Summer 2022

The use of deep learning solutions to develop a practice tool to support Lámh language for communication partners

Gabriel Bueno Pimentel Borges

The use of deep learning solutions to develop a practice tool to support Lámh language for communication partners

Gabriel Bueno Pimentel Borges

A Thesis Submitted in Partial Fulfilment
of the requirements for the
Degree of
Master of Science in Data Analytics

August 2022

Supervisor: Dr. Vladimir Milosavljevic

CCT College Dublin

Assessment Cover Page

To be provided separately as a word doc for students to include with every submission

---

Module Title:	Capstone Project
Assessment Title:	The use of deep learning solutions to develop a practice tool to support Lámh language for communication partners
Supervisor Name:	Dr. Vladimir Milosavljevic
Student Full Name:	Gabriel Bueno Pimentel Borges
Student Number:	SBA21590
Assessment Due Date:	19th of August 2022
Date of Submission:	16th of August 2022

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Abstract

This study has proposed an alternative to promote the learning and enhancement of Lámh language for communication partners that support current users by creating a real time detection tool to recognise 20 chosen Lámh signs based on existing studies in the field. This implementation was carried out by generating primary data composed by MediaPipe landmark numpy arrays of 40 frames and 45 repetitions per sign. The Neural Networks were built using the Python library Keras and the applied SVM models were built with the library sklearn. The real time detection was carried out by integrating the mentioned elements with the library OpenCV. Neural Networks with different architectures with Long Short-Term Memory (LSTM) and 1D Convolutional Neural Network (CNN) were compared with SVM classifications applied with cross-validations to achieve the optimal hyperparameters in order to determine the most appropriate model.

The final chosen model after the assessment of the training and testing accuracy and loss was the two 1-D CNN layers with 32 and 64 nodes respectively, a dropout of 0.2 followed by two LSTM layers with 32 and 64 nodes respectively and a dense layer of 32 nodes. The training accuracy was 99.86%, the testing accuracy was 93.33%, the training loss was 0.0035 and the testing loss was 0.1791. This was the model which performed better in a real-time detection environment, easily detecting 8 different Lámh signs and detecting other 6 with reservations.

For future work, some skeletal motion signs should be captured again and other data augmentation strategies should be adopted, like capturing hips and legs landmarks alongside the signs and explore the augmentation of the data by promoting offset measures of the landmark coordinates of the skeletons captured by MediaPipe. Once the

corrections of the methodology achieve better real time results, works toward tool accessibility and user experience should be investigated in order to generate a Lámh language real-time detection tool that could potentially promote Lámh and become a learning alternative for communication partners.

Acknowledgments and Dedication

Many thanks to the Lámh language representatives for providing online lecturing materials to support this research, and for the several meetings with deep and fruitful discussions about this language and its social context in Ireland.

I am extremely grateful to Springboard+ and National Broadband Ireland for having sponsored my Master's course and, consequently, this research.

I am also thankful to my supervisor, Dr. Vladimir Milosavljevic, and all the lecturers who I had the opportunity to know and learn from in CCT College Dublin.

I dedicate this research to my beloved wife, Julia, who has always been by my side, bringing the best of me. Even being far away, my family is my unconditional support: my parents, Karina and Daniel, my siblings, Davi and Maria Eduarda, my grandparents, uncles, aunts, cousins, my godson Heitor, and my little sister born this year, Maria Flor. They are all a part of who I am.

This research is specially dedicated to the Lámh users and the professionals of the field: pre-school teachers, primary school teachers, nurses, special needs assistants, speech therapists, social care workers and all the other professionals who make their will to help the others their work.

Summary

List of Figures

List of Tables

1. Research Question

At first glance, technology interventions might seem to contradict the essence of unaided augmentative and alternative communication (AAC) within key word sign (KWS) languages such as Lámh, since authors like Sigafoos *et al.* (2014) have defined this type of communication as the use of gestures and manual signs without the assistance of any other external object. However, considering the aspects investigated in this research regarding Lámh language and its promotion, accessibility and training for communication partners, what solutions could be proposed on a data analytics perspective to assist people who use or must learn Lámh language – or even other KWS languages?

2. Relevance

Communication partners (CP) are defined by Kent-Walsh and McNaughton (2005) as people who either compose the AAC user's personal life (family and friends) or retain an education or care nature towards the individual (e.g.: teachers, health care professionals). However, the authors attest that very little attention has been given to the improvement of AAC learning methods for CP intervention programs.

A solution focused on improving the learning methods specifically for Lámh language communication partners may be proposed considering the fragility of this field according to the authors presented in the literature review. Such a solution would not extinguish the practicality and the portability that Wilkinson and Hennig (2007) highlight as the main advantages of KWS languages, once the assisting tool would not be introduced in the direct relation between the user and their communication partner.

Instead, a data analytics supporting alternative could be applied at the preparation of the communication partners, in a stage of the learning process that Kent-Walsh and McNaughton (2005) classify as "Controlled Practice and Feedback". The latter consists in a controlled environment with only the communication partner, the Lámh user and an instructor.

In this context, if a Deep Neural Network model is able to classify Lámh signs in real time, this model can be used in the "Controlled Practice and Feedback" stage, giving the communication partner the possibility to train the learnt signs more often. Hence, the CP acquires confidence and knowledge to be able to use the language in a natural environment, directly with the AAC/Lámh user.

Therefore, the proposal of a solution focused on facilitating the learning methods for Lámh communication partners using Deep Learning for sign recognition would promote more democratic, accessible and playful learning opportunities for those involved in the Lámh user's life. Furthermore, the consistent implementation and use of this communication support can also provide a wide variety of benefits to one's inclusion in society.

3. Contribution

The novelty of this work is the application of machine learning functions and tools to classify Lámh language signs. The main contributions of this work are firstly the data acquisition methodology – since there is no existing database of Lámh signs for machine learning classification, after the best Machine Learning models are chosen, they were evaluated in actual real time detection applications – and secondly the comparison of Convolution Neural Network, Long Short Term Memory and Support Vector Machine for

the classification of these signs, defining the most appropriate method for the proposed context.

4. Objectives

4.1. Primary Objectives

The primary objective of this project is to obtain a highly accurate and generalised machine learning model that must effectively predict the movements of users through real time detection, labelling these in accordance to the respective meaning within the list of signs of the Lámh language. Due to the time frame imposed on this project, twenty signs should be trained by the model instead of the totality of the Lámh signs available.

4.2. Secondary Objectives

To achieve the primary objective, the secondary objectives below must be achieved:

- To gather and to capture comprehensive data that represent the proposed twenty Lámh signs, recurring to data augmentation strategies when applicable to enhance training and testing results. The data must reflect the needs of the project, capturing different people, under different lighting, making the same signs with their own ways as an attempt to predict the many different scenarios and circumstances that the final trained Deep Learning method must cover. The final accuracy of the learning model's algorithm depends on the integrity of the input-data representation (Alzubaidi *et al.*, 2021);
- To explore and choose the best data augmentation strategies and other methods like dropout to reduce possible overfitting of the trained models. Since, due to the specificity of Lámh language signs, datasets such as COCO, PASCAL VOC and

Imagenet are not viable sources. Hence, the image data must be methodically captured during the research process or from existing sources of the Lámh sector. The mentioned lack of image data related to Lámh signs in large scale requires data augmentation as an important alternative for the project to succeed when considering the application of Deep Learning techniques;

- To explore and to compare the different architectures and methods of DNN applied to the context of the project, choosing the most appropriate and generalized for the Lámh sign language recognition. These models should be compared with Support Vector Machine in real time detection context so it can be confirmed whether the best DNN model tested overcame the best Support Vector Machine tested. The types of DNN that will be tested in this research are based on previous successful models that carried out object detection and classification of sign languages;

- To consolidate the previous objectives into an artifact, obtained through coding, which will turn the machine learning outcome into an interface with users. Initially, the accessibility of this technology will not be explored as it involves complex design solutions and user driven studies and analyses that go beyond the scope of this project.

5. Literature Review

5.1. Augmentative and alternative communication, Key Word Signing and Lámh

Sigafoos *et al.* (2014) and Wilkinson and Hennig (2007) similarly define augmentative and alternative communication (AAC) as an area of research of speech-language that supplements and creates alternatives of communication for those who cannot communicate effectively through conventional verbal language. These methods can help the individuals to express themselves, Schlosser and Wendt (2008) even identify cases in which AAC helped users to develop natural language, although they alert that the magnitude of the learning can be minimal and should not be expected as an outcome.

