engineering proceedings

MDPI

Proceeding Paper

ASL Fingerspelling Classification for Use in Robot Control [†]

Kevin McCready [1,*], Dermot Kerr [1], Sonya Coleman [1] and Emmett Kerr [2]

[1] School of Computing, Engineering and Intelligent Systems, Ulster University—Magee Campus, Northland Rd, Londonderry BT48 7JL, UK; d.kerr@ulster.ac.uk (D.K.); sa.coleman@ulster.ac.uk (S.C.)
[2] Department of Electronic and Mechanical Engineering, Atlantic Technological University—Donegal Letterkenny Campus, Port Rd, Gortlee, F92 FC93 Letterkenny, Donegal, Ireland; emmett.kerr@atu.ie
* Correspondence: mccready-k@ulster.ac.uk

[†] Presented at the 39th International Manufacturing Conference, Derry/Londonderry, UK, 24–25 August 2023.

Abstract: This paper proposes a gesture-based control system for industrial robots. To achieve that goal, the performance of an image classifier trained on three different American Sign Language (ASL) fingerspelling image datasets is considered. Then, the three are combined into a single larger dataset, and the classifier is trained on that. The result of this process is then compared with the original three.

Keywords: sign language; machine vision; convolutional neural networks; visual communication

check for updates

Citation: McCready, K.; Kerr, D.; Coleman, S.; Kerr, E. ASL Fingerspelling Classification for Use in Robot Control. *Eng. Proc.* **2024**, *65*, 12. https://doi.org/10.3390/engproc2024065012

Academic Editor: Paddy McGowan

Published: 6 March 2024

1. Introduction

Part of smart manufacturing is focused on improving automation by utilising data and machine learning. This encourages greater uptake of robotics outside of mass production applications where tasks can vary significantly. Because of this, it could be advantageous to develop an intuitive means of Human–robot Interaction (HRI) that does not require specialist knowledge. One approach to this is to make use of voice recognition and Natural Language Processing (NLP) technologies. However, loud noises that can be common in manufacturing environments can render this means of communication difficult, so a visual form of communication may be preferable. An intuitive way to do this is to use hand and body actions. Several methods of visual communication exist, and some of the most sophisticated are sign languages. They are based on codified sets of gestures which can serve as an equivalent to words in a spoken language. One of their features is fingerspelling, a way of signing individual letters of the alphabet. Gesture and sign language recognition can form the basis of an intuitive form of HRI. This can be built around an image classifier for fingerspelling images, where a robot will take a different action as a response depending on the class an input image was assigned to. Several datasets of fingerspelling images are already available. Here, classifiers trained on three of these datasets are compared, as well as a classifier trained on a combination of the three.

2. State of the Art

Gesture recognition requires an input image and a classifier to extract useful data from it. Numerous techniques have been proposed for this task, combining different computer vision technologies with machine learning approaches. Image classifiers can operate on a number of machine learning principles, such as Support Vector Machines (SVM) [1], k-nearest neighbours (k-NN) [2], and artificial neural networks (ANN), among others.

Bhushan et al. [3] analysed multiple different approaches to computer vision for gesture recognition and found that RGB images classified with a Convolutional Neural Network (CNN) can produce accuracy results of 91.41%, the best of all methods they considered. Bao et al. [4] lend further support to this, as they were able to achieve an accuracy of 97.1% using RGB images classified using a CNN.

3. Materials and Methods

The three datasets used, referred to here as archives 0, 1, and 2, are all obtained from Kaggle [5–7]. All three datasets contain images of the fingerspelling signs that make up the American Sign Language (ASL) alphabet. Archives 0 and 2 also contain additional signs (such as "space" and "delete"), whereas archive 1 contains only the 26 classes corresponding to the letters of the alphabet.

Before progressing to classifying images, the additional classes must be removed, as well as ensuring that the remaining classes are balanced by containing the same number of images per class. Balancing is achieved by taking a random sample of images, with the size of the sample determined by the size of the smallest class. For example, in archive 0, all the classes contained 70 images, except for the letter t, which had 65, so a sample of 65 was taken from the other classes. Archive 0 now consisted of 65 images per class, Archive 1 3000 per class and Archive 2 4542 per class.

