Towards a Reliable Machine Learning-based Model Designed for Translating Sign Language Videos to Text

Maitha Essa Mohammad Ahli
mea9722@rit.edu

Follow this and additional works at: https://scholarworks.rit.edu/theses

Recommended Citation

This Master's Project is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
TOWARDS A RELIABLE MACHINE LEARNING-BASED MODEL DESIGNED FOR TRANSLATING SIGN LANGUAGE VIDEOS TO TEXT

by

Maitha Essa Mohammad Ahli

A Capstone Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Science in Professional Studies: Data Analytics

Department of Graduate Programs & Research

Rochester Institute of Technology
RIT Dubai
May 10, 2023
Master of Science in Professional Studies:
Data Analytics

Graduate Capstone Approval
Student Name: Maitha Essa Mohammad Ahli
Graduate Capstone Title: Towards a Reliable Machine Learning Based Model Designed for Translating Sign Language Videos to Text

Graduate Capstone Committee:

Name: Dr. Sanjay Modak
Chair of committee

Name: Dr. Khalil Al Hussaeni
Member of committee
ACKNOWLEDGMENTS

I would like to take this opportunity to express my deepest gratitude to everyone who has supported me throughout my data analytics studies. Firstly, I want to thank my husband for his unwavering support, encouragement, and patience throughout the entire duration of this journey. His constant love and support helped me stay motivated and focused, and I could not have done this without him. I would like to express my gratitude to my professors, and especially Dr. Khalil Al Hussaeni, who provided me with the knowledge, skills, and guidance necessary to complete my studies. Their expertise and feedback were invaluable, and I am grateful for the time and effort they invested in me. I would also like to extend my sincere thanks to my family and friends for their support and understanding during this time. Your words of encouragement and belief in me gave me the confidence to tackle this project and complete it successfully within a restricted time frame. Special thanks go to my group members, "The Outliers," for their hard work, dedication, and collaboration throughout this entire time. Their contributions and insights were invaluable, and I am grateful for the time and effort they all invested in this academic journey. Once again, thank you to everyone who supported me during this project. Your support and encouragement have been instrumental in helping me achieve this milestone in my academic journey.
ABSTRACT

Communication serves key roles in building relationships through sharing feelings, passing information, and connecting with others. Communication among the hearing impaired remains a significant stumbling block in today’s society since their communication means demands for an interpreter each moment. Various researchers agree that successful communication calls for the involvement of all individuals in a conversation and thus, deaf and hearing-impaired people require precise and welcoming communication to promote their working and learning relationships. Sign Language Recognition (SLR) is a critical and auspicious approach to promoting communication among hearing-impaired people. Sign languages greatly benefit from Machine Learning based translation techniques since they are authentic natural languages characterized by several grammatical requirements and wordbooks. Similarly, sign language may benefit from the computer vision approaches of encoding because it is a visual-spatial language. In the recent past, the advent of machine learning techniques has greatly contributed to significant advances in computer vision approaches and natural language processing. Such advances have motivated the researchers to make efforts on extending the learning techniques to promote understanding of sign language. However, sign language interpretation remains a significant challenge since it entails an unending visual-spatial modality where context helps in deriving the meaning. The proposed project seeks to leverage the benefits of emerging technologies to find an effective and reliable system for recognizing gestures. The project aims at designing a machine learning-based model for recognizing sign language to automatically transcribe sign language videos into text. The work suggests a novel model that utilizes video sequences comprising spatial and temporal features. The study will use different Machine Learning algorithms to train the extracted features from the given
datasets. An input video will be used to do sign language translation with the help of the machine learning models and the sign displayed in the video is detected and changed to text. The deep learning approach used had an accuracy of 80.01% on the testing set and 86.94% on the validation set. Using K=29, our k-nearest neighbor model performed with an accuracy of 88.02%. On the validation set, an accuracy of 82.50% was achieved using decision tree classifier. Also an accuracy of 76.86% was achieved using the boosted decision tree approach; similarly an accuracy of 85.00% was achieved using support vector classifier.