Byrne, Pyne and Sheehan (2019) synthesize that, within AAC, an alternative to support people that face difficulties in their communication is key word signing (KWS), which is communication through manual signing alongside the spoken word incorporating features from sign language (Tan *et al.*,2014), being Lámh the KWS language practiced in Ireland (Byrne, Pyne and Sheehan, 2019). Lámh vocabulary is made of 580 different signs categorised as actions, modifiers, objects, people and social words, and they are originally from the Irish Sign Language (Frizelle and Lyons, 2022).

Dolly and Noble (2018) define that, although this type of communication enhancement is important to be established in a one-to-one relation between the client and the speech therapist, the communication partners (CP) would play an important role by bringing key word signing to the natural setting of the individual.

In the Lámh context, Dolly and Noble (2018) and Byrne, Pyne and Sheehan (2019) highlight that on-going training is effectively important for communication partners, although the latter authors have identified that the studied staff that support Lámh users

had variable levels of knowledge of this language. This becomes a concern in regards to the inclusion of users that require Lámh language since key word signing is a method of language reception and expression (Frizelle and Lyons, 2022).

Byrne, Pyne and Sheehan (2019) also point out that the studied group had a will to learn more signs and believed in the benefits of Lámh to its users although 20% of them confirmed they rarely applied the language on their routine, working on a day-to-day with Lámh users. Seeking a rationale, the authors identified in their studied group a lack of confidence in their own performance with the language. The authors then concluded that the lack of training and practice interferes in the proper application of key word signing on these professionals' crafts.

In Ireland, Frizelle and Lyons (2022) alert that the only entry level course funded for primary school teachers, social care workers, special needs assistants (SNAs) and other professionals from related sectors is Lámh Module 1, that brings 100 words of the vocabulary and consists of three hours of online self-learning and three hours and a half of oriented learning with a Lámh Tutor. The authors also enhance that the vocabulary available in the course does not necessarily reflect the main words the professionals require in their routine with the Lámh users. The authors mention, for example, that there is a considerable difference between the main vocabulary needed by a child with down syndrome in pre-school from the main Lámh vocabulary for a child with down syndrome in primary school due to the different moments of the children’s development. This emphasizes that an extended vocabulary should be absorbed by the professionals in the area, alongside a frequent practice as recommended by Byrne, Pyne and Sheehan (2019).

In order to propose a beneficial intervention to the matters raised, an accurate object detection tool that is built with Deep Learning techniques could classify a physical sign based on labeled images, possibly generating a real time sign classification tool for practice of Lámh language signs, since Sennott *et al.* (2019) recognise the possibilities given by Deep Learning when managing unstructured data (images, videos, audios, texts). The authors also list the several applications of Deep Learning and the beneficial impact it has already caused in AAC so far, including speech recognition, moving detection and alternative access to AAC.

5.2. Machine learning and the computer vision studies and their applicability to existing sign languages

Sanchez, Romero and Morales (2020) have considered artificial intelligence as the core of the contemporary industrial trend, which the authors defined as the fourth industrial revolution, having as one of its main roles the processing and analysis of Big Data. Xue-Wen Chen and Xiaotong Lin (2014) set a common ground of the definition of Big Data as an exponential availability of digital data that cannot be managed by conventional software tools and techniques. Chan (2013) contextualizes the daily volumes of Big Data in the magnitude of terabytes (10^{12} bytes), petabytes (10^{15} bytes), or exabytes (10^{18} bytes) depending on the industry. In this context of exponential data, Sanchez, Romero and Morales (2020) highlight the field of Machine Learning within artificial intelligence as the capacity of computers to learn by themselves using data input with support of statistics. Within the several machine learning models to process data, the latter authors define deep learning (DL) as the area that thrives in terms of performance when dealing with the abundant growth of data previously commented, having as practical

advantages the automation of selection of features from the input data (Alzubaidi *et al.*, 2021) and the discovery of hidden patterns within the data (Shrestha and Mahmood, 2019).

Shrestha and Mahmood (2019) and Alzubaidi *et al.* (2021) explain the basic architecture of the DL neural networks as a similar structure of human neurons connected in deep layers with highly optimized algorithms. The architecture that organises these layers and their connections can be variable, but they will always include an input layer, hidden layers and a final output layer. Each layer is composed of nodes that contain values, the nodes of one layer are connected to the nodes of an adjacent layer by the application of a weight. Once the first input goes to the first node, it is priorly multiplied by a weight and then summed to the value in the node. A mathematical function is applied on each node in order to compute partial derivatives or error deltas generated by the application of the weight (Shrestha and Mahmood, 2019), this process is sequentially carried out until the final output layer is obtained.

Amongst the different types of neural networks, Sanchez, Romero and Morales (2020), Shrestha and Mahmood (2019), Alzubaidi *et al.* (2021), Srivastava *et al.* (2021) and Sharma, Jain and Mishra (2018) have identified Convolution Neural Network (CNN) as the DNN model ideal for the processing and the classification of images and videos, but not exclusively for this purpose, as the authors also identify the efficiency of CNN in other fields like speech recognition, a good example being the work by Kim *et al.* (2020). Since the outcome of this project is based on images and videos, a special attention will be given to CNN rather than other neural networks and machine learning models in order

to propose a better understanding of this model and the diversity of architectures and methods suggested by authors only within the last ten years.

CNN distinguishes from other DNN models mainly on the application of convolutional layers in the early stage of the Neural Network structure, as clearly described by Alzubaidi *et al.* (2021). The convolutional layer, according to the authors, is formed by the application of the Kernel, which is in essence a matrix of different weights that slide through the data (vertically, horizontally or both depending on the chosen CNN). Each value of this matrix will be multiplied by a corresponding value in the process of sliding through the data, the products of each multiplication are then summed and the process repeats for the next snippets of the data until the kernel is completely applied. The result of this process is an optimal reduction and selection of features from the data.

The authors also identify peculiarities of the CNN that makes it uniquely advantageous: the nodes from one layer are not connected each one of them to all the nodes from the adjacent layer, which is a memory-effective approach; the weights are applied equally in the whole data, reducing data training time and costs.

Alzubaidi *et al.* (2021) describe another important component of a CNN: the pooling layer, which is applied after the convolutional layers and it is responsible to generate samples from the feature maps generated from the convolution layers, keeping the relevant features. The authors highlight the most common pooling layers as max, min and GAP pooling.

In CNN, after every weight layer, Alzubaidi *et al.* (2021) identify the existence of activation functions, which will transform the input layer of the node into a value in accordance with the needs of the architecture of the CNN. The authors enhance that the

most common activation functions are ReLu, Sigmoid and Tahn. The former is widely applied but contains variations in cases that the deactivation of neurons must be avoided, a phenomenon described as "Dying ReLu" (Lu, 2020). Zhang and Wan (2019) enhance the importance of the activation functions as, without them, CNN models would be simple regression models that would not be able to handle complex data like images, videos or audios.

Finally, Alzubaidi *et al.* (2021) indicates that usually the fully connected layer is set at the end of the CNN model to classify the data from the transformed feature maps. All these elements that compose CNN - dimension of the kernel, number of convolutional layers, types of activation functions and others - are tested by researchers for the creation of more efficient architectures. Some of these architectures, as highlighted by Alzubaidi *et al.* (2021), are AlexNet (Krizhevsky, Sutskever and Hinton, 2017), Visual Geometry Group (Simonyan and Zisserman, 2014), GoogLeNet (Szegedy *et al.*, 2015), Residual Networks (He *et al.*, 2016) and Xception (Chollet, 2017).

In the last decade, methods have progressed to make CNN more accurate and fast. The main methods could be split into: region-based CNNs - R-CNN (Girshick *et al.*, 2014), Fast R-CNN (Girshick, 2015), Faster R-CNN (Ren *et al.*, 2017) and Mask R-CNN (He *et al.*,2017) -, which work with algorithms that pre-select the bounding boxes that have higher probabilities of containing the objects; single shot detection CNNs - You Only Look Once (YOLO) by Redmon *et al.* (2016) and Single Shot Multibox Detector (SSD) by Liu *et al.* (2016) - which, in a single network, apply bounding box regression and then classify the detected objects. Sanchez, Romero and Morales (2020) summarize a certain trade between accuracy (measured as mean Average Precision) and speed (measured

as Frames per Second) of these models: while the region-based models seem to be very accurate, the single shot CNNs compensate in speed but lose in accuracy of small objects as the authors of these models themselves highlight in their articles. Thus, none of these methods outperform the others in all possible perspectives and must be assessed on different cases.

