video through careful dataset augmentation and include a real-time video based analysis for evaluation. A comparison of CNN models used for static image classification is found in Table 1.

Solfège as a whole encompasses several techniques and is often combined with other methods to aid in learning musicianship skills. For example, there are methods that use a set of syllables to help memorization and audiation of pitches, or assigning words and phrases to different rhythms. Other works have

Table 1. Related CNN models used for classification

Classification domain	Input dimensions	Convolution layers	Fully connected layers	Regularization	Accuracy
Poultry health [6]	150 × 300	3	1	Dropout	86%
Dentistry [3]	996 × 564	6	2	Dropout	87%
Sign language [12]	128 × 128	4	1	None	92%
Age and Gender [1]	227 × 227	4	2	Dropout and batch normalization	96%

automated the assessing of accuracy for singing pitch or gestural tempo [13,14]. Another work uses pitches played back to visually impaired users to help detect and analyze the position of objects in front of them [7]. While these works do use Solfège elements in unique applications, they do not contribute to the specific domain of hand sign classification.

Solfège hand sign recognition is an under-researched area with only a single work covering the topic. The only related work feeds an isolated hand silhouette into a random forest classifier running on a Google Glass device to make near real time predictions [15]. While the classification accuracy reported in this paper is high at 95%, it uses only seven Solfège hand signs, which significantly reduces the musical possibilities. The Google Glass device used captures images in an egocentric perspective, which is an uncommon perspective in consumer capture hardware like laptops, cell phones, and webcams.

3 Dataset

The dataset used in these experiments was built specifically for this application. No other dataset for this kind of application is publicly available to the best of the Authors' knowledge. It contains 16,900 photos evenly split across 13 different label classifications and with images captured from several different environments. All photos were captured in RGB format at a resolution of 640×480 pixels using a readily available consumer webcam. Several considerations went into the design of this dataset, most notably the label selection, collection hardware, and background environments.

3.1 Labels

There are several Solfège hand signs and variations, so we must specify which are included in the dataset. A subset of the possible Solfège signs were selected by balancing functionality and complexity. To represent the 12 tones found in the chromatic scale, at least 12 symbols are needed. Solfège includes several enharmonically equivalent notes, meaning that there are multiple hand signs that map to the same notes. To reduce the number of labels, we have not included any

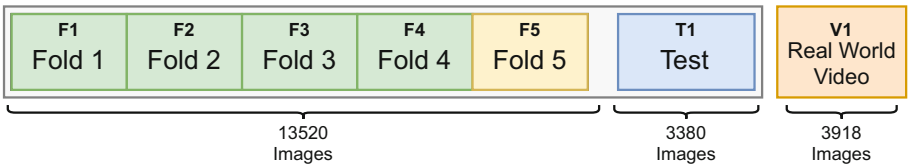

Fig. 1. Dataset and subsets breakdown

enharmonically equivalent symbols in our dataset. The 12 selected symbols used are the main seven Diatonic notes, as well as the five Flat notes, as seen in Fig. 2. Some methods of teaching Solfège add the additional element of assigning each symbol a location in vertical space. This potentially complicates the network used for identification and thus was omitted. In return, this grants the user freedom to show the sign anywhere within the frame and for the network to be able to recognize it. We discuss this further in our Sect. 6. Another necessary label in the dataset is a null symbol representing any frame that does not contain one of the 12 selected Solfège signs. This would be useful in both educational and artistic applications when a user does not want to play a note, as well as real time applications for disambiguation of hand movement between notes. A similar approach is used in the other work with their inclusion of a 'no gesture' class; however, they use another additional label for 'noisy' data, when no hand sign is present [15]. We combine the content of those two separate labels into a single one, which we call 'no symbol', since the functionality desired is the same in either case. The data collected for our null symbol include images with no hands visible, a hand that is visible but not showing any Solfège sign, and a hand that is showing a Solfège sign but is blurred or obscured beyond human recognition. The latter class of images was derived from empirical tests that showed this improved the misclassification of hand movements between symbols in the real-time case where motion blur may be extreme on consumer webcams. This brings the total number of labels to 13.