**Keywords:** Sign Language Recognition (SLR) sign languages Machine Learning, computer vision, transcribe.
# TABLE OF CONTENTS

ACKNOWLEDGMENTS .................................................................................................................. 3
ABSTRACT .................................................................................................................................. 4
LIST OF FIGURES .......................................................................................................................... 7
LIST OF TABLES ............................................................................................................................ 7
CHAPTER 1 ................................................................................................................................... 8
  1.1 BACKGROUND INFORMATION ......................................................................................... 8
  1.2 STATEMENT OF THE PROBLEM ..................................................................................... 8
  1.3 PROJECT DEFINITION AND GOALS .............................................................................. 9
  1.4 METHODOLOGY ................................................................................................................ 10
  1.5 LIMITATIONS OF THE STUDY ....................................................................................... 11
CHAPTER 2 - LITERATURE REVIEW ......................................................................................... 12
  2.1 MAIN TAKEAWAYS .......................................................................................................... 17
CHAPTER 3 – PROJECT DESCRIPTION ..................................................................................... 18
  3.1 DATA COLLECTION .......................................................................................................... 18
  3.2 DATA CLEANING AND PREPROCESSING ..................................................................... 21
CHAPTER 4 – PROJECT ANALYSIS .......................................................................................... 24
  4.1 FEATURE EXTRACTION .................................................................................................... 24
  4.2 METHODS OF DATA ANALYSIS ...................................................................................... 25
    4.2.1 K-NEAREST NEIGHBORS (K-NN): ........................................................................... 26
    4.2.2 DECISION TREE: ................................................................................................... 26
    4.2.3 BOOSTED DECISION TREES ............................................................................... 26
    4.2.3 SUPPORT VECTOR MACHINE: .............................................................................. 27
  4.3 CROSS-VALIDATION ......................................................................................................... ERROR! BOOKMARK NOT DEFINED.
  4.4 MODEL EVALUATION METRICS ..................................................................................... 27
  4.5 SOLUTION FLOWCHART .................................................................................................. 27
  4.5 DATA ANALYSIS RESULTS ............................................................................................. 30
  4.6 COMPARISON .................................................................................................................... 34
CHAPTER 5 ................................................................................................................................ 35
  5.2 RECOMMENDATIONS AND FUTURE RESEARCH .......................................................... 35
BIBLIOGRAPHY .......................................................................................................................... 36
LIST OF FIGURES

Figure 3.1.1: A visual representation of the number of signed videos per signer ....................... 19
Figure 0.1.2: A visual representation of variation in Video background and luminance ............. 20
Figure 0.2.1: A graph of frames from a sample video decoded .............................................. 22
Figure 3.2.2: A graph of frames after sampling frames from the video .................................... 22
Figure 3.2.3: A graph that represents the distribution of videos across train, test, and validation sets ... 23
Figure 4.2.1: A graph of the time distributed layers of the model .............................................. 25
Figure 4.5.1: Solution Flowchart ................................................................................................. 29
Figure 4.6.1: A visual representation of the model training history ............................................ 30
Figure 4.5.2: Optimal number of Neighbourhoods: .................................................................... 31

LIST OF TABLES

Table 4.5.1: Model performance ................................................................................................. 31
Table 4.5.2: Hyper parameter Tuning for the decision tree model ............................................ 32
Table 4.5.3: Optimal combination of hyper parameters for Boosted regression approach ........ 33
Table 4.5.4: Optimal parameter combination for Support vector classifier .............................. 33
Table 4.5.5: Performance evaluation metrics for DT, SVM, BoostedDT and KNN .................... 34
CHAPTER 1

1.1 Background Information

Communication plays a key role in building relationships within any group of people. It enables the passing of information from one individual to another as well as the expression of one's feelings, experiences, and thoughts. Good communication uses techniques that ensure the inclusion of every individual regardless of their diverse backgrounds, colour, physical conditions, or mental conditions. There are different forms of communication used by different people. For instance, sign language is a method of interaction that leverages gestures and signs to pass information from one individual to another. The method used by deaf and hearing-impaired patients is to relay information between themselves and normal people. The gestures and signs help this group of people to express themselves, and each of the symbols represents a particular emotion, letter, or word that needs a translator to facilitate their interaction with those who do not know sign language. However, some translators may misinterpret the gestures and signs hence resulting in misunderstandings. Additionally, their communication is not always accompanied by a translator, leading to misunderstanding or miscommunication. In this regard, there is a need for an automatic system for converting sign language gestures and signs into text to facilitate the interaction between hearing-impaired people with those who do not understand sign language.