In the context of sign language detection and classification studies, Liao *et al.* (2019) enhance that Faster R-CNN presents an outstanding performance when predicting more subtle gestures of the hands. This is imperative when it comes to detecting and classifying sign language vocabulary. Although Frizelle and Lyons (2022) conceptually indicate that KWS languages are simpler than other sign languages in terms of hand positions and finger spelling - the latter is nonexistent in the Lámh language -, one can still identify Lámh signs that differ from others by small differences in the fingers' disposition - for example, the initial gesture of the action vocabulary 'to open', which is composed by the fingers of one hand touching the fingers of the other hand on a single centralized reference at the chest level, could be considered quite similar to the sign of 'ball', which contains the same disposition of palms facing each other at the chest level with fingers of one hand mirroring and touching the fingers of the other hand, but with a wider distance between the fingers on the same hand, forming the ball shape. Hence, single shot detection CNNs may present inaccuracies if compared to region-based CNNs when classifying different signs that may be distinguished by gestural subtleties, but this must be confirmed through experimentation.

In order to assess the CNN model and its varieties of architectures and methods, Karim, Pujak and Sewak (2018) demonstrate in their book how Tensorflow, an open

source library with API for several programming languages, contains functions and resources for creation and processing of CNN structures. To make it more accessible for data analysts, the authors identify the use of Keras, which comes from TensorFlow. This library is a DNN API in python that makes the coding more accessible, with simpler commands to train and process the DNN models. Shrestha and Mahmood (2019) enhance in a chart comparison how Tensorflow is the most well-evaluated deep learning library in GitHub when compared to other renowned libraries like scikit-learn or pytorch.

In terms of object detection and classification of human gestures, studies carried out after the advent of better classification algorithms discuss the different methods of data acquisition. Vo *et al.* (2017) summarise these methods as sensorial-based and vision-based data acquisition. The sensorial-based methods make use of additional technologies with sensors that must be worn by the user. Mehdi and Khan (2002) generated an American Sign Language (ASL) predictor with an Artificial Neural Network with seven neurons on the input layer, fifty four neurons on the hidden layer and 26 neurons on the output layer. The input of the data was made using a seven-sensor glove that should be worn by the user and each sensor would return integer numbers that would indicate the bent of the sensor. These integers would be processed by the neural network that would classify the input in accordance to the ASL alphabet, which would then be transformed from text to speech. Although, at the time it was published, the model proposed pioneer elements combined and with a basic neural network it was able to achieve 88% of accuracy, bringing such solution to the contemporary context raises limitations: the glove sensors only detects static gestures that specifically depend on the hand and no other part of the body; this type of solution, alongside other types of sensors,

like the energy sensor of the model proposed by Kosmidou and Hadjileontiadis (2009), is not widely available, which becomes a problem considering that the input of the models rely on the use of the proposed technologies. Lefebvre et al. (2013) present a model of gesture recognition that contains as input 6D input vectors that combine the linear acceleration and the angular velocity captured by the accelerometer and gyroscope from Micro Electro Mechanical systems, which, as the authors confirm, are technologies present in Smartphones. Although Smartphones are more used than previously mentioned sensor sources, this approach is also dependent on either the right hand or left hand, limiting the scope of this data input alternative. Still, it is important to acknowledge the discoveries made by these studies that could potentially bring benefits for users that were able to access these sensors and solutions.

The vision-based data acquisition, differently from the sensor-based, consists of data inputs of images and videos translated into matrix representations that would then be processed by the machine learning model. There are technologies of image capture that can generate a three dimensional imagery, known as RGB-D - a widely known example is the Microsoft Kinnect, that contains, not only an RGB camera, but also an Infrared (IR) camera and sensor, and this set is able to read images in three dimensions (Zhang, 2012). Authors like Jiang *et al.* (2018) were able to implement the depth input and the skeleton tracking available by Kinnect to increase the performance of the CNN model applied, overperforming state of the art models of the CNN itself and Support Vector Machine (SVM) sign language classifications. However, as mentioned by Zhang *et al.* (2019), RGB-D technologies are more expensive and require additional computation for the depth data. Liu *et al.* (2017) mention that there is another technique that composes

the types of visual-based data acquisition with RGB-D, which is the skeletal data. By the time the authors published the article, one of the concerns regarding the skeletal data in comparison to RGB-D is the dependency on a tracking algorithm that tended to be harder to obtain than getting the data from RGB-D images.

However, the advent of Google's MediaPipe must be enhanced (Lugaresi *et al.*, 2019). This tool started as an object detection with bounding boxes and face landmark recognition framework but already counts with additional frameworks derived from its initial development, which are the Blazepose for body poses (Bazarevsky *et al.*, 2020) and the MediaPipe Hands (Zhang *et al.*, 2020), creating new opportunities for gesture detection using skeletal data. These tools have pre-trained neural network models that generate landmarks in a human's pose, face and hands. Their application is already noticeable: Halder and Tayade (2021) used MediaPipe on their first stage of data acquisition by capturing the landmarks on the images used in their study. The authors created a model using Support Vector Machine (SVM) for the classification of signs of American, Indian, Italian and Turkey sign languages. This combination outperformed models that classified the same languages using SVM itself and CNN models, with accuracy above 99%. The authors conclude that MediaPipe played an important role by precisely identifying complex hand gestures prior to the machine learning application. Bagby *et al.* (2021) built a CNN classification model based on 15,000 images of twenty six signs of the ASL alphabet on the top of MediaPipe, which would detect the landmarks of the hand gestures. Although the authors obtained great results for 14 classes of the ASL alphabet (99%), the same result was not identified for classification of 20 different signs. The authors concluded that a larger dataset needed to be adopted for the

classification of a greater variety of signs. This converges to the concept of Neural Networks, which theoretically perform better with exponential data increase.

It is also important to distinguish the gesture classification studies between the static gesture recognitions and the dynamic gesture recognitions. Solutions and outstanding achievements from both types of gesture recognition must be considered as Lámh language contains signs that can be considered statical, but there are also dynamic ones. The vision-based studies mentioned so far dealt with static gestures and signs. As mentioned by Koller *et al*. (2018), although CNN models have outperformed other approaches in all computer vision tasks, CNN-based approaches don't take the temporal domain of video data into consideration as they should. However, it is possible to utilise statistical and machine learning solutions to enhance the prediction of dynamic gestures. The Hidden Markov Model (HMM) is defined, but not introduced, by Rabiner and Juang (1986) as a statistical probability method with an underlying stochastic process that is not observable (hidden stages), but can be observed through another stochastic process that produces the observed symbols. In the context of sign language detection of a video, the temporal sequence is the hidden stage, whereas the generated vocabulary is the observed symbol.

According to Koller *et al*. (2018), HMM dominates automatic speech recognition but remains unpopular in computer vision tasks. Vo *et al.* (2017) compared SVM and HMM to recognise dynamic gestures of the Vietnamese sign language (VSL) using as a base 30 gestures detected with Kinect of 5 different volunteers. The frames would be adjusted to a normalized height and width, and the dimension of the number of frames would be transformed to the same dimension of the height and width. The SVM model

drastically outperformed the accuracy of the HMM model in the three trained grid sizes (4x4x4, 16x16x16, 32x32x32) reaching 95% in one of the cases. However, Koller *et al*. (2018) applied the probabilities of HMM on the top of CNN with a given probability of n-gram language model from the SRI Language Model Toolkit (Stolcke, 2002) since the predicted datasets were syntactic constructions of sequential words with sign language. As a result, the authors were able to generate a model that outperformed HMM models and combinations of CNN neural networks with other neural networks. The comparison was established in Word Error Rate (WER) instead of prediction accuracy. It can be concluded that the author used speech recognition techniques to predict sign language videos and the results seem to be promising.

Raheja, Mishra, and Chaudhary (2016) built a model that predicted 4 Indian sign language gestures using SVM and data captured by regular webcams and Microsoft Kinect, reaching 97,5% for the 4 signs. Extending to models that classified more gestures, different machine learning solutions were presented. Zhang and Wang (2019) applied a 3D CNN to predict 26 gestures that do not correspond to an official sign language. Using a regular camera to capture 12 frames per second videos as inputs, the authors were able to obtain 90% accuracy with the proposed model. The third dimension of the CNN would be related to the order of the frames. As the authors described, one can convert the sequence of frames of a video into another dimension of the matrix that represents a single image. However, the convolutional layer and the pooling layer must consider this additional dimension, becoming 3D layers.

Zhang *et al.* (2019) present in their article a gesture recognition by a 2D CNN model with VGG architecture (previously mentioned) and, for the first time in a study that

addresses gesture recognition, also applies a 3D Densenet. Densenet was created by Huang *et al.* (2017) in order to solve the vanishing or exploding gradients in deep CNN architectures. When the Neural Network, not only the Convolutional one, gets too long and deep, the multiple application of activation functions and weights interfere in the gradient of the loss function, either generating gradients that are too small to be accounted for or too big, drastically interfering in the optimization of the training of the model.

