Real-time Sign Language Detection

Nikhil Singh Thakur
Artificial Intelligence
Illinois Institute of Technology
nthakur4@hawk.iit.edu

Sanjana Rayarala
Artificial Intelligence
Illinois Institute of Technology
srayarala@hawk.iit.edu

PROBLEM STATEMENT

Our main aim in this research is to create a smart computer program that can quickly and accurately understand American Sign Language (ASL) when people use it in real life. ASL is a vital language for the Deaf and Hard-of-Hearing community in the U.S., and we believe that technology can make it even more accessible. Imagine a computer that can 'read' sign language on a live video and understand what's being said. This could be a game-changer for the Deaf and Hard-of-Hearing community, making communication easier and more inclusive.

Lot of difficulties could be posed in the process of identifying ASL signs from video footage, such as camera angles, changes in lighting, and hand gestures that can affect how the signs look. Moreover, the meaning of ASL signs can be greatly influenced by the context in which they are used, underscoring the significance of taking and the surrounding signs and background into account when categorizing individual signs. ASL sign recognition is made more complex when the use of body language and facial expressions is taken into consideration. Tackling these challenges, the model to be developed requires a large, diverse dataset that includes data of wide variety styles of signs, by many people.

Mitigating these challenges necessitates the utilization of an extensive and diversified dataset, encapsulating a spectrum of signers, signing styles, and environmental conditions for effective model training. The model should leverage state-of-the-art deep learning techniques, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to capture both spatial and temporal attributes within the video data.

The success of this research project hinges on the model's capacity to achieve high accuracy and rapid recognition of ASL signs in real-time. Prospective applications of this technology could conceivably be integrated into a myriad of virtual assistants, communication tools, and educational resources, ushering in a more inclusive and accessible era for the DHH community.

Notwithstanding the substantial strides made in ASL recognition in recent times, significant shortcomings persist in the precision and resilience of existing methodologies. Present models often falter in the face of ASL's intricacies, particularly in real-world settings characterized by ambient noise or idiosyncrasies in signers' styles.

This research initiative aspires to surmount these limitations by crafting a more sophisticated and robust model, characterized by its adaptability across different signers, signing styles, and environmental conditions. The proposed model will not only harness the spatial and temporal nuances of the video data but also discern the underlying context and meaning of signs within a broader linguistic context.

The potential ramifications of this research project transcend the realm of assistive technology for the DHH community. The capacity to interpret ASL signs from video data holds significance across multiple domains, including human-computer interaction, robotics, and surveillance. Furthermore, the insights gleaned from this research endeavor may provide the blueprint for analogous recognition models for other sign languages, thus extending the reach of this technology to Deaf communities worldwide.

KEYWORDS

Real-Time Recognition, Transfer Learning, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Classification model, Image Processing.

DATA SET

The Sign Language MNIST dataset serves as a drop-in replacement for MNIST, featuring a CSV format that includes labels and pixel values in single rows. It is tailored to represent the American Sign Language (ASL) alphabet, excluding letters J and Z due to their reliance on motion. With a total of 27,455 training cases and 7,172 test cases, this dataset maps each case to a label ranging from 0 to 25, corresponding to the alphabetic letters A to Z. Each case includes a 28x28 pixel grayscale image with values spanning from 0 to 255.

The creation of this dataset entailed a meticulous process to ensure its richness and diversity. The original data consisted of hand gesture images with multiple users performing gestures against various backgrounds. To augment the dataset's size and diversity, a comprehensive image processing pipeline was implemented. The process began with precise cropping, focusing on isolating the hand region to provide a close-up perspective of the gestural element.

Next, the images were converted to grayscale, ensuring uniformity and allowing algorithms to concentrate on shape and form rather than color. Subsequently, the images were resized to a standard 28x28 pixel resolution, aligning with the MNIST convention and ensuring compatibility with various machine learning methods, including Convolutional Neural Networks (CNNs).

A substantial portion of dataset expansion was achieved by creating over 50 variations of each image. These variations were introduced systematically to enrich the dataset, simulating real-world challenges. This included applying various image filters such as 'Mitchell,' 'Robidoux,' 'Catrom,' 'Spline,' and 'Hermite' to create distinctive visual effects. To emulate image distortion commonly seen in different lighting conditions or varying camera qualities, 5% random pixelation was introduced. Additionally, brightness and contrast adjustments within a ±15% range were made to introduce variations in image exposure. To account for different hand orientations during signing, rotations of up to 3 degrees were applied.

This deliberate dataset expansion strategy enhanced data diversity and complexity, providing real-world challenges for machine learning models. The Sign Language MNIST dataset offers a more challenging and practical counterpart to MNIST, suitable for a wide range of applications.
Inspiration:
The Sign Language MNIST dataset takes inspiration from the Fashion-MNIST and the machine learning pipeline for gestures by Sreehari.