With the datasets balanced, the decision was made to split them in half, so that the classification task would be carried out on the first half of the alphabet only, i.e., letters A through M. These thirteen classes corresponded to thirteen already programmed robot actions, such as opening and closing a gripper and moving the end effector to predefined locations.

Because of this, a CNN was selected for the classifier. The same CNN structure was used throughout. As some of the images in these datasets depicted signs being made by both left and right hands, adding some image augmentations could be beneficial, particularly horizontal flips. The image augmentations used included random horizontal flips, random rotations, random zooms, and gaussian noise. In addition, dropout was added to the model. Next, all images were resized to 200 × 200 pixels, and the datasets were split into training and testing subsets, with an 80–20 training-testing split.

The first layer of the model was the image augmentation layer, which was followed by a rescaling layer. This took the pixel values of the image and rescaled them to range between 0 and 1, rather than 0 and 255. This was followed by a convolutional layer of 16 nodes and a max pooling layer. This convolutional/max pooling combination was repeated 2 more times, first with 32 nodes, then 64. After the third max pooling layer, there was the dropout layer, a flattened layer, then a dense layer with 128 nodes. Then, there was the final output layer with 13 nodes, one for each of the classes the image could be assigned to.

4. Results

This model was now trained on the three datasets, starting with archive 0, as, at this point, it is unknown what the ideal number of training epochs are. Ciresan et al. [8] investigated this, and they found that a deeper CNN does not require as many epochs to achieve the same accuracy as a shallower CNN. When testing CNN classifiers on the MNIST dataset, they found an accuracy of 98.42% after 3 epochs, improving to 99.32% after 17 epochs. Based on these findings, 10 epochs are deemed a reasonable compromise between accuracy and training speed.

The training process was completed in 115 s, and the classifier's performance on testing data peaked after 7 epochs at 100%, before going into slight decline, arriving at 96.45% after 10 epochs due to some overfitting.

The same model was now trained on the archive 1 image dataset. Again, it was trained over the course of 10 epochs. This process took 114 min, which is not unexpected considering the archive 1 dataset contains many more images than archive 0. Performance on testing data peaked at 34.14%, again after the seventh epoch, but given that there were still significant fluctuations between each epoch's results, accuracy did not appear to have plateaued after 10 epochs, so training for additional epochs may lead to improved results.

Finally, the model was trained on archive 2. This took 162 min—the longest training process yet, as archive 2 was the largest dataset of the three. The training accuracy steadily improved over the ten training epochs, peaking at 100% after the tenth and final epoch. This classifier performed best when trained on datasets 0 and 2.

A combined dataset was now created, consisting of all the images used previously. The same model as before was now trained on 7607 images per class, with 65 from archive 0, 3000 from archive 1, and 4542 from archive 2. The classifier now peaked at a validation accuracy of 55.76% after 9 epochs. However, just like the archive 1 classifier, there was still some fluctuation from one epoch to the next.

As seen in Figure 1, the classifier performs best when trained on the archive 2 dataset, sitting consistently in the high 90% range throughout all 10 training epochs. This is likely due to it being the largest of the three datasets. When trained on archive 0, the classifier also performed well, as it caught up to archive 0's validation accuracy within 7 epochs. The classifier performed the poorest when trained on the archive 1 dataset. This is likely due to it being the most varied of the datasets. While all images in archive 1 are of the same signer, lighting conditions change significantly over the course of the images, as does the position of the signer's hand in the image and how close it is to the camera. When trained on the combined dataset, the classifier shows a performance somewhere between archives 1 and 2, as would be expected. Archive 0 has little effect on this, as it is by far the smallest of the three original datasets.