Before the release of Densenet, the previously mentioned Residual Networks (Resnet) by He *et al*. (2016) observed that, the more the CNN layers were increased, the greater was the error of the model when compared to the CNN with less layers. Differently from the authors of Densenet, the Resnet authors don't see the vanishing or exploding gradient as the problem of regular deep CNNs as they believe the batch normalization by Ioffe and Szegedy (2015) - which introduced normalised activation within Neural Networks - already addressed the gradient control in regular CNNs. Instead, the authors described the increase of the error of the model alongside the increase of the layers to - as the authors named - a degradation in deep Neural Networks. The authors then proposed an intervention based on shortcuts between some of the stacked layers of their model to diminish the mentioned degradation in deep CNNs that showed increased errors when the layers were increased. By applying this measure of residual layers, the authors won first place in benchmark competitions of object detection without the use of dropout measures. In the Resnet article, it is demonstrated that models with residual layers actually reduce their error when the number of layers are increased. Nevertheless, this does not mean that complex and deep neural networks become a response to all machine

learning problems. As Simonyan and Zisserman (2014) defined in their introduction to the architecture Visual Geometry Group (VGG), deep CNN models are a solution to large scale data, and, as it could be seen with Resnet and batch-normalization, by addressing a complicated subject such as large data, deep CNN models inherit problematic phenomena from their multiple layers, like the CNN degradation, that require non-trivial corrections to improve the deep layer models' accuracy.

What Densenet suggested, having its publication after Resnet, was a group of measurements that compose its model in order to reduce the impact of deep layers in the gradient - which they believed was the core problem of deep layers. They proposed: severe reduction of the size of layers, 12 neurons, narrowing the model in comparison to previous ones; dense blocks in which all layers are connected to all the following ones within the block in a process that the authors described as a concatenation of features, which would then reduce effects of vanishing or exploding gradients of the model.

In the context of the implementation promoted by Zhang *et al.* (2019), the 3D Densnet would have the 2D Densenet architecture, but replacing the 2D CNN layers by 3D ones, as previously mentioned, in order to process the additional dimension of the matrix related to the frame sequence of the dynamic gestures. Not only the 3D Densenet model was adopted, but also the 2D CNN that would process the same training data in parallel to the 3D Densenet. The way the authors encountered to process sequential images in a 2D Neural Network was the conversion of the image sequences into a single one using as reference a simple but effective process of image representation: movement history image (MHI) by Bobick and Davis (2001). The latter authors describe MHI as a variety of motions represented in a spatially indexed way, in which the intensity of the

pixels is mathematically related to the temporal variable. Interestingly, by applying this method and fusing the classifications of 2D CNN with MHI and 3D Densenet through a simple probabilistic function that would take into consideration the probabilities of classifications from both models (2D CNN and 3D Densenet), Zhang *et al.* (2019) identify that the predictions of the fused models at feature level outperform the predictions of each model separately with 87.8% and 87.1% of accuracy in two datasets, and the fused models at decision level instead of feature level achieve 89.1% and 89.5% when predicting the same two datasets. The authors indicate that the accuracy of the model could be potentially increased if Internal Transfer Learning (Wang *et al.*, 2017) or Recurrent Neural Networks (RNN) – more specifically the Long short-term memory functionality (Hochreiter and Schmidhuber, 1997) – were applied to the model.

Internal Transfer Learning (Wang *et al.*, 2017) seems to have been developed specifically to address the application of CNN models in small datasets. The concept behind this technique is to train the model as a binary classification for each pair of classes that can be combined from the N classes that must be predicted. The top five trained networks from these binary classifications would be picked and there would be a transfer learning of its weights to five networks that would then need to classify all the N classes instead of two. The model that performed better amongst the five would be the final trained model. This method was applied to a pre-trained 3D CNN and, applied in datasets of videos with dynamic actions, the authors were able to reach high predictions above state of the art models - 98.2%, 100%, 93.6 and 96.1 depending on the dataset. Probably the application of this method for a big dataset would not be ideal as the multiple training rounds of binary classifications would be more time consuming for bigger

datasets. Potentially, in theory, this technique would not be required for large data as they fit better in regular CNN solutions.

RNN is another type of Neural Network. Its structure is distinguished from other NN models as it is not a feedforward NN (Shrestha and Mahmood, 2019). Instead, they are processing units that form a cycle. RNN had limitations in relation to vanishing and exploding gradients, similarly to what was exposed previously for deep CNNs. In the case of the sequential architecture of RNN, Long short-term memory (LSTM) by Hochreiter and Schmidhuber (1997) was created to control the gradient issue by adding what the authors called memory cells and gate units with binary outputs that would control the access to the memory cells in order to stabilize the error flow. Because of its approach, RNN models are widely used in sequential problems like time series or speech recognition (Shrestha and Mahmood, 2019; Alzubaidi *et al.*, 2021).

Putting LSTM in the context of classification of sequential image frames, there can be similarities drawn with the previous examples in which RNN were applied. Du, Wang and Wang (2015) demonstrate the importance of LSTM for RNN performance when they compare 5 different types of RNN architectures for skeletal human action recognition and the models containing LSTM layers outperform the accuracy of the models that only have layers with Tanh functions. The authors also compare models that use deep RNN with the whole skeletal as input with hierarchical models that use fusion layers through its architecture to merge neural network layers of 5 different parts of the body (right leg, left leg, trunk, left arm and right arm). The authors concluded that the hierarchical models had a better performance when applied to the analysed datasets than a trained deep RNN in the same study and state of the art deep RNNs that were trained with the same

dataset. However, the authors emphasized that the LSTM layer tended to generate overfitting, and, curiously, dropout strategies would not help to diminish the overfitting condition. The solution found by the authors was the adoption of strategies - being the most relevant the weight noise - described by Graves (2011) in their study of controlling overfitting of the hierarchical RNN model developed to recognise phonemes from a speech corpus. The concept and different types of weight noises specifically applied in RNNs as opposed to feedforward neural networks are thoroughly described by Jim, Giles and Horne (1996) and they consist on additional or multiplicative parameters applied to the weights of the Neural Network primarily to generalise the model, reducing the gap of the training error to the testing and validation error, but the authors theoretically demonstrate that the convergence can also be improved with weight noises. This is an important consideration for overfitting strategies when dealing with Neural Networks having as a reference the improvement of the RNN model generalisation brought by the mentioned authors.

Liu *et al.* (2016) extended the concept of LSTM to ST-LSTM, which consists of the LSTM structure with spatial and temporal processing. The authors added in the article an efficient way of representing the skeletal joints in a tree-based structure for RNNs in a transversal fashion. This chain disposition of skeletal joint relations can discover stronger long-term spatial dependency of the adjacent structures of the joint. The authors also implemented memory cells that work similarly to the LSTM but specifically for the context of the human motion, in which the model predicts how the joint will move and this prediction is compared to the actual move. The error generated between the prediction and the actual input would generate what the authors called ‘trust gate’, which will

influence the memory cells of the model. This means that, if a predicted input was quite different from the actual input of the movement, the trust gate could block the input gate to prevent the memory cell from being updated based on an irregular input. Curiously, the model is therefore trusting the prediction more than potential noisy inputs. This is due to the context of the study in which the Kinect RGBD camera could eventually not detect the skeletal structure correctly, leading to data noises in some joints of the body. By applying the newly ST-LSTM model with the transversal tree to represent the skeletal structure and using trust gates, Liu *et al.* (2016) were able to generate an accuracy that overperformed the accuracy of other models that analysed similar Kinect datasets of skeletal actions.

Liu *et al.* (2017) advance the concepts from ST-LSTM by creating the Global Context-Aware Attention LSTM (GCA-LSTM). The main difference of this other approach for LSTM is the creation of a global context memory cell that is initialised as the sum of input of each joint on each frame divided by total number of joints multiplied by number of frames. In a sequence of equations, the global context memory cell is used to obtain an informativeness score that is a real number between 0 and 1 for each joint on each frame and is applied in the same function of the trust gate of the ST-LSTM previously mentioned. By doing this, the informativeness score is supposed to enhance the joints that are more relevant in the action. In the end of the LSTM layer, the global context memory is recalculated based on the last output and the process is repeated through different iterations until the memory cell is refined to be applied to a softmax classification function to classify the action. The authors were able to improve the accuracy of the ST-LSTM proposition, which had already overcome the state of the art.

One could notice that the structure of RNN and its improvement, LSTM, follow a sequential structure that could be considered similar to the HMM, previously mentioned. Salaün, Petetin and Desbouvries (2019) compare both approaches and they have considerable differences, like the transition from one step to another: whilst RNN operates with activation functions, HMM considers transition probabilities. Also, according to the authors, RNN makes deterministic deductions from the current observation and the previous variable to move to the next stage, while HMM is an stochastic model. Therefore, although HMM and RNN contain similarities in the types of mathematical problems both solve (timeseries, speech recognition) they make essentially different probabilistic assumptions. Moreover, being RNN a type of deep learning technique, it would tend to be more efficient in the classification of large datasets. However, it was demonstrated earlier that HMM can be merged with Neural Networks to enhance the model accuracy in problems of sequential events like the sign language detection of Koller *et al*. (2018).