3.2 Preprocessing

The first stage of processing the dataset is to break it into three different subsets; training, validation, and test sets. Twenty percent of the dataset was reserved for testing, $T1$ as seen in Fig. 1. The remaining set is broken into five even pieces, $F1$ through $F5$, which will be used for cross-validation during training. All inputs into the network are scaled down to a resolution of 80×60 pixels for network input. This resolution allows our network to quickly process images, and we found that, below this threshold, important features for classification were being lost and accuracy significantly degraded.

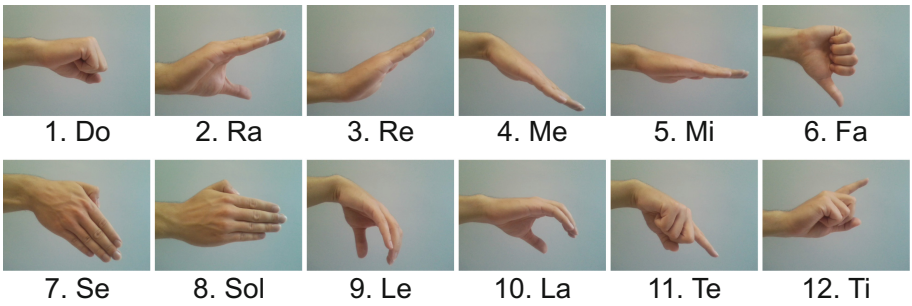

Fig. 2. Solfège hand signs samples from dataset, in ascending musical order. Typical samples from the dataset include noisy backgrounds, humans, and typical indoor background elements, the samples shown here have been cropped for symbol clarity.

There were several different augmentations applied to the dataset during training. The set of possible augmentations, S, correspond to seven unique parametric transformations, A_1 through A_7, such that S_n can be any combination of A_1 through A_7 within their respective ranges given in Table 2.

Table 2. Transformations used for dataset augmentation

Index	Transformation	Range
A_1	Rotation	±12°
A_2	Zoom out	10–30%
A_3	Width shift	±15%
A_4	Height shift	±15%
A_5	Shear	±15%
A_6	Brightness reduction	30%
A_7	Horizontal flip	Binary

We constrain the range for A_1 such that it preserves the image's orientation, since some of the hand signs are rotationally variant, and we are designing for a landscape input. A_2 is used to ensure that after any combination of transformation, the hand sign is still within the input frame. After S_n is applied to an image, any part of the transformed image that does not fill the entire CNN input dimension is filled with black pixels.

4 Method

The proposed CNN architecture and variations were inspired by similar state-of-the-art designs. This section outlines an effective CNN, discusses our training procedures, and compares variations in an ablation study.

4.1 CNN Architecture

Our architecture uses three convolutional layers, each followed by a batch normalization layer, Maxpool layer, and then dropout layer, after which is flattened and passed to a single fully connected layer before the output layer with the final 13 neurons. Figure 3 provides a visual representation of our architecture. Every convolutional and dense layer used a ReLU function for activation, with the exception of the output layer, which used a Softmax function. Each convolution uses a 3×3 kernel size, a stride of two, 64 filters, and is zero padded to output the same height and width dimension as the input. The Maxpooling layer uses a 2×2 window and is used to downsample the feature maps. Each dropout layer randomly sets 30% of input units to zero. After being flattened, the input is passed into a fully connected layer with 1024 neurons. The last layer uses a Softmax activation to output a probability across our 13 classes. The input to the system is a 80×60 RGB image array.