Overall, developing a robust visual recognition algorithm using this dataset holds great promise. It not only challenges modern machine learning techniques, including Convolutional Neural Networks but also has the potential to significantly improve computer vision applications for the deaf and hard-of-hearing community. American Sign Language (ASL) is a complex and vital language, serving as the primary means of communication for many deaf North Americans. ASL ranks as the fifth most widely used language in the U.S., following Spanish, Italian, German, and French. By implementing computer vision on affordable hardware like the Raspberry Pi with OpenCV and Text-to-Speech, we can enable improved and automated translation applications.

COMPLETED WORK

The completion of this project has been marked by a series of substantial accomplishments in line with our project proposal's timeline. Here is an overview of the key milestones achieved thus far:

1. Reviewing Existing Research

Our project embarked with a comprehensive review of existing research on American Sign Language (ASL) recognition, where deep learning techniques played a pivotal role. We delved into a diverse array of topics, encompassing image and video processing, feature extraction, and classification algorithms. Recent strides in ASL recognition have been notably underpinned by deep learning methodologies, particularly leveraging the capabilities of convolutional neural networks (CNNs) and recurrent neural networks (RNNs). These advanced techniques have shown remarkable potential in elevating the accuracy and precision of ASL recognition systems.

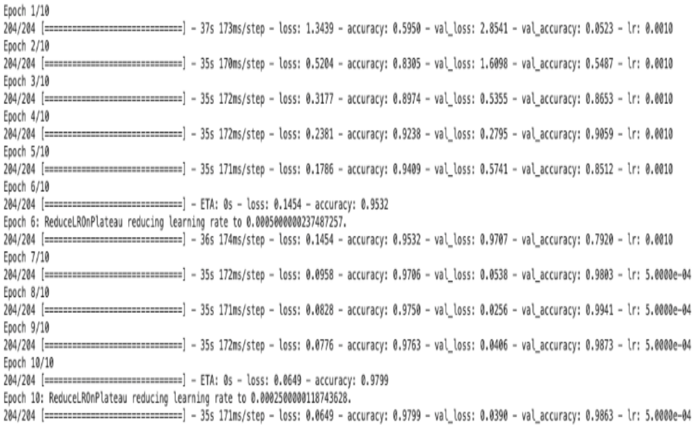
Our exploration extended to the study of diverse datasets developed for ASL recognition. Among these, the ASL Alphabet dataset stands out, as it houses images of hand gestures, each representing a letter of the ASL alphabet. Furthermore, we delved into datasets featuring video recordings, showcasing signers performing phrases or sentences in ASL. These datasets offered profound insights into the intricate temporal dynamics of sign language, shedding light on the efficacy of deep learning techniques in capturing the subtleties and nuances of ASL.

2. Dataset Collection and Pre-processing

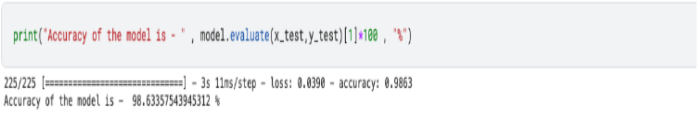
The foundation of our work was laid by selecting a diverse dataset comprising a substantial number of training and testing images. This dataset was meticulously curated to facilitate the development of our classification model. The diversity within the dataset aligns with our goal of building a robust and versatile model for ASL recognition.

