Efficient Interpretation of Irish Sign Language: Building and Compressing Machine Learning Models

Emma Urquhart
Department of Computer Science, Cambridge University
Cambridge, United Kingdom
eu233@cam.ac.uk

Abstract

Machine learning methods have the potential to increase accessibility across many domains, including empowering the deaf and hard-of-hearing though automation of sign language interpretation. Irish Sign Language (ISL) is among the under-resourced sign languages, with only one dataset publicly available. As a result, research on machine learning for ISL remains extremely limited. This work proposes a step towards bridging this gap. It is shown that lightweight ISL static sign interpretation is possible, achieves test accuracy of 97% and achieves satisfactory performance in real time. Furthermore, dynamic sign interpretation is also explored in the context of the LSTM architecture and the results of successful compression of this model is demonstrated. The baseline architecture developed achieves 96% test accuracy. Pruning reduces the model size by a factor of 4.8, resulting in accuracy of 92%, while post-pruning quantization achieves an overall reduction in model size of a factor of 18, resulting in accuracy of 87.5%. Various measures are implemented towards mitigating the shortcomings within the dataset and their effectiveness may be verified through reality-centric evaluation. In the case of static sign interpretation, a real-time translation system is successfully implemented, while the dynamic gesture recognition models successfully classify pre-recorded video clips. These measures facilitate an increased confidence level in the performance capabilities of the proposed solutions. While the need for continued data acquisition and research in this area remains, this work provides a baseline for future works and an easily-extensible framework for further model training on new data. This work aims to contribute to the reduction of communication barriers in Ireland and provide an extensible solution which can be refined upon further data collection. Project link: https://github.com/eurquhart1/ML_Systems_Project_ISL_Translation

1 Introduction

Rapid developments in machine learning have led to advancements across a wide range of applications in recent years. Language processing and computer vision are among the many applications that have undergone significant improvements. These advancements have the potential to benefit many different stakeholders, including the deaf community through automated sign language translation. Lightweight, automated sign language recognition using machine learning could facilitate easier communication between the minority of deaf people whose preferred language is sign language, and the general population. Deployment on low-resource devices could enable offline, real-time translation, which would provide an increased level of independence for sign language users. Having been somewhat left behind in the wave of AI developments, Irish Sign Language (ISL) remains a popular indigenous language. This study proposes an initial step towards reconciling this gap, demonstrating the efficacy of an ISL translation application, and carving a path forward to making AI innovations in computer vision, deep learning and model compression beneficial to a wider audience.

Two different approaches are proposed for static and dynamic sign language translation, respectively, with the latter requiring a more complex neural network due to the temporal relationship between frames in a video clip of a sign language gesture. Furthermore, different model compression techniques are explored in the context of the dynamic gesture translation model. Knowledge distillation, pruning and quantization are evaluated for their efficacy in preserving accuracy while reducing the model size.

2 Background

2.1 Irish Sign Language (ISL)

Irish Sign Language (ISL) is used by many people on the island of Ireland, being the first language of over 5,000 deaf people and used for communication by over 40,000 [8]. ISL comprises over 5000 different signs, each of which is composed of a hand shape and a dynamic movement. The 23 basic signs in ISL and 3 primary motion gestures are each denoted by a different letter of the alphabet. ISL is distinct from other sign languages, such as British and American, however little research has been conducted to advance the state of automation for ISL translation. There remains only one publicly-available dataset for ISL, the ISL-HS dataset [7]. Although limited in size, its data follows a format that makes it easily extensible (video clips of signs, split into frames).

2.2 Related Literature

The only current publicly-available dataset for Irish Sign Language translation was developed by Oliveira, M. et al. [7]. The authors developed and evaluated the performance of a model incorporating principal component analysis and a k-nearest neighbours classifier on this dataset, which resulted in 95% accuracy, but the model was designed for static gesture recognition only. Holmes, R. et al [4] demonstrate the effectiveness of fine-tuning larger models which have been pre-trained on larger datasets for different sign languages, for ISL interpretation. This work uses models and datasets which are not publicly available, including the "Signs of Ireland" dataset [6]. A rotation-agnostic approach to static sign recognition is proposed by Kalam, M. A. et al [5], whereby the authors develop a convolutional neural network which classifies sign gestures irrespective of the camera angle. This approach towards making the model more robust has inspired the data augmentation technique applied in this study, which applies rotation and scaling transformations to the dataset. Many researchers have proposed different approaches to sign language gesture recognition. For continuous, dynamic sign language translation, Long Short-Term Memory (LSTM) has been successfully implemented in several publications for different sign languages; Yang, S. proposed a new approach for dynamic translation of Chinese Sign language [9], Abraham, E. for Indian Sign Language [2] and Abdullahi, S. for American Sign Language [1]. An LSTM is a form of recurrent neural network, which incorporates memory cells to enable retention of data observations for longer periods of time. Given the prevalence of this approach in the literature and its evident versatility across different sign languages, it was selected as an appropriate model for dynamic gesture recognition in this study.