Li *et al.* (2017) explored the spatial-domain-features used in the LSTM architecture and temporal-domain-features used as input in the CNN architecture to classify the actions of NTU RGB+D Dataset. However, instead of creating a single architecture having both layers, the authors created 10 neural networks being 3 LSTM that are trained based on the spatial features of the skeletal data and 7 CNN models trained based on temporal features of the skeletal data. The choice of the authors is curious when verifying the previous concepts presented for LSTM and CNN as, according to their strengths and benefits, CNN seems to be more powerful to enhance the main features on an image, for example, whereas LSTM is widely used for sequential or temporal data as several authors applied this Neural Network calculation in data composed by videos. Li *et al.* (2017)

therefore inverted the applications of CNN and LSTM. Their final structured was based on a final score fusion that, for each classification, would choose the index that contained the highest when multiplying the scores of the 10 parallel Neural Networks. The method was able to achieve the state of the art applied to the same dataset.

Differently from Li *et al.* (2017), Ercolano and Rossi (2021) propose an architecture in which the LSTM layers are applied after multiple CNN layers. The dataset used by the study was another RGB-D dataset, CAD-60, composed by 60 videos with 12 different actions performed in 5 different environments. The final model aimed to generate a classification of activities of daily living (ADL) to monitor the actions of elderly people in their homes. In contrast to Li *et al.* (2017), Ercolano and Rossi (2021) support the CNN layers as spatial dependency handlers, whereas LSTM learns the pattern from sequences. These assumptions from the authors converge with previous discussions in the literature review. The authors consider that the final CNN-LSTM model achieved and outperformed the state of the art with 98% of precision and 97% of recall.

Another matter to be discussed regarding the presented machine learning models is that their authors utilised widely spread datasets for object detection or gesture classification to attest the result of their models. Common Objects in Context (COCO), presented by Lin *et al.* (2014), and PASCAL Visual Object Classes (VOC) - presented as an annual benchmark challenge by Everingham *et al.* (2005) - contain millions of labeled instances in images and these are used for object detection model benchmark comparison. For sign language detection, there are studies that developed sign language datasets for application in algorithms, like the dataset created by Martínez et al. (2002) for American Sign Language (ASL). In a context in which the image inputs cannot come

from existing datasets, the limited data problem requires solutions to avoid overfitted models of DNN (Shorten and Khoshgoftaar, 2019). Krizhevsky, Sutskever and Hinton (2017) define two relevant label-preserving methods for data augmentation that increase the training performance of input images in a deep CNN model: image translations alongside horizontal reflections and Principal Component Analysis (PCA) - the latter is a concept introduced by these authors that consists in changing the RGB values of an image by adding multiples of its principal components.

By assessing in this review previous achievements of authors that aimed to classify gestures and sign language vocabulary, several combinations could be noticed in the different stages of a project such as this one and many of the presented solutions achieved excellent results in their respective contexts. Utilising the correct permutation of these elements tends to take one closer to the objectives of this research, which is to create a Lámh sign language detection and classification. The main stages that are common amongst all the mentioned studies are the data acquisition, data preparation and the machine learning application that generate the predictions as a consequence.

Moreover, thoroughly inspecting each stage, there are opportunities for them to be tuned. In terms of data acquisition, the data can be obtained through sensors, or RGB-D cameras like the Microsoft Kinect or regular RGB cameras, being the latter the most popular due to its accessibility (Zhang *et al.*, 2019) and it was be the tool implemented by this study since the result is expected to be used by different users that will not have access to more expensive and complex alternatives like RGB-D.

Even when the data comes from existing data sources, there is still possibility to improve the image perception by the machine learning with the application of bounding

boxes for object detection, available for different computing languages, or landmark detection performed by pre-trained frameworks like MediaPipe (Lugaresi et al., 2019) or different techniques of motion representation like MHI in case of dynamic actions. MediaPipe would bring benefits in relation to a pixel array approach as indicated by Nunnari, F., España-Bonet, C. and Avramidis (2021) – the data capturing would not be influenced by the signer's features or clothing; the light would not play a relevant role; the coordinate location of the poses would be enhanced; the fact that the skeleton data is two dimensions smaller than picture based data tends to require a smaller neural network and, consequently, more efficient and faster.

Data preparation also contains variables within its sub-stages that can be taken into consideration and may affect the final result: normalization, picture dimensions and data augmentation. These elements can define the success of models in some cases. Some CNN models only accept certain picture dimensions, for example, and the data augmentation may be crucial to avoid overfitting.

Finally, the choice of the machine learning model is another core matter. This review has presented efficient solutions involving machine learning models that are not within the deep learning classification, mainly SVM. Deep Learning also presents options that can even be merged in the same model, being the 2D CNN, 3D CNN, 3D Densenet and LSTM (from RNN), HMM was also applied with neural networks in order to improve existing models. When applied in correct cases, these Deep Learning solutions seem to address greater datasets with increased classes to be predicted, and good results were obtained by applying 3D Neural Networks or LSTM to predict sequential frames, which is

a requirement in this study since Frizelle and Lyons (2022) identify Lámh and other KWS as dynamic signs alongside the spoken word.

Therefore, MediaPipe holistic model was applied to the produced videos of Lámh gestures in order to enhance the human landmarks. This framework was chosen rather than bounding boxes since it tends to optimize the process for videos, instead of the bounding box frame by frame classification. Also, a CNN models were applied to the prepared data in order to achieve the main research objectives. If the predictions don't achieve considerably good accuracy, LSTM models were also be applied in separate models and integrated with CNN layers as well. Finally, the models were compared to SVM in order to promote a relevant conclusion regarding the role of neural networks in comparison to another Machine Learning classifier, since CNNs and SVM seem to be competing alternatives in the analysed state of the art studies, being LSTM also a relevant alternative in terms of sequential frames and videos.

Investigating the core studies related to the Lámh language, it can be noticed that constant training is considered crucial by the authors. Thus, this study proposes the creation of a Lámh language detection tool, which is unprecedent, utilising machine learning techniques in order to efficiently recognise Lámh language signs. By applying all the given concepts and theories to this project, DNN models were be explored through TensorFlow and Keras in Python language, alongside the library OpenCV (Open Source Computer Vision), which brings important tools for real time detection (Bradski and Kaehler, 2008). All these elements were applied with a methodical data acquisition detailed in the Primary Research, Methodology and Ethics chapter and methods to avoid overfitting, in order to distinguish the Lámh signs and create an accurate and generalised

detection and classification of the signs in real time for all possible circumstances. Hence, this Lámh sign detection tool could potentially pave the creation of an interface with users, in order to generate a learning and reinforcement tool for communication partners, being teachers, nurses, special needs assistants, speech therapists or the Lámh user's family members and friends. Consequently, the promotion of the constant training of Lámh signs would promote the inclusion of individuals in natural settings (Dolly and Noble, 2018).

6. Validity

As per Machine Learning benchmark measurements, the validity of the final result was given by the use of the loss function of the model, which indicates the discrepancy between the prediction and the actual classification of the labeled gestures, and the accuracy of the model - sum of true positive and true negative predictions divided by the sum of all predictions. These measures was applied to the split training and testing data and, if their loss and accuracy did not demonstrate a behaviour of overfitting or underfitting. If this criteria is not attended, the previous steps carried out by the study must be reviewed with the possibility of requiring further studies

Once the model was validated in terms of having no overfitting or underfitting, then the most appropriate and generalised models were selected for a real time detection procedure described in the chapter Primary Research, Methodology and Ethics, and the actual accurate detections of the best models were counted and finally compared amongst themselves so a single best model could be chosen.

7. Sampling Strategy

The sampling strategy of this research is judgment sampling. Once the population of this study is a potential Lámh user, which ultimately could be anyone, the choice of this non-probabilistic and qualitative sampling method is related to the nature of the expected outcome and its expected purpose: to accurately detect Lámh language signs. Under this main objective, the source of the data must come from experts of the matter (Etikan, Abubakar Musa and Sunusi Alkassim, 2016), that, in the context of this research, have a holistic knowledge of the Lámh language and can reproduce the signs precisely. The characterization of judgment sampling resides on the fact that an expert of the matter was selected as the representative group by the author.

Thus, videos and pictures were produced using as reference the gestures and signs by a professional that utilises Lámh language on their daily basis. The specialist voluntarily agreed to share their knowledge and to be used as the sources for the image production of this study. This is detailed in the Primary Research, Methodology and Ethics chapter.