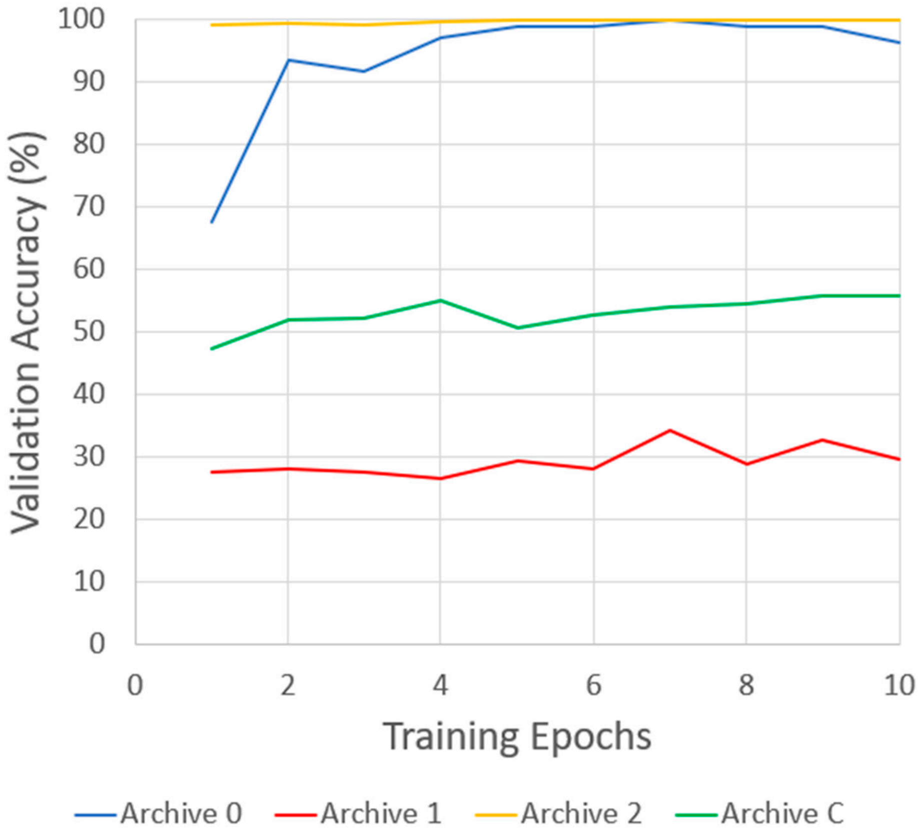

Figure 1. Validation accuracy results of classifiers trained on the three initial datasets (archive 0–2) and the combined dataset (archive C).

5. Conclusions

A CNN image classifier was presented and trained on three separate datasets of ASL fingerspelling images. An accuracy of 100% was achieved using archives 0 and 2 within seven epochs of training but archive 1 was not able to match that result, likely due to greater variance in the images within that dataset. The same is true of the combined dataset. For the purposes of gesture-based robot control, the combined dataset would be the best to train the classifier on but will likely need more epochs of training to improve its accuracy to the point where it could be used reliably.

Author Contributions: Writing—original draft preparation, K.M.; writing—review and editing, E.K., D.K. and S.C.; supervision, E.K., D.K. and S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in American Sign Language Dataset at: https://www.kaggle.com/datasets/ayuraj/asl-dataset, American Sign Language at 10.34740/kaggle/dsv/2184214, and ASL Alphabet at 10.34740/kaggle/dsv/29550.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bar, O.; Trivedi, M.M. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2368–2377.
2. Vailaya, A.; Jain, A.; Zhang, H.J. On image classification: City images vs. landscapes. *Pattern Recognit.* **1998**, *31*, 1921–1935. [CrossRef]
3. Bhushan, S.; Alshehri, M.; Keshta, I.; Chakraverti, A.K.; Rajpurohit, J.; Abugabah, A. An experimental analysis of various machine learning algorithms for hand gesture recognition. *Electronics* **2022**, *11*, 968. [CrossRef]
4. Bao, P.; Maqueda, A.I.; del Blanco, C.R.; García, N. Tiny hand gesture recognition without localization via a deep convolutional network. *IEEE Trans. Consum. Electron.* **2017**, *63*, 251–257. [CrossRef]
5. Thakur, A. American Sign Language Dataset. 2019. Available online: https://www.kaggle.com/datasets/ayuraj/asl-dataset (accessed on 28 March 2023).
6. Londhe, K. American Sign Language. 2021. Available online: https://www.kaggle.com/datasets/kapillondhe/american-sign-language (accessed on 24 March 2023).
7. Nagaraj, A. Asl Alphabet. 2018. Available online: https://www.kaggle.com/datasets/grassknoted/asl-alphabet (accessed on 24 March 2023).
8. Ciresan, D.C.; Meier, U.; Masci, J.; Gambardella, L.M.; Schmidhuber, J. Flexible, high performance convolutional neural networks for image classification. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, 16–22 July 2011; Citeseer: State College, PA, USA, 2011.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.