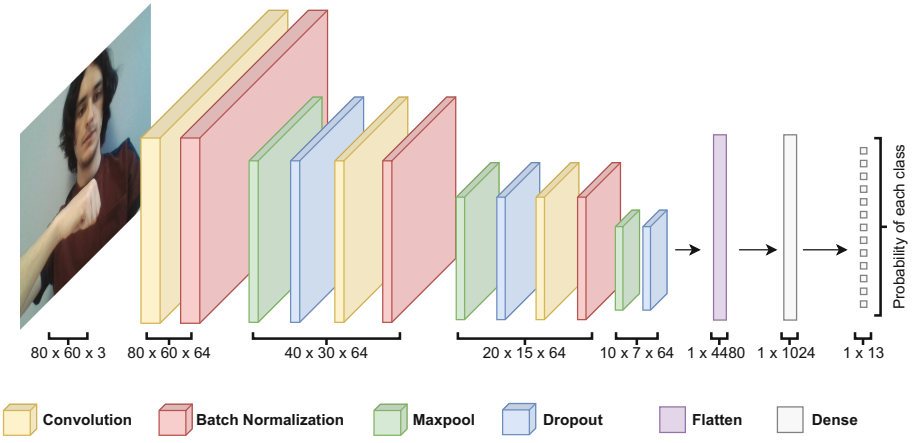

Fig. 3. Model architecture, from the input image on the left to the output probability on the right.

4.2 Training

Our proposed model, as well as all other models constructed for comparison, use the same training methods and parameters outlined in this section. We used Keras as our framework for all of our experiments, and trained the models using an AMD Ryzen 9 5900X CPU. We used the Adam optimization algorithm, with a learning rate of 0.001, beta 1 of 0.85, beta 2 of 0.999, and epsilon value of 1e−7. Each model was trained over 250 epochs with a batch size of 64. Each image used during learning was passed through S_n for augmentation at run-time. Checkpoints were saved as the model trains, and the checkpoint with the highest validation accuracy was selected.

4.3 Ablation Study

In this section we perform an ablation study on our proposed architecture. Four elements of our architecture were tested in isolation to validate the design. The components considered in this study are the pooling layer methods, normalization techniques used, convolution layer configuration as well as number of neurons in the fully connected layer.

We compare two different pooling methods in Table 3, where every pooling layer in the model is replaced with the corresponding method. Both batch normalization and dropout are techniques used to mitigate overfitting and reduce training time [11,16], and as seen in Table 1, are commonly used in similar applications. We compare the effects that a variety of configurations has on our model in Table 4. The batch normalization and dropout layers used in this experiment have been used between every convolution layer in the network. Table 5 shows the result of adding an additional duplicate convolution layer, as well as removing one. Table 6 compares a range of values for the number of neurons in the fully connected layer of the network. Each component of our proposed network utilizes the optimal solution within the range of tested configurations.

Table 3. Pooling method comparison

Pooling method	Accuracy
MaxPooling2D	**93.9%**
AvgPooling2D	91.6%

Table 4. Regularization comparison

Regularization method	Accuracy
None	89.9%
Dropout	88.6%
Batch normalization	90.4%
Both	**93.9**%

Table 5. Convolution layer comparison

Convolution layers	Accuracy
2	86.6%
3	**93.9**%
4	84.3%

Table 6. Fully connected layer comparison

Layer width	Accuracy
1024	**93.9**%
512	90.9%
256	90.5%

5 Evaluation

Our proposed model was five-fold cross-validated and achieved an average accuracy of 93.3%, with the best model having an F1 score of 93.9% on the $T1$ dataset. The best model was selected from the cross-validation and was used for the metrics in Table 7 and Fig. 4. The precision, recall and F1 scores are presented in Table 7. Figure 4 is a confusion matrix, which highlights a couple

classes that perform poorly. The largest anomaly in the confusion matrix illustrates the class Le's poor recall performance, in particular that it misclassifies it as Te 5.4% of the time. This can be attributed to the nature of the hand symbols sharing similar features, as seen in Fig. 2.

Table 7. Model classification report. Each class has 260 samples of support.