3 Methodology

The primary objective of this study was to provide a solution which can reliably interpret ISL sign gestures with high accuracy and efficiency, in real-world settings. The ISL alphabet is comprised of both static and dynamic sign gestures and as a result, this work examines both separately. A lightweight neural network, which processes images in real time, is implemented for static sign gesture recognition. Due to the importance of the temporal relationship between frames in dynamic gesture recognition, a more complex LSTM network is implemented, which processes video clips. Due to the limited nature of the dataset, the validation results were supplemented with real-world evaluation involving interacting with a real-time system for static recognition and recording video clips for classification by the dynamic sign recognition system. Fine-tuning of a pre-developed model which had been trained on a larger dataset for another sign language was also considered. However, given the promising performance demonstrated by the developed solutions, it was decided to invest further time in pursuing them.

3.1 Working with a low-resource dataset

Specific measures were implemented to overcome the challenges posed by the limited dataset used in this study, with the aim of providing an extensible solution which achieves satisfactory performance on the data available and on real-world samples, but which can be refined upon further data collection. The following steps were taken to ensure that the model maintained generalisability in broader contexts. These steps were carefully chosen to also ensure conformation with the low-resource specification of the target architecture.

3.1.1 MediaPipe Hands

The MediaPipe Hands hand and finger-tracking model (Zhang, F. et al [10]) was used in the pre-processing stage. This model identifies hand landmarks corresponding to the positions of various parts of the hand in images. Since the dataset consisted of images which had been staged for easy thresholding (signers wore black clothes and the background was black), models which relied on this feature would not perform well in natural, realistic settings. Pre-processing with MediaPipe Hands circumvents this issue. Since only landmark co-ordinates are recorded, the model should not respond differently to variations in size, skin colour, etc. of the hands in the images. The use of MediaPipe Hands therefore reduced, to a certain extent, the size of the dataset that should be required to accurately train the model. Furthermore, because the resulting model processes lists of landmark coordinates as opposed to entire images, it is more lightweight by nature.

3.1.2 Dataset Augmentation

Data augmentation was applied to the existing dataset after the development of the initial static gesture recognition model. It was noted that when interacting with the real-time system, slight variations in hand angle caused the model to misclassify certain signs. To combat this issue, the proposed data augmentation approach applies 3D rotation and scaling to the existing landmarks to create a more varied and diverse dataset. It was important to retain the realism of the hand image, so the same scaling and rotation factors were applied to all landmarks in the same image. This resulted in a more robust model which was less prone to failure given input images from different angles. In the case of dynamic gesture recognition, the data points were similarly augmented using scaling and rotation.

3.2 Long Short-Term Memory Architecture

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network, first proposed by Hochreiter and Schmidhuber in 1997 [3] to address the vanishing gradient problem found in traditional RNNs. The key innovation behind LSTMs is their capability to keep track of long-term dependencies in time-series or sequence data. An LSTM network is comprised of LSTM cells, each of which contains three gates. The forget gate serves the purpose of determining which information should be retained or discarded from the cell's previous state. The input gate updates the cell state when a new inut is received. Finally, the output gate determines the next hidden state, a state which encapsulates information about previously-processed inputs. The cell state keeps track of important information pertaining to the overall sequence of data that has been encountered and is updated via interaction with the gates. Due to the exceptional ability of LSTMs to capture temporal relationships in sequential data, it is an ideally-suited architecture for processing video clips consisting of sequences of frames and depicting dynamic hand gestures.

4 Experiments and Evaluation

4.1 Dataset

The dataset used in this study was the Irish Sign Language Hand shape dataset (ISL-HS). This dataset is the only publicly-available one of its kind and consists of images of real hands performing the 26 hand gestures of the alphabet, including 23 static and 3 dynamic gestures ('J', 'X' and 'Z'). It consists of 468 videos and 58,114 images.

4.2 Implementation and Training

Model implementations were built from scratch in this project due to the lack of publicly-available ISL interpretation models. The approach followed was to begin with a simple model and increase complexity until the accuracy achieved was satisfactory (>95%). A target accuracy score of 90% was set for the compressed models, as this is a generally-accepted threshold for well-performing models in similar applications. Awareness of the danger of overfitting on the dataset was crucial, but high accuracy was achieved without overly complexifying the models.