3. Model Development and Analysis

A critical aspect of our project involved building an initial model to recognize American Sign Language (ASL) gestures. We diligently assessed the model's performance by rigorously tracking accuracy and validation loss for each training iteration. Achieving an impressive accuracy of 98.63%, we conducted our training across 10 epochs with a batch size of 128. This iterative analysis helped us monitor the model's progress and identify potential concerns, such as overfitting or underfitting. The ongoing refinement process was based on this detailed evaluation, ensuring the model's accuracy and readiness for practical applications in ASL recognition.



```
Epoch 1/10
204/204 [==============================] - 37s 173ms/step - loss: 1.3439 - accuracy: 0.5950 - val_loss: 2.8541 - val_accuracy: 0.0523 - lr: 0.0010
Epoch 2/10
204/204 [==============================] - 35s 170ms/step - loss: 0.5204 - accuracy: 0.8305 - val_loss: 1.6098 - val_accuracy: 0.5487 - lr: 0.0010
Epoch 3/10
204/204 [==============================] - 35s 172ms/step - loss: 0.3177 - accuracy: 0.8974 - val_loss: 0.5355 - val_accuracy: 0.8653 - lr: 0.0010
Epoch 4/10
204/204 [==============================] - 35s 172ms/step - loss: 0.2381 - accuracy: 0.9238 - val_loss: 0.2795 - val_accuracy: 0.9059 - lr: 0.0010
Epoch 5/10
204/204 [==============================] - 35s 171ms/step - loss: 0.1786 - accuracy: 0.9409 - val_loss: 0.5741 - val_accuracy: 0.8512 - lr: 0.0010
Epoch 6/10
204/204 [==============================] - ETA: 0s - loss: 0.1454 - accuracy: 0.9532
Epoch 6: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.
204/204 [==============================] - 36s 174ms/step - loss: 0.1454 - accuracy: 0.9532 - val_loss: 0.9707 - val_accuracy: 0.7920 - lr: 0.0010
Epoch 7/10
204/204 [==============================] - 35s 172ms/step - loss: 0.0958 - accuracy: 0.9706 - val_loss: 0.0538 - val_accuracy: 0.9803 - lr: 5.0000e-04
Epoch 8/10
204/204 [==============================] - 35s 171ms/step - loss: 0.0828 - accuracy: 0.9750 - val_loss: 0.0256 - val_accuracy: 0.9941 - lr: 5.0000e-04
Epoch 9/10
204/204 [==============================] - 35s 172ms/step - loss: 0.0776 - accuracy: 0.9763 - val_loss: 0.0406 - val_accuracy: 0.9873 - lr: 5.0000e-04
Epoch 10/10
204/204 [==============================] - ETA: 0s - loss: 0.0649 - accuracy: 0.9799
Epoch 10: ReduceLROnPlateau reducing learning rate to 0.0002500000118743628.
204/204 [==============================] - 35s 171ms/step - loss: 0.0649 - accuracy: 0.9799 - val_loss: 0.0390 - val_accuracy: 0.9863 - lr: 5.0000e-04
```

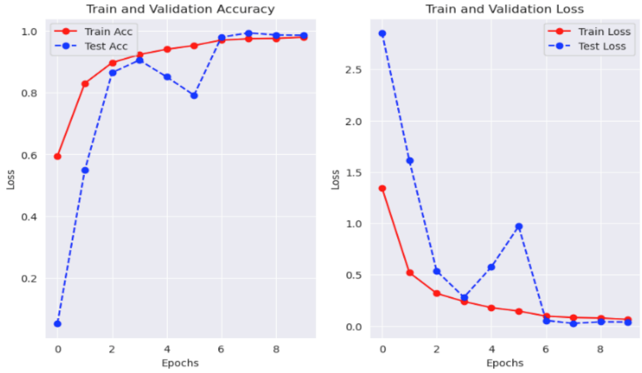


```
print("Accuracy of the model is - " , model.evaluate(x_test,y_test)[1]*100 , "%")
```

```
225/225 [==============================] - 3s 11ms/step - loss: 0.0390 - accuracy: 0.9863
Accuracy of the model is -  98.63357543945312 %
```

4. Model Evaluation

The primary objective was to achieve high accuracy and low loss for both the training and validation sets, signifying the model's capacity to generalize effectively to new examples. Discrepancies between training and validation metrics, particularly when training metrics significantly outperformed validation metrics, were indicative of potential overfitting. In response, we implemented regularization techniques, including dropout and weight decay, to prevent overfitting. The in-depth analysis of training and validation metrics played a pivotal role in model optimization. This iterative assessment not only aids in identifying potential issues but also provides insights for fine-tuning the model to optimize its performance and generalization capabilities.



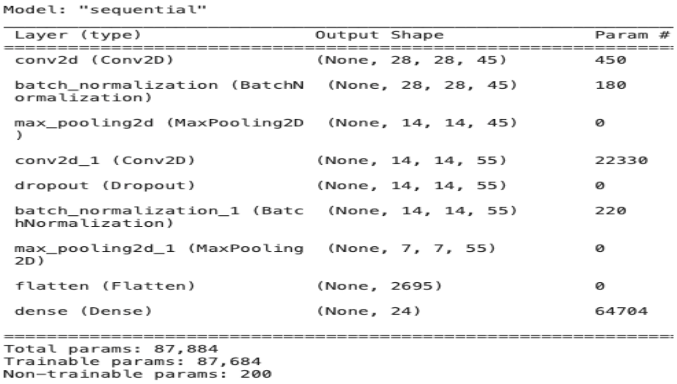
The above image shows the training accuracy, validation accuracy, training loss and validation loss of the classification model that we implemented.

5. Testing Dataset Validation

To ensure the model's correctness and robustness, we subjected it to rigorous testing using a dedicated dataset designed for validation.