	Precision	Recall	F1 score
Do	0.881	0.912	0.896
Fa	0.959	0.908	0.933
La	0.956	0.923	0.939
Le	0.940	0.896	0.917
Me	0.969	0.969	0.969
Mi	0.918	0.950	0.934
No symbol	0.936	0.954	0.945
Ra	0.961	0.958	0.960
Re	0.944	0.969	0.956
Se	0.946	0.942	0.944
Sol	0.945	0.985	0.964
Te	0.892	0.923	0.903
Ti	0.968	0.919	0.943
Average	0.940	0.939	0.939

5.1 Real World Application

Two experiments have been conducted to explore the real world performance of the proposed CNN.

The resolution comparison in Table 8 explores the relationship between accuracy and computational cost. The model was retrained using different input resolutions. The computational cost is calculated by averaging the time it takes to make a prediction using only a CPU, an Intel i7-6700HQ and an AMD Ryzen 9 5900X, in a Jupyter Notebook environment, across the 3380 samples in $T1$. To contextualize the requirements for computation time of a single frame in terms of music, a single 32nd note at 180 beats per minute lasts 42 ms. Another work studied how sensitive humans are to latency when using a gesture controlled instrument [9]. The work claims a 20 to 30 ms just noticeable difference, which means our proposed model is just within the acceptable tolerance. Using only the CPU for predictions increases accessibility since it does not rely on the user possessing a capable, discrete GPU. A GPU implementation may be preferable for optimal performance because of the classification speed benefits [18].

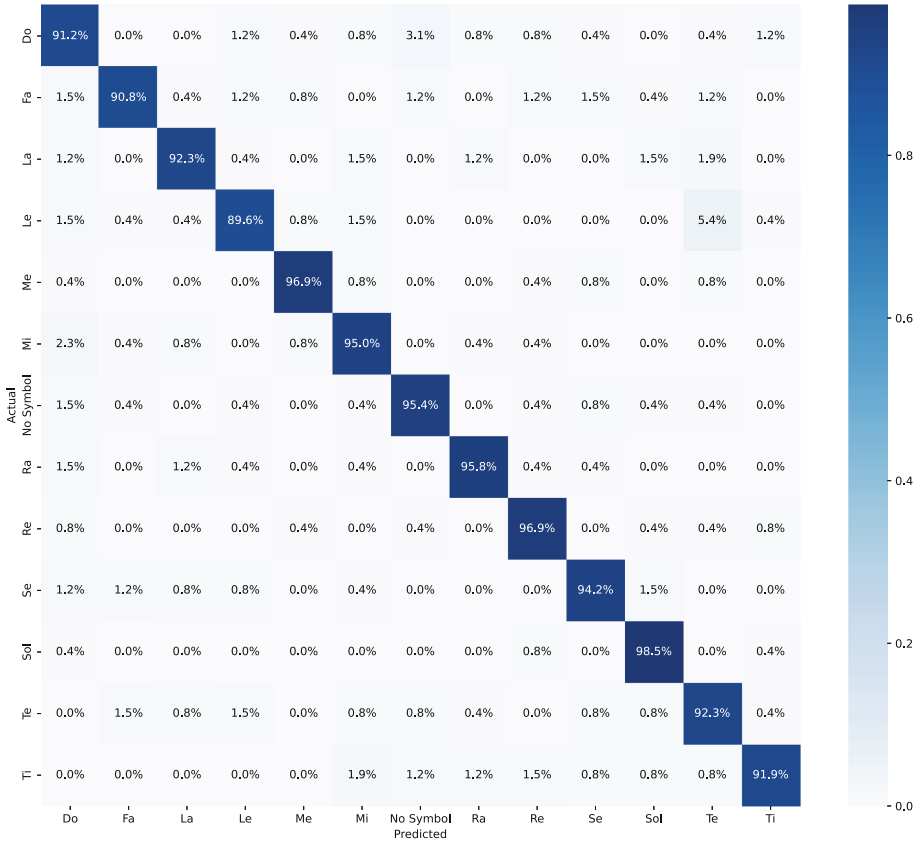

Fig. 4. Confusion matrix

Table 8. Resolution performance comparison

Resolution	Accuracy	Computation time i7-6700HQ	Computation time Ryzen 9 5900X
160 × 120	90.9%	51.3 ms	25.1 ms
80 × 60	**93.9%**	**42.5 ms**	**19.5 ms**
40 × 30	88.0%	38.6 ms	17.0 ms