4.2.1 Data Pre-processing

In the case of the static image frames, the pre-processing phase involved extraction and labelling of the images from the zipped files in the dataset, application of hand landmarks, storing of the landmark data in CSV files, and finally data augmentation. In the case of the video clips for dynamic gesture recognition, it was necessary for each video clip to be submitted to the network as a sequence of a constant number of frames. The clips were non-uniform in length and therefore an approach was devised to sample a constant number of frames at even intervals throughout each video, or repeat certain frames in the case of a clip being too short. Similarly, landmark data was recorded and the temporal relationship of this data between files enabled the network to capture the features of the gesture.

After the introduction of data augmentation into the pre-processing pipeline for static gesture recognition, the classification accuracy of this model improved from 95% to 97%. Experimentally, it also appeared that the real-time static gesture recognition model had become more robust to deviations in camera angles of the images. The size of the dataset for static sign recognition was scaled by a factor of 40, with each successive augmentation only differing slightly from the previous, with all landmarks scaling uniformly. This ensured that the integrity of the signs were preserved. In the case of dynamic gesture recognition, the dataset was increased by a factor of 12. Rotation adjustments were more restricted in this case, in the interest of preserving realistic arm movement. The code defines two scaling factors ([0.9, 1.1]), two rotation angles (10 and 20 radians), and three rotation axes ([x, y, z]).

4.2.2 Model for Static Gesture Recognition

Because the majority of alphabetic letter gestures in ISL (23/26) are static, an initial model was implemented to recognise these static signs. This neural network was a Sequential classifier built using Keras, with Tensorflow at the back end. It consisted of three dense layers: an input layer with 64 neurons and ReLU activation, a single hidden layer and an output layer which used the softmax activation function, given the multi-class classification nature of the problem. This model processes individual frames or images and is thereby appropriate for image classification of static hand gestures only. After training for 30 epochs with a batch size of 16, validation accuracy of 97% is reached. The final model contains 6935 (trainable) parameters. Its size is 113KB. Given the small size and resource requirements of this model, it is already fit for use on microcontrollers such as the BBC micro-bit. Because the purpose of compression in this context is to achieve as close as possible to real-time inference, an application was built for real-time recognition of static sign gestures using a webcam. All static alphabetical letter signs were successfully detected in real time using this application, some examples of which are shown in 5. Therefore, further compression of this model was deemed unnecessary at this stage.