6. Model Summary

Below, we present a concise yet comprehensive model summary that encapsulates the architectural and operational characteristics of the Real time sign language detection recognition model. This summary serves as a reference point for the model's structure and configuration.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 45)        450

 batch_normalization (BatchN  (None, 28, 28, 45)       180
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 14, 14, 45)       0
 )

 conv2d_1 (Conv2D)           (None, 14, 14, 55)        22330

 dropout (Dropout)           (None, 14, 14, 55)        0

 batch_normalization_1 (Batc  (None, 14, 14, 55)       220
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 7, 7, 55)         0
 2D)

 flatten (Flatten)           (None, 2695)              0

 dense (Dense)               (None, 24)                64704

=================================================================
Total params: 87,884
Trainable params: 87,684
Non-trainable params: 200
```

In essence, the completion of these tasks represents significant progress in our project, moving us closer to the ultimate goal of developing a high-performance Real time sign language recognition model with practical applications in mind. The model, born from these endeavors, holds the potential to revolutionize communication accessibility for the Deaf and Hard-of-Hearing community, offering a valuable tool for bridging language barriers and enhancing inclusivity.

NEXT STEPS

Real-Time Input Integration - Initially we have used a pre-determined testing dataset in order to measure the accuracy of the model. Although this approach gives us almost 98.63% accuracy, the input given to the model is not a real-time input. Therefore, one of the most important update to the project would be to accept real time video input and classify the hand gestures into the alphabets of the American Sign Language.

This critical step involves the adaptation of our ASL recognition system to accept real-time video input. Unlike the initial phase, which relied on a predefined testing dataset, real-time input processing means that the model can analyze and classify ASL hand gestures as they occur during live video capture. This transition is pivotal as it mirrors real-world applications where individuals use sign language in real-time conversations. The model will continuously process video frames, providing immediate and accurate translations of ASL gestures, thus significantly improving accessibility and communication for the Deaf and Hard-of-Hearing community.

To further fine-tune our model and enhance its accuracy, we will explore the hyperparameter optimization. These parameters, including learning rate, batch size, and the number of training epochs, play a crucial role in a model's performance. By meticulously optimizing these values, we aim to boost the model's overall accuracy.

Diversification in our approach is on the horizon. We will explore into various techniques for creating a classification model for ASL recognition. This exploration includes Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Graph Neural Networks. Through comprehensive analysis and comparative studies, we will evaluate the strengths and weaknesses of each network architecture to determine the best-suited approach for ASL recognition.

Another Important step in this project is to try facial expressions and body movements. Recognizing their pivotal role in sign language, we will work on incorporating these elements into our model. This will enable a more comprehensive understanding of ASL and significantly enhance the accuracy of sign recognition. Two approaches will be explored: a multimodal framework that combines video and audio data to capture spatial and temporal features, and an attention mechanism that focuses on relevant regions like the face and body while ignoring irrelevant areas. These inclusions will reduce misinterpretations of signs and improve the robustness of the model.

We recognize that people have different ways of using sign language and that it can vary based on where and how it's used. To tackle this challenge, we'll tap into the strengths of extra elements. By paying attention to things like facial expressions and body movements, we'll help the model understand sign language better across various signers and situations. This way, our model will stay accurate and reliable, no matter who's signing or the setting in which they're doing it. It's like giving the model a broader view, so it can handle all the differences and still get things right.

In summary, our next steps are designed to transform ASL recognition into a real-time, practical tool, with optimized performance, diverse model architectures, and the incorporation of facial expressions and body movements. These efforts will lead to an improved, accurate, and robust ASL recognition model, making it more comprehensive and inclusive in representing this vital language. This ambitious roadmap will pave the way for broader applications in assistive technology, communication, and accessibility. The Deaf and Hard-of-Hearing community stands to gain immensely from these developments, bridging the language gap and fostering inclusivity.

REFERENCES

[1] Computer Vision: Algorithms and Applications" by Richard Szeliski: It covers computer vision fundamentals, which are essential for real-time image and video analysis in sign language detection systems.

[2] Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville: This book is great to understand deep learning techniques, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), which are crucial for image and sequence processing in sign language recognition.

[3] Hands-On Computer Vision with OpenCV and Python" by Riaz Munshi, Steven L. Palmer, and Kemal Tugrul Sandeep Mudigonda:

[4] Multimodal Interaction in Image and Video Applications" edited by Cha Zhang and Zhengyou Zhang

[5] Sign Language Recognition, Translation, and Production" edited by Rami Abielmona and Onur Tuncer: This book is specifically focused on sign language recognition and can provide insights into the challenges and solutions in this field.

[6] Real-Time Sign Language Recognition from Video: A Comprehensive Overview" by Tanzeem Syeda Anjum: This research paper provides an in-depth exploration of real-time sign language recognition, making it a valuable reference for your project.

[7] Real-Time Sign Language Recognition Using a Range Camera. Authors: Ariya Rastrow, Jin Z. Zhang Published in: Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008
Link: https://ieeexplore.ieee.org/document/4563097