1.2 Statement of the problem

Sign language is a method used by hearing-impaired people to interact with others using gestures and signs. They entirely rely on sign language to express their ideas, thoughts, experiences, and feelings which creates the need for a translator to facilitate communication
between deaf people and normal people. However, their communication is not always accompanied by translators resulting in misunderstanding and miscommunication. In this regard, there is a need for an automatic system that can transcribe sign language videos into text without the help of human beings. Emerging technologies such as machine learning are seen as the immediate solution to the identified problem. Machine learning techniques are critical in designing and developing an automatic model for transcribing sign language videos into text. The proposed study seeks to develop a machine learning-based model to facilitate communication among muted and hearing-impaired persons and those who do not know sign language by developing an ML-based model that converts sign language videos to text. Converting sign language to text is essential in facilitating interaction among people with hearing disabilities and those who do not know sign language without any misunderstanding or miscommunication.

1.3 Project Definition and Goals

The proposed project deals with the translation of sign language videos into text. Technology is critical in facilitating communication among muted and hearing-impaired individuals. The overarching aim of the proposed project is to determine the various approaches used in converting sign language into text among disabled people, particularly the deaf and hearing-impaired. Besides, the project seeks to determine the various innovators and creators exploited in recognizing sign language and converting sign language into text. Finally, the project aims to determine the most efficient, accurate, and reliable model for converting sign language to text.
1.4 Methodology

The proposed study would make use of machine learning techniques to provide a more dependable and effective framework for text conversion from sign language videos. A dataset of videos in sign language is needed for the study's analysis. Therefore, the dataset will be obtained from an online database known as Kaggle. The dataset contains over 2,000 words from more than 100 different signers, and it is publicly accessible. The data will be cleaned and pre-processed by eliminating the duplicate, incomplete, and wrongly formatted data. This is to ensure that the data is error-free and avoid any unreliable results and model. Since each video sequence contains both spatial features derived from video frames and temporal features derived when the video frames are linked with time, categorizing videos continues to be a substantial difficulty. The proposed system will be trained on all types of features using a variety of machine learning networks, according to the study. The framework will be trained on spatial information using a CNN model and on temporal features using an RNN model. The main model will be compared with different classification models as K-Nearest Neighbors, Decision Tree, Boosted Decision Tree, and Support Vector Machine, the suggested framework will be compared in the study. Time complexity and classification accuracy will serve as the evaluation metrics. The next step will be model tuning, which is important to optimize hyperparameters and improve the predictive accuracy of the model. Finally, the model will be evaluated by comparing it with the other algorithms in order to select the most reliable and efficient model. By transcribing sign language videos into text, the suggested framework aims to increase communication between the deaf community and people who do not understand sign language. This will help to prevent misunderstandings and misunderstandings.
1.5 Limitations of the Study

One major limitation of this study was the limited computation resources available. The processing of thousands of videos with high frame rates demanded a lot of RAM and long training times. Another limitation was that the data collected consisted of only single words per video, limiting the model's ability to predict multiple words in a sentence. To address this limitation, more infrastructure would be needed to split videos with multiple words or signs into single-word videos.

In contrast to spoken and written language, sign language does not have a direct connection between the patterns of characters or sounds and the label. Therefore, it is not possible to use connectionist temporal classification (CTC) layers to recognize new words. Instead, each sign or gesture is treated as a separate category, resulting in a limited vocabulary of only 2000 words.
CHAPTER 2 - LITERATURE REVIEW

Various researchers, engineers, and scientists have been on the front line in carrying out different types of studies on sign language recognition as well as human action. However, most of the researchers have given much attention to vision-based techniques since it is efficient and cheaper compared to conduct based-techniques. This study seeks to examine the relevant existing studies and conduct a critical analysis of the suggested or tested techniques in the aforementioned works to aid in establishing and explaining the particular research gap, which is the main focus of the current study. Brour & Benabbou (2021) found out that various solutions including hearing aids have been greatly considered in the long run in an attempt to bridge the gap between persons with hearing disorders and the hearing community. The vision-based and conduct-based techniques have continued to gain popularity in developing human action and sign language recognition techniques (Abraham & Rohini, 2018). For instance, the study by Bantupalli & Xie, (2019) exploited computer vision techniques and the ASL datasets to obtain spatial and temporal sequences using machine learning techniques. The study utilized Convolutional Neural Network (CNN) to extract spatial features from the video while the Recurrent Neural Network (RNN) for extracting the temporal features from the video. The model attained a high data training accuracy. For instance, Kartik et al, (2020) attained a 98.67% accuracy from a test size of 8,700 images extracted from a dataset comprising 87,002 which were trained on 78,300 images. Irrespective of the high accuracy of the computer vision models, it remains an expensive technique since it relies on cumbersome and costly hardware which demands a computer and thus makes it inefficient for use.