8. Primary Research, Methodology and Ethics

This study opted for the visual-based data acquisition rather than the sensorial-based approach in accordance with the matters raised in the literature review: a visual-based approach is more accessible for a communication partner than a model based on sensor inputs. Considering a visual-based data input for the model, as previously discussed in the objectives and in the literature review, due to the specificity of the Lámh sign language, the images that show the reproduction of the signs must be generated by the author.

Therefore, a primary research was carried out to produce the required data. This research was assisted by a volunteer that works as an aim support worker with an autistic child on their daily routine. Moreover, the representative of Lámh Development Office of an Institute of Technology in Ireland contributed to this research by giving access to the complete online Lámh content. As a consequence, the volunteer could consult the online content provided by the representative of Lámh in case they wanted to confirm they were performing the signs correctly by following the steps provided in the videos and written instructions.

In relation to the words chosen for the project, the twenty signs are based on the top twenty words of the core Lámh vocabulary for children with Down Syndrome in primary schools in Ireland developed by Frizelle and Lyons (2022). These words can be found in Appendix A. This methodology would be closer to the in-depth interview concept, although it is not directly structured in a question and answer structure.

When dealing with such primary research, it is crucial to consider the ethical implications involved: the dataset generated cannot be shared out of the academic environment in which this research is inserted. Firstly, the identity of the volunteer should

be protected since it is their right to remain anonymous and it is also the volunteer's right to entirely understand their role in the study, the final objectives of the research and the type of data that was going to be generated with their help. An advantage of the skeletal data in relation to the picture data in the ethical perspective is that there is no risk of exposing the volunteer's visual identity. Even though, the volunteer signed a Participant Consent Form confirming the scope of the project was properly explained to them, agreeing with the proposed methodology and attesting their awareness that the final data generated would not store their image and their names would not be shared.

Secondly, the copyrights of Lámh belong to the Health Service Executive (HSE) in Ireland. Therefore, the Lámh representative was reached out to be made aware of the existence of this research and it was agreed that the ownership of the final machine learning code and the dataset capturing the signs made by the volunteers will not be claimed by the author as a strict condition to progress with the research. Moreover, an NDA was signed by the author of this study to confirm that the complete Lámh lecturing material shared with the author by the Lámh representative will not be shared with external people that are not related to this study since the content is an intellectual property that belongs to Lámh - Communication Augmentation Sign System Ltd.

The movements of the volunteer were captured in order to obtain forty five videos with 40 valid frames per volunteer per sign. Since the proposed model aims to distinguish twenty different signs, 36,000 frames were generated as primary data. It is important to enhance that the forty five videos of each sign were equally split in three categories: 15 videos in which the volunteer was between 2.5 and 3 meters away from the laptop camera, 15 videos in which they were between 1.5 and 2 meters away from the laptop

camera and 15 videos in which they were less than 1 meter away from the laptop camera. This is due to the fact that the ‘z’ variable, or the depth of the landmark in the video, is one of the variables calculated for pose, hand and face landmarks of the MediaPipe library. Moreover, the distance to the camera changes the pose landmarks that would show up: the closer distance to the camera would eventually not capture the elbows, whereas the medium distance would get elbows at all times and the farther distance to the camera would also capture the hips. Thus, different depths needed to be recorded per sign in an attempt to create an augmented strategy that would make the data more comprehensive for a final model. As proposed by Nunnari, España-Bonet and Avramidis (2021), the augmentation of skeletal motion data would be related to a variation in the camera framing conditions. Therefore, methods of image rotation, image flip and PCA colour augmentation were not applicable in the chosen context.

As a consequence of the skeletal data collection, a numpy array was concatenated based on the landmarks generated by MediaPipe Holistic for the left hand, the right hand, the pose and the face. The pose generates 33 landmarks and each one of them contain the x, y – which are automatically normalized by the MediaPipe function in accordance to the width and height of the frame – and z coordinates and the visibility, which varies from 0 to 1 in a non-binary way; the left hand and the right hand generate 21 landmarks each with x, y and z coordinates; the face generates 468 landmarks with x, y and z coordinates. Flattening the appended arrays of all landmarks and concatenating the resulting arrays of each body component, the final result is a single dimensional array of 1,662 inputs. As previously introduced in the literature review, this brings an optimization of the machine learning models in comparison to arrays originated from image pixels, which are two

dimensional for black and white images and three dimensional for RGB images. Furthermore, the calculation of normalized x, y and z coordinates given by MediaPipe save several data preparation steps listed by Ercolano and Rossi (2021), like establishing coordinate references and normalizing the obtained coordinates.

The face landmarks were included in the calculations because, although the signs are mainly generated by the movement of arms, hands and fingers, Frizelle and Lyons (2022) enhance that KWS languages like Lámh differ from natural sign systems because the spoken word is combined with the gestural sign. Therefore, the movements of the volunteer were captured alongside the spoken word, turning the face landmarks important variables of the applied methodology. It is also important to enhance that, in the case of Lámh language, for the single hand signs, there is a determined hand to perform the sign, this means that a right hand sign should not be performed with the left hand. This reduces the permutations required to perform each sign. This makes a data preparation step of skeletal data described by Ercolano and Rossi (2021) as symmetrization inappropriate for this context since mirroring the one hand signs would lead to incorrect signing.

The programmed integration between Open CV and MediaPipe Holistic allowed a real time landmark capture per frame. The latter used the default inputs from Media Pipe Holistic (min_detection_confidence=0.5 and min_tracking_confidence=0.5). The 1662 landmarks were stored in separated folders for each of the 40 frames per sign. The final data was stored in 20 folders – one folder for each sign – and each of these folders contained 40 other folders – one for each frame – with numpy arrays with 1662 landmark numerical representations. Looping through these folders and appending the numpy arrays saved, the result is a three dimensional array A x F x L in which A stands for the

number of actions – in this case the number of different signs times the number of times each sign was repeated by the signer –, F is the number of frames and L is the standard number of landmarks of MediaPipe Holistic. Therefore, 900 x 40 x 1,662.

If the manipulated data were associated with RGB pixels represented by arrays, the Landmark dimension would be replaced by 3 other dimensions: W x H x C, in which W is the width of the image, D is the height and C are the 3 RGB colours. Moreover, the consequence of the reduction of dimensions of the skeletal data when comparing to pixel based data becomes evident when building a Convolutional Neural Network, for example. In this study, a 1D CNN is required instead of a 3D CNN (RBG pixels) or a 2D CNN (black and white pixels) which were explored by other studies.

In this sense, the skeletal model is less dependent of the quality of the captured image and the features of the different signers, and more dependent on the efficiency of the skeletal landmark capture. Putting into a context of limited resources, volunteers and time, the skeletal data becomes then pivotal for one's achievements in such computer vision study.

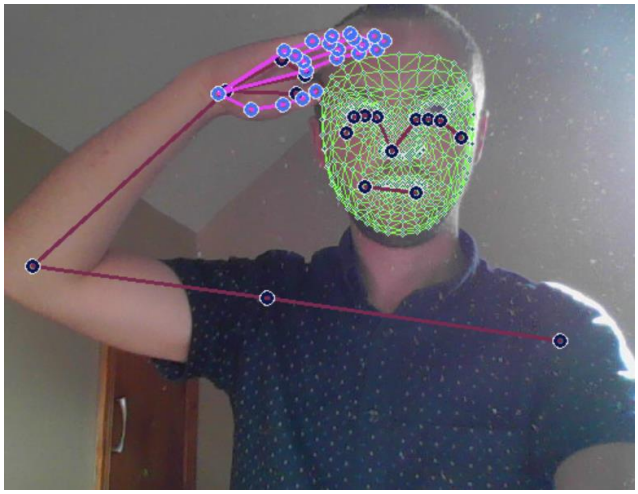

Figure 1: Demonstration of data collection integrating MediaPipe and OpenCV – 'Hello' Lámh Sign (Source: Personal Collection)

Once the data was collected following the procedures described, several models of machine learning were applied based on the successful techniques of sign language detection assessed and compared in the literature review. Support Vector Machine was applied with cross validation function of 10 folds and combination of different hyperparameters: kernels poly, RBF and sigmoid – which are non-linear kernels; C 0.1, 1, 10 and 100; gamma 10, 1, 0.1, 0.01, 0.001. This means that the cross validation function applied 60 combinations of Support Vector Machine hyperparameters for 10 folds, leading to 600 fits, in order to determine the most appropriate in terms of the accuracy in relation of the increase of variable C, which is directly related to the error acceptance by the SVM model. The data was split in 80% for training and 20% for testing. To apply the previously mentioned 3 dimensional array 900 x 40 x 1662 (A x F x L) to the Support Vector Machine, a vectorial transformation was required. This Machine Learning Model would accept the following format: A x V in which A is, as defined before, the number of actions – in this case the number of different signs times the number of times each sign was repeated by the signer –, whereas V is the vector that represents the whole action concatenating the 40 frames per action together, so it would be the presented F (number of frames) times L (number of landmarks). Using the reshape function of the numpy library, the data 900 x 40 x 1662 was transformed into 900 x 66,480. No transformation was required for the labels, as they remained with the encoded 900 labels between the range 0 and 19. After all the cross validations were carried out, the average results of each hyperparameter combination was stored in a csv file for further plots. The Support Vector Machine models were compared using the mean accuracy score – which is a standard choice in the scikit-learn library for classification problems.