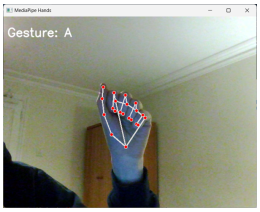

Figure 1: A

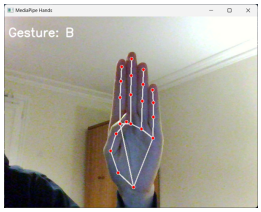

Figure 2: B

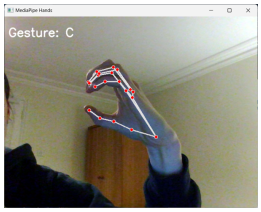

Figure 3: C

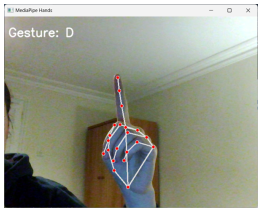

Figure 4: D

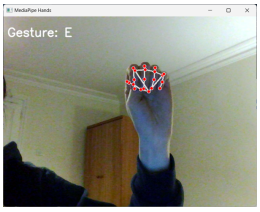

Figure 5: E

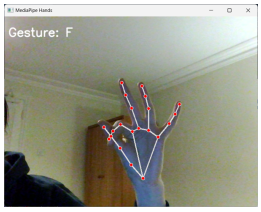

Figure 6: F

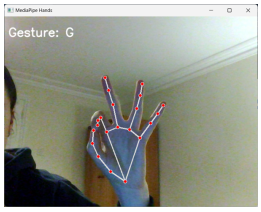

Figure 7: G

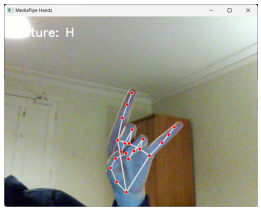

Figure 8: H

4.3 Model for Dynamic Gesture Recognition

The developed model for static gesture recognition did not extend to the application of dynamic gesture recognition. Increasing the complexity of the network failed to improve validation accuracy, which was approximately the same as random classification. As a result, a more complex LSTM neural network was designed to recognise dynamic gestures from short video clips, classifying them into the relevant classes ("j", "y", "z"). This neural network architecture is a sequential model with four layers, specifically designed for processing sequential data such as gestures across a series of frames. It consists of two LSTM layers, followed by a fully-connected dense layer and the output dense layer. The first LSTM layer uses ReLU activation and by setting return_sequences to true, outputs a sequence to the next layer. The addition of these two layers to the network enabled the model to capture the temporal relationships between the hand landmarks in consecutive frames. The final accuracy achieved after 1000 epochs was 96%. Similarly to the static gesture recognition system, it was deemed prudent to verify the performance on a real-world data sample. The model was verified on four pre-recorded data clips of each of the different gestures, recorded in a more natural setting than the original dataset (simply using the webcam). A sample of the clips used are publicly available on the project repository (in test_videos/). Its success suggests the generalisability of the model to real-world applications. It consists of 67971 (trainable) parameters and its size is 774,284KB. For successful deployment on microcontrollers such as BBC micro-bit, the peak memory usage value should be below 64 KB and the model size should be less than 120 KB. Therefore, the developed model required compression in order to achieve sufficiently lightweight performance.

4.4 Compression of Dynamic Gesture Recognition Model

Three compression approaches were investigated in relation to the dynamic gesture recognition model: knowledge distillation, pruning and quantization. The aim of this phase of development was to compress the model to a suitable size for deployment to a microcontroller such as the BBC micro-bit, while maintaining high accuracy on the validation and test data samples.

4.4.1 Knowledge Distillation

Knowledge distillation was initially tested for its applicability to this compression task. A student model was designed as a smaller and less complex version of the original (teacher) model. The student model consisted of half the number of neurons in each layer of the neural network. It contained 21,187 parameters, with size of 110KB. However, this approach was only marginally successful, achieving a validation score of 83%. Since the target validation accuracy was 90%, this approach was not investigated further and instead, other compression techniques were investigated.

4.4.2 Pruning

The model was then compressed by magnitude-based weight pruning and fine-tuned on the training data over 10 epochs. The TensorFlow Model Optimization Toolkit was used for this purpose. The baseline model was the original LSTM network for processing dynamic gestures.

The pruning parameters were defined using *sparsity.keras.polynomialDecay*, which included setting the original and final sparsity. Following hyperparameter tuning, the optimal values were determined to be: original sparsity = 0.0, final sparsity = 0.6. Several iterations of hyperparameter tuning were conducted; including directly after the pruning phase and again after the post-pruning quantization phase. The optimal value for final sparsity was determined to be 0.5. This was determined to achieve the approximate accuracy limit of the model in this configuration, as extensive hyperparameter tuning did not identify superior alternatives.

Fine-tuning of the pruned model was conducted for 100 epochs, with only a slight performance degradation observed. For an original model which achieved 97% accuracy, the pruned version achieved 92% accuracy. The size of the model had not been reduced by the preceding steps. This is due to the operation of the pruning mechanism. It increases the sparsity of the model by setting a significant portion of the weights to zero, but not removing them. Specifically, the .h5 format stores these weights in a dense format. Additional metadata is also introduced in order to keep track of and record the pruning process.

TensorFlow strip_pruning was used to remove any metadata only required for training. The GZIP standard compression algorithm was applied to effectively compress the zero weights when storing the model. Following this step, significant reductions in model file size were observed 1. The gzipped pruned Keras model was 4.8 times smaller than the original Keras model.

4.5 Post-Pruning Quantization

The TensorFlow Lite converter was used to convert the pruned model into TensorFlow Lite format. The default optimization strategies were applied. These convert the weights, and potentially the activations in the pruned model to a lower precision according to the optimization strategy, e.g. 8-bit representation. This step is performed after training has been completed and is designed to make the model more efficient and suitable for deployment on resource-constrained devices. Following this step, a further significant reduction in model size (3.75 times) was observed over the size of the pruned model. The quantized and pruned model was evaluated on the test dataset and yielded a slight performance degradation, resulting in a final accuracy of 87.8%..