In a recent study, Ko et al., (2019) presented a sequence-to-sequence approach that entirely relies on the human key point approach to translate sign language to text. The study exploited
14,672 high-quality and resolution videos in developing a KETI sign language corpus comprising of the training set (64%), test set (29%), and development set (7%). The GRU cells-based model attained a 55.28% accuracy, a 63.53% ROUGE score, and a 52.63% BLEU score on the gloss level. Wangchuk et al. (2021) exploited a CNN technique to design Bhutanese sign language digits using approximately 20,000 sign images. The model aimed to recognize the Bhutanese sign language of 10 different static signs extracted from various individuals. The study compared the proposed CNN algorithm with different sign languages where the proposed model achieved a 97.62% testing accuracy and a 99.94% training accuracy. The study found that the testing and misclassification accuracies entirely relied on the sum of images in the data set after assessing the model’s F1 score, precision, and recall. The transfer learning techniques including VGG16, ResNet, and MobileNet could have been exploited for further modification of the accuracy. The authors suggested future research to expound on the dynamic alphabets and gestures of the Bhutanese sign language.

The work of Sharma & Anand (2021) presented a vision-based framework for recognizing sign language to automatically translate the signs of Arabic sign language to text. The translation model entailed four major steps including feature extraction, data classification, tracking, and the use of a skin detector for hand segmentation. The experimental outcomes attained a signer-independent recognition rate of 97% which is higher than the pre-existing approaches in terms of accuracy. Deep learning CNN has witnessed extensive popularity in data science studies, such as the use of vision-based recognition of hand gestures for translating sign language using deep learning. The authors further introduced a CNN-based approach for recognizing sign language to convert Indian sign language to text. The study was the Indian sign language’s initial comprehensive evaluation. The absolute initial ML state was used to train a three-layered CNN
algorithm and achieved a 99.8% recognition accuracy on the numerals and a 97.8% accuracy on the alphabets of the Indian sign language. The authors recommended a future study to expound on exploiting a comparative analysis for choosing the most effective model for recognizing sign language (Sharma & Anand, 2021).

Various researchers consider vision-based approaches as a cost-effective, efficient, and reliable sign language recognition model compared to other models (Dhulipala et al., 2022), (Sharma & Anand (2021). The model uses a camera mounted on a computer like a webcam to capture gestures and signs for decoding with the help of machine learning algorithms. Many researchers, engineers, and scientists have conducted studies on different vision-based approaches to determine the most effective and reliable predictive approach. The authors suggested a vision-based model for real-time sign language recognition to help in translating the southern Indian language into text (Sruthi & Lijiya (2019) whereas the authors Marjusalinah et al. (2021) developed a finger spelling approach. The work of Elsayed & Fathy (2020) also exploited Deep learning and Ontology to develop a Semantic Translation Framework for sign language. The authors employed a CNN-trained system for gesture and sign recognition by adding the semantic layer. The authors trained and tested the framework using 10 Arabic gesture signs and their French and English meanings.

In another study, (Deebak et al., 2021) suggested a deep learning based system for recognizing Indian Sign Language via gesture recognition. The study used a retraining VGG16 model based on CNN's classification method to identify the kinds of characteristics from an input picture and categorize them. The suggested system was able to recognize 92.8% of the 5,500 photos in the dataset. Additionally, authors have recommended the use of knowledge transfer techniques to improve the model's accuracy.
A study by (Yu et al., 2018) has suggested a multimodal fusion-based vision-based recognition system for Chinese sign language. The technology combined the identification of facial expressions and hand motions using deep learning techniques. The suggested system has 89.6% identification accuracy for 5,000 gestures in Chinese sign language. The authors have further suggested using a larger dataset to further enhance this model.

According to (Jiang et al., 2020). Based on video and accelerometer data, a multimodal methodology has been proposed for the identification of American Sign Language. A deep learning-based linguistic classifier was used in this work to categorize the hand gestures used in sign language. The proposed approach has a 100-sign accuracy of 94.2% on a dataset of 15,000 samples. (Abbasi et al., 2008) suggested that a range of sensors and modalities be used to conduct multimodal techniques for Sign Language interpretation in order to learn more about them.

The research by (Pham et al., 2020) included a proposal for a sign language recognition system employing wearable armbands and motion sensors. A Deep Learning-based classifier was used in this work to identify the sign language motions. The proposed method has a 95.5% accuracy recognition rate for a database of 3,000 American Sign Language. (Wang et al., 2003) advocated conducting additional study on wearable devices for sign language detection in order to increase accessibility for hearing individuals.