Moreover, the Support Vector Machine most appropriate choice was compared with different deep neural network models applied to the data. These neural networks were based either in Convolutional Neural Network 1D with Max Pooling layers, or Long Short Term Memory or a combination of both as some successful models enhanced in the Literature Review achieved good results by adding both layers to their architecture when classifying sign language gestures or other sequential actions. Since there is no consensus regarding the best architecture of Neural Networks, the first Neural Network models applied started with the simplest architecture, with one layer of CNN or LSTM – 32 nodes respectively – and, once these were tested, other models were tested by increasing the number of layers one by one for every new model so the impact of additional layers in the final accuracy and loss could be verified. It is important to enhance that, except for the output layer, all the other layers used *relu* as the activation function since it is considered a standard activation function as defined by Alzubaidi *et al.* (2021). The activation function of the output layer was softmax as this is the acvation function of multiclassification Neural Networks.

For every new layer added, the latter had the double number of nodes of the previous layer. In the case of overfitting models, dropout layers of 20% have been added to the models in variable positions of the architecture in order to identify whether the overfitting aspect would be controlled. For the Neural Network models, the data was also split in 80% for training and 20% for testing. All Neural Networks were compiled with 'accuracy' as the metric – Grandini, Bagli, and Visani (2020) define accuracy as a popular multiclass metric and is used when the weight of each class is not relevant, which is the case –, 'categorical_crossentropy' as the loss since the labels were one-hot encoded –

converging with the *softmax* activation function in the output layer – and Adapative Moment Estimation (Adam) as the gradient optimizer, which was originated in the study by Kingma and Ba (2014) and is widely used in machine learning models that deal with large datasets. As the authors define, Adam joins two advantages of other gradient optimizers – it deals with sparse gradients and is able to manage non-stationary objectives. To fit the model, the only parameter changed was the number of epochs. Some experimentations were firstly applied to determine an appropriate value for the number of epochs and it was defined that 500 epochs allowed the development of all the models to the point in which the behaviour of the models and their trends could be correctly verified.

After the best models were chosen, they were experimentally applied using again Open CV and Media Pipe Holistic exactly as the data collection setup. However, instead of storing the arrays, they would be appended and transformed to be submitted to the chosen models in order to generate a real time Lámh language classifier. The criteria adopted in this study was the threshold of 0.9 and each prediction that would pass the chosen threshold would be stored in an array. If the last 10 predictions of the array corresponded to the same label, the classification would be displayed on screen. This measure allows the displayed predictions to be more stable, since they need to fulfill a high threshold requirement and must remain constant for at least 10 accurate predictions. As enhanced previously, the design of the written content on the Open CV page was not a concern as it would deviate from the main objectives of the research. The main models were applied in this real time context so their final performance could be assessed, adding an additional analysis layer to the experimentation.

The SVM interface and the Neural Network interface in OpenCV were conceived with a difference as the *softmax* output of the Neural Network architecture led to a real time probability display on screen in which the user is able to follow how the probabilities are distributed for each one of the twenty words as colorful bars representing the words change their sizes proportionally to the change of the probabilities outputted from the Neural Network model and, when the model achieved the threshold and twenty sequential prediction requirements, the classification would finally be displayed on the top of the OpenCV window. In contrast, the SVM model only has a single classification as output and therefore the probability interface was removed and only the final result after the requirement fulfillment was displayed on the top of the OpenCV window. The signer executed the same signs again in real time and the accuracy of the classification of the final models was analysed. For each model, the signs were categorised in three statuses: ‘Detected’, ‘Detected with reservations’ and ‘Not Detected’. Additional comments were added for the signs that were detected with reservations and not detected in order to compare the models and the dataset together, since the dataset was generated by this research as primary data and it was not validated in previous studies. The real time detection also assessed the behaviour of the models when the designer was not carrying out any apparent sign.

It is important to highlight that this last step of applying the model in practical real time detections and taking conclusions from this practical experiment is not something common to be observed in discussion sections from articles of computer vision as they tend to stop at the confusion matrixes of the split test data and assess where the false positives and false negatives were observed. However, going beyond the training and

testing relation is important considering the proposed research question and the primary objective: ultimately, a successful model applied in practice would be an ideal tool for Lámh language training purposes. Therefore, considering that the data was also generated by this research itself and was not validated in previous studies, successful training and testing relations are not enough to attest the success of the machine learning model. By applying the models in a real time detection, the generalization of the model can be really tested in an actual execution. Notes were taken for each sign done in a certain distance to the camera and in different angles of the camera in order to exhausted the real possibilities.

All the coding of the research was executed in a windows laptop Intel(R) Core(TM) i5-8250U CPU 1.60GHz with 8GB of RAM. The study did not count on a hardware with GPU to carry out any step of the methodology.

9. Timeframe and Supervisor Meetings

The tasks of this project have been fulfilled within the stipulated time frame, from 16th of May until 19th of August. Hence, the tasks have been distributed chronologically based on the proposed secondary objectives established in the Objective chapter as per Table 1.

Table 1: Gantt chart representing time frame of each task

	Week **Starting on 16/05/2022 in week 1** **Ending on 19/08/2022 in week 14**													
Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Data collection														
Set data augmentation strategies														
Apply different Machine Learning models and choose the most generalised and accurate one														
Generate final artifact														
Elaborate the written discussion and conclusion of the project														

(Source: Personal collection)

The meetings with the supervisor were carried out on the days 1st of June, 8th of June, 22nd of June, 18th of July and 08th of August. In the meetings, the research question,

problem definition, novelty and methodology of the research were the main themes discussed in order to successfully fulfil the tasks within the proposed timeframe.

10. Results and Discussion

After the methodology was applied to gather the dataset, the final document containing the numpy arrays contained 562 Megabytes of data. As discussed in the Primary Research, Methodology and Ethics chapter, the main augmentation of the data was made in the development of the data itself when changing the distance of the signer to the webcam camera in 3 different ranges in order to variate the depth of the data but also slightly change the x and y coordinates, as well as bringing different pose landmarks to the camera frame.

The first Neural Network applied to the mentioned collected data contained a single CNN 1D layer with 32 nodes, followed by a Max Pooling 1D layer, a flattening layer and dense layer with 32 nodes. This first model generated a clear overfitting relation between training and testing results, which can be observed in both the accuracy and loss plots. The accuracy ended with 77.22% for training and 39.44% for testing, whereas the loss ended 0.61 for training and 4.326 for testing.

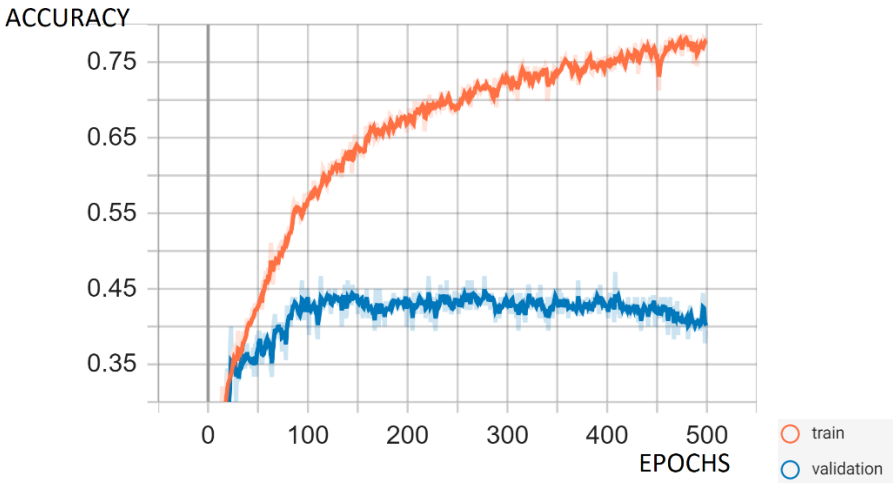

Figure 2: Single CNN (32 nodes) – Training and Testing Accuracy (Source: Personal Collection)