To evaluate the model's accuracy on a real world incoming video feed, we created a separate dataset, $V1$, to simulate the environment. A continuous video was captured at 30 frames per second where all 13 classes were performed in sequential order with each appearing for approximately 10 s. The frames were then annotated by a human, and then compared against the model's offline predictions. The model achieved an accuracy of 89.3% over the 3918 frames. Another experiment is necessary to identify the other computational costs not factored

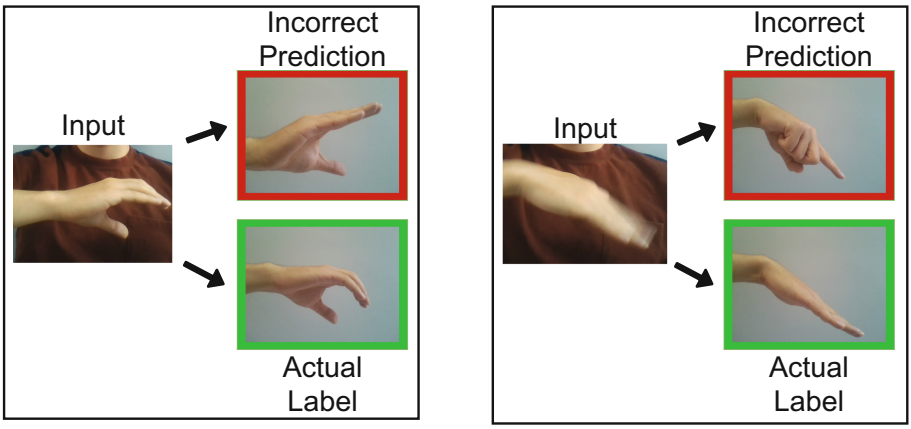

Fig. 5. The two most misclassified labels during real world experiment. Input images have been cropped to focus on hand symbol for this diagram.

into these calculations. The latency for capturing webcam frames, processing them into the network, outputting the class, creating the MIDI signal, and synthesizing the corresponding sound were not included in our analysis. Figure 5 illustrates some of the worst performing classes during the real world experiment, and highlights the similarity of visual features between certain symbols. A three-dimensional CNN which uses a sequence of frames could be a possible solution for low confidence predictions and the transitions interval between hand signs.

6 Conclusion

This work builds a foundation for future Solfège applications that require hand sign classification. Our proposed convolutional neural network achieves a 94% accuracy and is capable of real-time predictions. Our evaluations motivate further work into utilizing the model as a viable hand sign input method. In future work, we hope to further improve handling low confidence classifications, as well as, ensure the model's robustness in a variety of environments. This also includes the investigation of temporal methods and even sequence or Markov methods for future sign prediction and correction. The video and static image datasets are carefully varied to cross a broad range of environments and lighting conditions, however they include the same participant. Similarly, a broader range of camera sources and methodologies for normalizing sources for input should be investigated for Solfège specific applications. Thus further dataset development is needed to broaden the generalizability to arbitrary users. With the 13 classes used in our dataset, we are limited to represent a single musical octave. By having another input to represent octaves, one could combine such a method with our model to map the entire chromatic scale. This is advantageous for designing

a fully featured MIDI input method. In fact, with the success of this method and the utility of Solfège, many human-computer interaction, usability, usefulness, education, and training opportunities arise. We look forward to expanding the dataset, exploring usability and use cases, and optimizing the network for specific applications.