According to (Khan et al., 2021), A convolutional neural network-based deep learning system has been presented for the Bangla language to recognize sign language. To boost the model's functionality, the study has incorporated transfer learning techniques. A dataset of 900 motions in Bangla Sign Language was used to test the suggested system, and its recognition
accuracy was 94.5%. More research should be done on transfer teaching techniques for sign recognition, according to the authors, in order to increase the efficacy of this model.

In a research by (Jiang et al., 2021), a system that integrates computer vision and natural language processing through a combination of two methodologies has been proposed for the recognition of sign languages. A deep learning-based classifier was employed in the study to identify the sign language motions, and a model for natural language processing was utilized to create text using the detected gestures. The suggested system's recognition accuracy for a dataset of 100 American Sign Language sign motions was 93.4%.

(Li et al., 2022). A wearable glove equipped with sensors to record hand gestures has been proposed as part of a system to understand sign languages. The study employed a Deep Learning-based classifier to identify sign language gestures. The recommended technique produced an estimated recognition accuracy of 94.5% in a database of 4,000 Chinese sign language motions. The authors suggested more research into Wearable Technology in Sign Language Recognition to improve accessibility for those with hearing impairment.

As demonstrated in the work of (Li et al., 2020), a multimodal strategy incorporating computer vision and audio characteristics has been proposed for a sign language recognition system. The study employed a voice recognition model to create text from the detected sign language motions and a deep learning-based classifier to identify the sign language gestures. The suggested system was able to recognize 500 gestures in China Sign Language with a recognition accuracy of 93.7%.
2.1 Main takeaways

Evaluation of the existing literature shows that:

- Most of the existing works are based on prejudiced assumptions and lacked efficient methodologies for determining the most reliable and effective sign language translation system.

- Future research focusing on selecting a more reliable, cost-effective, and efficient system for translating sign language is recommended since the existing studies are characterized by low-reliability factors.

- Most of the existing works presented sign language translation models based on images, and thus, there is a need for future research focused on developing a translation model using videos.
CHAPTER 3 – PROJECT DESCRIPTION

This project aims to create a machine learning model that can correctly translate videos of American Sign Language (ASL) into text. This initiative is particularly significant since it addresses the communication challenges that people who are deaf or hard of hearing encounter on a regular basis. It is possible to close the communication gap and create a more welcoming environment for the deaf and hard of hearing people by creating a model that can automatically translate ASL videos into text. The study requires a dataset for sign language videos for analysis. This objective is accomplished by selecting a dataset of ASL videos and the matching English text transcripts. Using this dataset, a deep learning model will be trained to recognize ASL signs' gestures and movements and translate them into text. The English text representing the meaning of the ASL signs will be the model's output. The project will combine objective criteria such as accuracy, sensitivity, and F1-score to assess the performance of the model. To show the effectiveness of the model, it will also be compared with other models. Overall, by giving the deaf and hard of hearing community a more accessible form of communication, this project has the potential to have a tremendous impact on their lives.

3.1 Data Collection

The study uses a secondary dataset called Word-Level American Sign Language (WLASL) dataset, which is available on the Kaggle repository. The dataset consists of 11,980 sign videos, signed by a total of 121 signers. Figure 3.1.1 visually represents the number of signed videos per signer. Each video features a single word, and there are a total of 2000 unique words. The videos differ in terms of signers, background, and length. Figure 3.1.2 represents the variation in video background and luminance. The frame frequency in each video is 25 frames per second.
Additionally, the data is readily available for public access through the following link: WLASL (World Level American Sign Language) Video | Kaggle. The new large-scale dataset will enable the analysis with deep learning techniques for recognizing world-level signs and assessing their performances in wide-ranging instances.

Figure 0.1.1: A visual representation of the number of signed videos per signer
Figure 0.1.2: A visual representation of variation in Video background and luminance.
3.2 Data Cleaning and Preprocessing

Data cleaning involves fixing or eliminating the duplicate, incomplete, corrupted, incorrect, and wrongly formatted data in a given dataset. Compilation of many data sources is usually characterized by data mislabelling or duplication and thus, data cleaning and processing are essential in attaining error-free data. In this study there was missing videos whose IDs were captured on data glossary, the data was cleaned by dropping videos, video IDs corresponding to such videos.