Model Description	Size (bytes)	Accuracy (%)
Size of gzipped baseline Keras model	774284.00	96
Size of gzipped pruned Keras model	159991.00	92
Size of gzipped pruned TFlite model	161470.00	92
Size of gzipped pruned and quantized TFlite model	43030.00	87.8

Table 1: Comparison of models

5 Limitations & Future Work

The primary limitation of this work was the limited size of the dataset available. Although the challenges posed were mitigated as effectively as possible and satisfactory real-world performance was demonstrated, more extensive evaluation is required to validate the solution. In order to further verify the robustness of the models, it would be beneficial to recruit a representative sample of participants to conduct a thorough evaluation on the devised solutions. Participants would interact with the real-time static gesture recognition model and pre-record videos of the dynamic gestures for processing. However, ethical approval may be required for this and due to time constraints and related limitations on project scope, this was not possible within the timeframe of this project. Although the author has conducted demonstrations of the systems in operation which indicate its generalizability, a more diverse and independent sample would be required for conclusive evaluation. In addition to expansion of the dataset, future work on this project could involve integration of the dynamic gesture recognition model with the static real-time gesture recognition model into a single, seamless,

real-time recognition system. Different architectures, in addition to LSTM, could be comparatively evaluated. Furthermore, such a system could be integrated into practical applications such as a mobile application which offers offline finger-spelling interpretation as an alternative for typing.

6 Discussion and Conclusions

The development and compression of the proposed machine learning models for Irish Sign Language translation yielded successful results, with potential for improvement and further research. The static sign gesture recognition model achieved high accuracy (97%) and efficiency with limited resources. This surpasses the previously-achieved benchmark accuracy score of 95% using Principal Component Analysis on the same dataset [7]. The dynamic gesture model also proved successful, achieving high accuracy scores (96%). Both solutions were also tested on new data from webcam footage of sign gestures, which were classified correctly. Due to resource constraints, the real-world testing was limited in scope but the initial results suggest that this research is applicable to practical applications.

Knowledge distillation of the dynamic gesture recognition model achieved a final accuracy score of 83%. This illustrated the inability of a simpler version of the LSTM network to capture adequate information for correct classification of video clips. A potential reason for this was the approach taken to establish the baseline implementation. The baseline implementations were iteratively made increasingly complex until a satisfactory level of accuracy was reached. Therefore, it seems reasonable that by reducing the model back to a previously-tested size, it is possible that it will not be sufficiently complex, even provided with a teacher model for training.

Magnitude-based weight pruning proved a more promising compression alternative, achieving accuracy of 92%. While this introduced a slight degradation in accuracy, it is still above the target threshold of 90%. Post-pruning quantization also proved to be an effective compression technique, further minimizing the resource requirements of the model to 43KB, although introducing a slight performance drop below the desired threshold (87.8%). The observation that hyperparameter tuning in the pruning phase did not result in significant fluctuations in accuracy, suggests that the dataset may not have been extensive enough for the reduced model to maximize information derivation. It is possible that if the dataset was suitably enlarged and diversified, the pruned model could attain a higher accuracy score which could in turn lead to improved results from the quantization phase. This is a potential avenue for future research. It is clear that low-precision values are sufficient for the model to maintain reasonably high accuracy. In addition, selective pruning of weights is a superior alternative to knowledge distillation to a smaller architecture for this task.

References

[1] Sunusi Bala Abdullahi and Kosin Chamnongthai. American sign language words recognition of skeletal videos using processed video driven multi-stacked deep lstm. *Sensors*, 22(4):1406, 2022.

[2] Ebey Abraham, Akshatha Nayak, and Ashna Iqbal. Real-time translation of indian sign language using lstm. In *2019 global conference for advancement in technology (GCAT)*, pages 1–5. IEEE, 2019.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[4] Ruth Holmes, Ellen Rushe, Mathieu De Coster, Maxim Bonnaerens, Shin'ichi Satoh, Akihiro Sugimoto, and Anthony Ventresque. From scarcity to understanding: Transfer learning for the extremely low resource irish sign language. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2008–2017, 2023.

[5] Md Abul Kalam, Md Nazrul Islam Mondal, and Boshir Ahmed. Rotation independent digit recognition in sign language. In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 1–5. IEEE, 2019.

[6] Lorraine Leeson. Moving heads and moving hands: Developing a digital corpus of irish sign language. *Ireland</i>,< i*, pages 25–26, 2006.

[7] Marlon Oliveira, Houssem Chatbri, Ylva Ferstl, Mohamed Farouk, Suzanne Little, Noel E O'Connor, and Alistair Sutherland. A dataset for irish sign language recognition. 2017.

[8] Irish Deaf Society. Irish sign language | irish deaf society, 2017.

[9] Su Yang and Qing Zhu. Continuous chinese sign language recognition with cnn-lstm. In *Ninth international conference on digital image processing (ICDIP 2017)*, volume 10420, pages 83–89. SPIE, 2017.

[10] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020.