References

1. Agbo-Ajala, O., Viriri, S., et al.: Age group and gender classification of unconstrained faces. In: Bebis, G. (ed.) ISVC 2019. LNCS, vol. 11844, pp. 418–429. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33720-9_32
2. Ardiansyah, A., Hitoyoshi, B., Halim, M., Hanafiah, N., Wibisurya, A.: Systematic literature review: American sign language translator. Proc. Comput. Sci. **179**, 541–549 (2021)
3. Campos, L.S., Salvadeo, D.H.P.: Multi-label classification of panoramic radiographic images using a convolutional neural network. In: ISVC 2020. LNCS, vol. 12509, pp. 346–358. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64556-4_27
4. Huang, J., Zhou, W., Li, H., Li, W.: Sign language recognition using 3D convolutional neural networks. In: 2015 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2015)
5. Islam, M.M., Islam, M.R., Islam, M.S.: An efficient human computer interaction through hand gesture using deep convolutional neural network. SN Comput. Sci. **1**(4), 1–9 (2020)
6. Jørgensen, A., Fagertun, J., Moeslund, T.B., et al.: Classify broiler viscera using an iterative approach on noisy labeled training data. In: Bebis, G. (ed.) ISVC 2018. LNCS, vol. 11241, pp. 264–273. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03801-4_24
7. Kalra, S., Jain, S., Agarwal, A.: Fixed do solfège based object detection and positional analysis for the visually impaired. In: 2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), pp. 594–598. IEEE (2017)
8. Khan, A., Sohail, A., Zahoora, U., Qureshi, A.S.: A survey of the recent architectures of deep convolutional neural networks. Artif. Intell. Rev. **53**(8), 5455–5516 (2020). https://doi.org/10.1007/s10462-020-09825-6
9. Mäki-Patola, T., Hämäläinen, P.: Latency tolerance for gesture controlled continuous sound instrument without tactile feedback. In: ICMC. Citeseer (2004)
10. McClung, A.C.: Sight-singing scores of high school choristers with extensive training in movable solfège syllables and curwen hand signs. J. Res. Music Educ. **56**(3), 255–266 (2008)
11. Park, S., Kwak, N.: Analysis on the dropout effect in convolutional neural networks. In: Lai, S.-H., Lepetit, V., Nishino, K., Sato, Y. (eds.) ACCV 2016. LNCS, vol. 10112, pp. 189–204. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54184-6_12
12. Rao, G.A., Syamala, K., Kishore, P., Sastry, A.: Deep convolutional neural networks for sign language recognition. In: 2018 Conference on Signal Processing And Communication Engineering Systems (SPACES), pp. 194–197. IEEE (2018)
13. Schramm, R., Nunes, H.D.S., Jung, C.R.: Audiovisual tool for solfège assessment. ACM Trans. Multi. Comput. Commun. Appl. (TOMM) **13**(1), 1–21 (2016)

14. Schramm, R., de Souza Nunes, H., Jung, C.R.: Automatic solfège assessment. In: ISMIR. pp. 183–189 (2015)
15. Sörös, G., Giger, J., Song, J.: Solfège hand sign recognition with smart glasses. In: First International Workshop on Egocentric Perception, Interaction, and Computing (EPIC 2016). First International Workshop on Egocentric Perception, Interaction, and ... (2016)
16. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
17. Steeves, C.: The effect of Curwen-Kodaly hand signs on pitch and interval discrimination within a Kodaly curricular framework. University of Calgary (1984)
18. Strigl, D., Kofler, K., Podlipnig, S.: Performance and scalability of GPU-based convolutional neural networks. In: 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing, pp. 317–324. IEEE (2010)
19. Von Hardenberg, C., Bérard, F.: Bare-hand human-computer interaction. In: Proceedings of the 2001 Workshop on Perceptive User Interfaces, pp. 1–8 (2001)
20. Wadhawan, A., Kumar, P.: Deep learning-based sign language recognition system for static signs. Neural Comput. Appl. **32**(12), 7957–7968 (2020). https://doi.org/10.1007/s00521-019-04691-y