The retrieved data was processed using a TensorFlow pipeline to convert it into a form that can be used for modeling. The first processing step was to read the videos and decode the frames into a standard size of 244 x 244. Due to the high number of frames per second, there were many similar frames representing consecutive time points, which is computationally intensive. To address this issue, the frames were sampled by skipping every other 4 frames. Figure 3.2.1 below visualizes frames on a sampled video from the training data, and Figure 3.2.2 visualizes the same dataset after sampling the frames.

TensorFlow performance pipeline operations, including batching, were used to put images into batches of 10 videos. This means that the pipeline processed 10 videos per time during model training. Because the videos had different lengths, blank frames were padded at the end of short videos to match the longest video. This was necessary because batching does not allow varying video lengths to be put in the same batch. Prefetching was used to overlap preprocessing and training to ensure that videos were processed before the training processes requested them for model fitting. Caching was used to store video frames of already processed videos in memory to ensure that preprocessing was not repeated every time the videos were required for training.
Figure 0.2.1: A graph of frames from a sample video decoded

Figure 3.2.2: A graph of frames after sampling frames from the video.
The published data exists in three main splits, including a training set which consists of 69.4% of the data and serves the purpose of learning the models, a validation set with 18.8% of the data, and a testing set with 11.8% of the data, where the models are tested and validated. Figure 3.2.3 below visualizes this information.

Figure 3.2.3: A graph that represents the distribution of videos across train, test, and validation sets.
CHAPTER 4 – PROJECT ANALYSIS

4.1 Feature extraction

Over the past few years, Convolutional Neural Networks (CNNs) have gained immense popularity as feature extractors for text, image, and video datasets. Researchers have utilized various types of CNN layers to improve the performance of these networks. In addition, many pre-trained models have been developed which utilize these CNN layers and have shown to produce better results. Some of the most commonly used pre-trained models include VGG16, ResNet50, EfficientNet, and InceptionV3, among others. These pre-trained models provide a powerful and efficient way to utilize CNNs for a variety of tasks, including image classification, object detection, and video analysis. By using these models, researchers can save significant amounts of time and computational resources while achieving state-of-the-art performance on their tasks.

In this study, a CNN-RNN model will be used to extract features from the video frames, the features extracted are matrices of pixel values representing the Red blue and green colors scales, these are expected to change across frames on the same video representing visual signs corresponding to different words. The learned features are passed to a time distributed LSTM layers which learn the temporal patterns from the videos. The learned temporal patterns will then be passed to dense Artificial layers for the final classification. Figure 4.2.1 below visualizes the time distributed layers of the model.
4.2 Methods of Data Analysis

Video categorization remains a significant challenge since each video sequence entails spatial features obtained from video frames and temporal features obtained when the video frames are linked with time. The study exploited various machine learning networks to train the proposed system on every sort of feature. A CNN model will be used for training the framework on spatial features while an RNN model for training the framework on temporal features. A dense neural network was placed at the end of the model to emit the classifications.

Figure 4.2.1: A graph of the time distributed layers of the model.
The study will compare the proposed framework with the other classification models such as K-Nearest Neighbors, Decision Tree, Boosted Decision Tree, and Support Vector Machine. The evaluation metric will be accuracy, Sensitivity, and F-Score.

### 4.2.1 K-Nearest Neighbors (k-NN):

k-NN is a machine learning algorithm that works on the basis that there is a similarity between the input record and its neighbours, and therefore, its class will as well be the same as the neighbour’s class. The k-NN algorithm calculates the Euclidean distance to every training sample to enable the selection of the k-nearest samples. The test sample class is finally obtained from the class corresponding to the highest sample numbers in the k-nearest neighbours.

### 4.2.2 Decision Tree:

A decision tree classifier is a Machine learning technique that introduces a decision tree where the single internal node represents an attribute test in that the test outcomes determine the branch to be selected and every leaf corresponds to the data classification label. The resulting decision tree is then utilized to categorize the test record into its proportionate leaf (class) assisted by several tests in the tree nodes. The functions used to create decision tree classifiers include; entropy, which determines the dataset homogeneity, and information gain responsible for determining the entropy change after splitting the datasets into a particular attribute. The decision tree classifier utilizes an attribute with the most information gain as the node where dataset splitting occurs and the step is repeated to a point where no further splitting can occur to create a leaf node with the corresponding class.

### 4.2.3 Boosted Decision Trees:

This is the process of compiling several weak learners to achieve better predictive outcomes. The model entails inserting new decision trees aimed at providing the previous subsequent tree misclassifications with additional weight (Wu et al., 2020)
and the process is repeated until no further improvement on the tree's performance can be performed.

4.2.3 Support Vector Machine: This is a machine learning technique used for solving both regression and classification issues. It exploits the kernel functions in mapping the data to a higher dimensional space with the aim of classification (Sun et al., 2019). The key roles of the SVM involve RBF, linear as well as polynomial kernel.

4.3 Cross-Validation

Model performance remains a critical aspect as far as accuracy is concerned. In this regard, the framework will be evaluated based on performance to determine if it’s suitable. The model performance is compared with other models to determine the most effective and reliable recognition model.

4.4 Model Evaluation Metrics

Accuracy score in classification is the rate at which the model accurately predicts the class labels of the data. It is calculated by dividing the sum of true positive and true negative predictions by the total number of predictions made by the model, as shown below:

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP}
\]

Sensitivity is capability of the model to accurately identify counterfeit news articles and it is determined by the ratio of correctly identified fake news articles to the total number of fake news articles in the evaluation set.

\[
\text{Sensitivity} = \frac{TP}{TP + FN}
\]
In cases where the dataset is unbalanced, the **F-score** is a useful evaluation metric as it represents a trade-off between recall and precision. The F-score is the harmonic mean of recall and precision and is computed as shown below:

\[
F \text{ score} = 2 \times \frac{\text{Precision} \times \text{sensitivity}}{\text{Precision} + \text{sensitivity}}.
\]

### 4.5 Solution Flowchart

The figure 4.5.1 below represents the solution flowchart starting from reading and decoding the videos into frames until evaluating the CNN-RNN model and comparing it with the other classification models.
Figure 4.5.1: Solution Flowchart.
4.5 Data Analysis Results

The deep learning approach was implemented using the architecture presented in Figure 3.2.3. The model was compiled with an early stopping callback that monitors the model loss and stops the training if it starts to rise. The callback has a patience of 20 epochs, which means it persists for this number of epochs before stopping the training. With this data, the model fitting was halted after 170 epochs. Figure 4.5.1 below visualizes the training history of the model.

The resulting model was capable of accurately classifying 80.01% of the videos in the testing set and 86.94% of the videos in the validation set. On the validation set, the average number of correctly classified videos for each word (sensitivity) was 59.12%, and on the testing set, it was 53.07%. The F1-score of the model was 56.22% and 52.39% on the testing set, representing how well the model was able to strike a balance in accuracy across the 2000 categories featured. Table 4.5.1 below presents this information.
Table 0.5.1: Model performance

<table>
<thead>
<tr>
<th>Measure</th>
<th>Testing</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.8001</td>
<td>0.8694</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.5307</td>
<td>0.5912</td>
</tr>
<tr>
<td>F-score</td>
<td>0.5239</td>
<td>0.5622</td>
</tr>
</tbody>
</table>

The other types of machine learning models that were used can only accommodate two-dimensional input data. The CNN-RNN model extracted features that were taken just before the final dense ANN layer featured in Figure 4.2.1. These features were then flattened into a two-dimensional data format that could be fed into these machine learning models.

**K-Nearest Neighbors (k-NN)**

To run the k-nearest neighbor model, values for the parameter K were tried between 2 and 100. It was discovered that K = 29 was the optimal value, with an accuracy of 88.02% cross-validation accuracy. Figure 4.5.2 below reports this information.

![Figure 0.5.2: optimal number of Neighbourhoods](image-url)
The model achieved an accuracy of 85.80% on the validation set and 83.10% on the testing set. The average sensitivity was 78.13% on the validation set and 75.12% on the testing set. The F-score values were 62.09% and 68.32% on the testing and validation sets, respectively.

**Decision Tree**

To fit the decision tree classifier, hyperparameters were tuned using grid search to find the best combination. Table 4.5.2 below reports the results of the search, and the optimal result corresponds to 81.27% cross-validation accuracy.

*Table 0.5.2: Hyper parameter Tuning for the decision tree model.*

<table>
<thead>
<tr>
<th>Measure</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_depth</td>
<td>None</td>
</tr>
<tr>
<td>max_features</td>
<td>None</td>
</tr>
<tr>
<td>max_leaf_nodes</td>
<td>None</td>
</tr>
<tr>
<td>min_samples_leaf</td>
<td>3</td>
</tr>
<tr>
<td>Splitter</td>
<td>best</td>
</tr>
<tr>
<td>min_weight_fraction_leaf</td>
<td>0</td>
</tr>
</tbody>
</table>

The optimal decision tree model had an accuracy of 82.50% on the validation set and 80.50% on the testing set. The average sensitivity of the model was 67.14% on the validation set and 65.15% on the testing set. The F1-score of the model was 60.17% on the validation set and 56.01% on the testing set. Table 4.5.2 below reports this information.

**Boosted Decision Trees**

In the boosted decision tree approach, the optimal combination of features included a learning rate of 0.01, a loss of log loss, a max depth of 10, auto max features, a minimum sample split of 5, and 100 estimators. This resulted in an accuracy of 76.86%. Table 4.5.3 below reports this information.
Table 0.5.3: optimal combination of hyper parameters for Boosted regression approach.

<table>
<thead>
<tr>
<th>Measure</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>learning_rate</td>
<td>0.01</td>
</tr>
<tr>
<td>loss</td>
<td>log_loss</td>
</tr>
<tr>
<td>max_depth</td>
<td>10</td>
</tr>
<tr>
<td>max_features</td>
<td>auto</td>
</tr>
<tr>
<td>min_samples_split</td>
<td>5</td>
</tr>
<tr>
<td>n_estimators</td>
<td>100</td>
</tr>
</tbody>
</table>

The approach yielded an accuracy of 85% on the validation set and 83.10% on the testing set. The sensitivity was 78.61% on the validation set but 78.50% on the testing set. The F1-score was 74.61% on the validation set and 74.62% on the testing set. See Table 4.5.5 for this information.

**Support vector classifier**

A support vector classifier was also implemented with grid search to optimize the hyperparameter combination. The four parameters searched included cost, gamma, kernel type, and tolerance. Table 4.5.4 reports the optimal combination, which yielded 83% cross-validation accuracy.

Table 0.5.4: optimal parameter combination for Support vector classifier

<table>
<thead>
<tr>
<th>Measure</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>1.0</td>
</tr>
<tr>
<td>gamma</td>
<td>scale</td>
</tr>
<tr>
<td>kernel</td>
<td>linear</td>
</tr>
<tr>
<td>tol</td>
<td>5</td>
</tr>
</tbody>
</table>

The model was able to achieve 85.00% accuracy on the validation set and 85.50% on the testing set. The average sensitivity was 74.44% on the testing set and 77.10% of the validation set. The F score for the model was 79.30% on the validation set and 79.20% on the testing set. See table 4.5.5 below.
### Table 0.5.5: Performance evaluation metrics for DT, SVM, BoostedDT and KNN

<table>
<thead>
<tr>
<th>Model</th>
<th>Measure</th>
<th>testing</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN</td>
<td>Accuracy</td>
<td>0.8310</td>
<td>0.8580</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>0.7512</td>
<td>0.7813</td>
</tr>
<tr>
<td></td>
<td>F- score</td>
<td>0.6209</td>
<td>0.6832</td>
</tr>
<tr>
<td>Decision tree</td>
<td>Accuracy</td>
<td>0.8050</td>
<td>0.8250</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>0.6515</td>
<td>0.6714</td>
</tr>
<tr>
<td></td>
<td>F- score</td>
<td>0.5601</td>
<td>0.6017</td>
</tr>
<tr>
<td>Boosted regression</td>
<td>Accuracy</td>
<td>0.8550</td>
<td>0.8500</td>
</tr>
<tr>
<td></td>
<td>Sensitivity</td>
<td>0.7444</td>
<td>0.7710</td>
</tr>
<tr>
<td></td>
<td>F- score</td>
<td>0.7920</td>
<td>0.7930</td>
</tr>
</tbody>
</table>

#### 4.6 Comparison

Comparing the five models based on their performance metrics, the proposed model (ANN) achieved the highest accuracy (86.94%), followed by KNN with 85.80% accuracy on the validation set. Boosted decision tree and support vector classifier had the lowest accuracy at 85%.
CHAPTER 5

5.1 Conclusion

The study has successfully implemented a machine learning classifier that can predict the words in a sign language video with precision. Five machine learning models were implemented and compared, with an artificial neural network approach found to be the most suitable for this task. For all methods, feature extraction was carried out using deep neural layers of CNN and RNN.

5.2 Recommendations and Future Research

In future research, it is necessary to expand the vocabulary of the models to increase the number of words that can be recognized. To achieve this, larger computing platforms such as GPUs and computing environments capable of handling larger datasets are needed. In addition, more research time is required to fine-tune the models to improve their performance.
ABBREVIATION


https://doi.org/10.1145/1361684.1361685