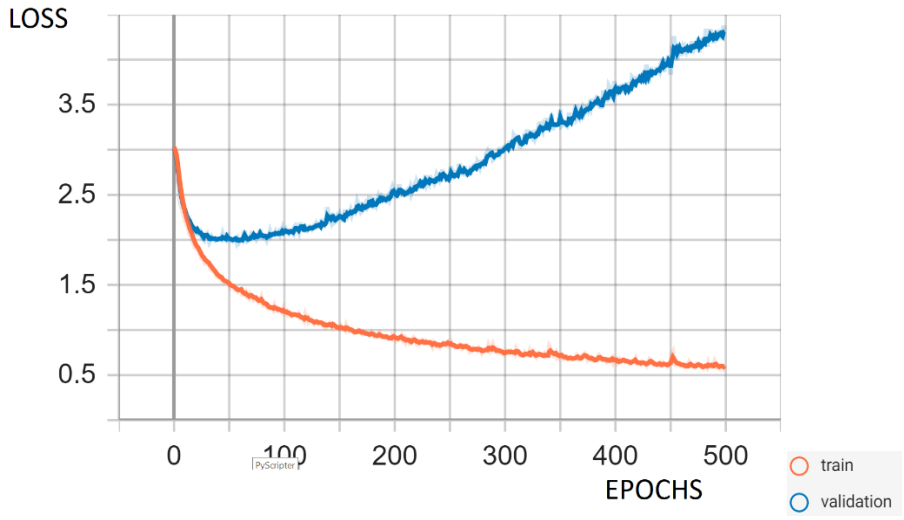

Figure 3: Single CNN (32 nodes) – Training and Testing Loss (Source: Personal Collection)

When the second model was tested adding one more layer of CNN 1D with 64 nodes, the architecture was: CNN 1D with 32 nodes, Mac Pooling 1D layer, CNN 1D with 64 nodes, Max Pooling 1D layer, the flattening layer and dense layer of 32 nodes. Although it was overfitting since it could be observed a certain gap in the distance of the training results to the testing results as the epochs progressed – mainly in the loss plot –, it was perceived that both accuracy and loss showed better results with an overfitting severely more discrete. The training accuracy was 100% whereas the testing accuracy was 89.44%. The training loss was 0.003 and the testing loss was 0.487.

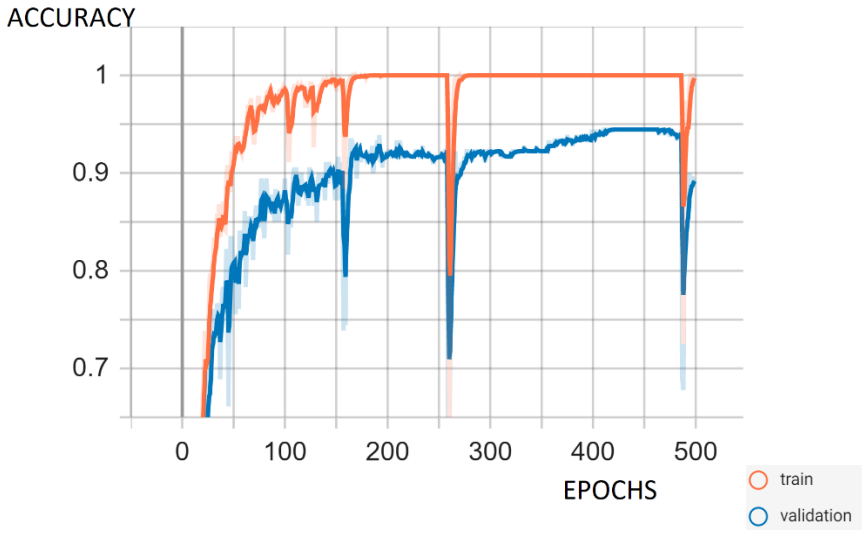

Figure 4: Double CNN (32 and 64 nodes) – Training and Testing Accuracy (Source: Personal Collection)

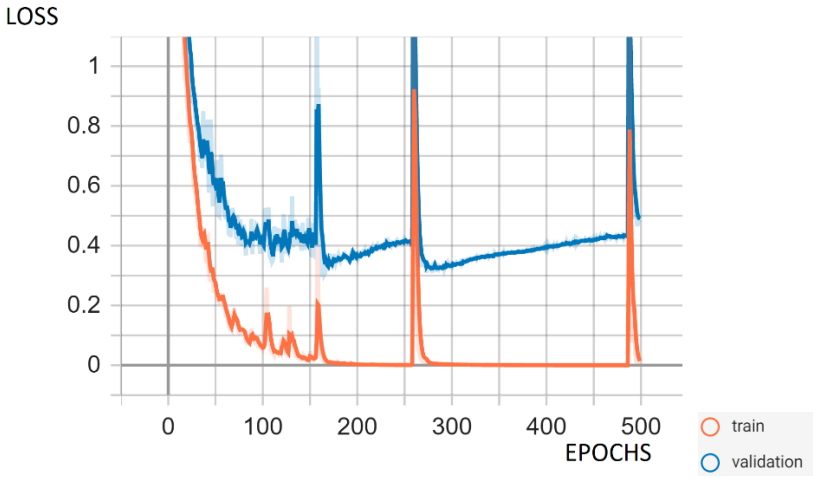

Figure 5: Double CNN (32 and 64 nodes) – Training and Testing Loss (Source: Personal Collection)

Since an improvement was noted by increasing the CNN layers but the overfitting aspect was still present, a simple method of drop out layer was added to the latter architecture. After the CNN 1D and Max Pooling 1D layers, a single Drop Out layer of 0.2 was added. This addition proved to be efficient as it narrowed the distance between training and testing accuracy, although the training accuracy and loss were still too high

and too low respectively. The training accuracy was 100% whereas the testing accuracy was 94.44%. The loss values were more discrepant as the training loss was 0.0002 and the testing loss was 0.3416. The effect of the dropout layer is noticeable on the plots where it can be seen a constant drop of the accuracy values and a constant sharp increase of the loss during the epoch progression as an attempt to control to overfitting aspect of the model.

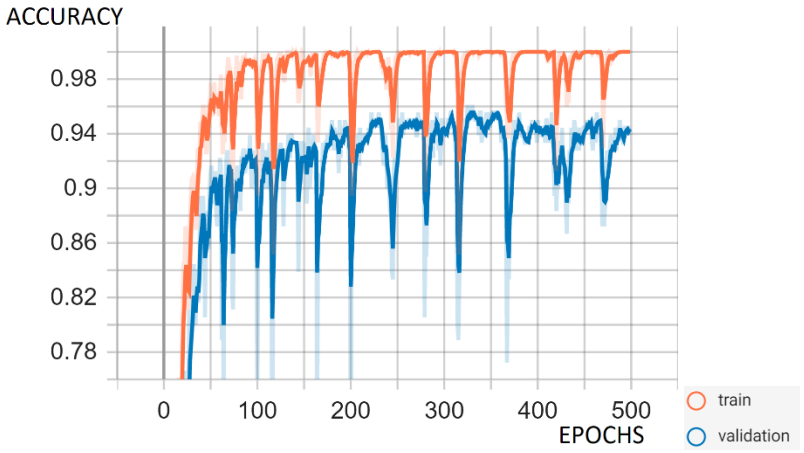

Figure 6: Double CNN (32 and 64 nodes) and 0.2 Dropout – Training and Testing Accuracy (Source: Personal Collection)

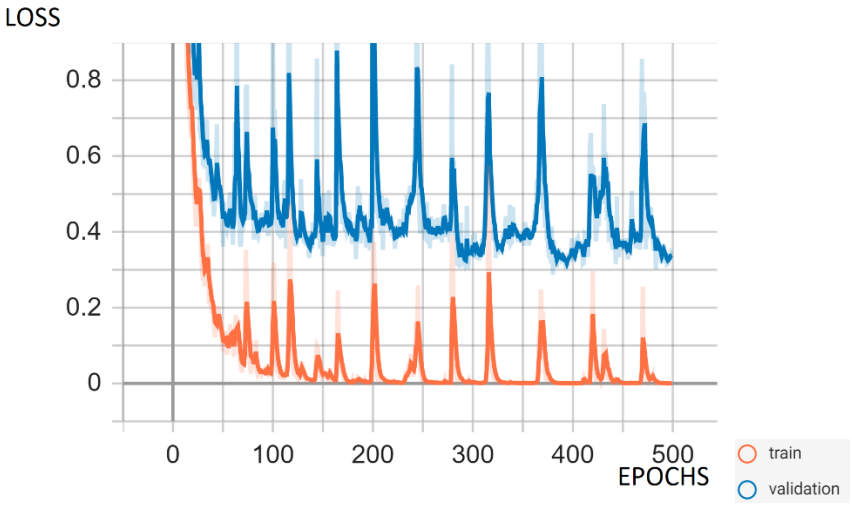

Figure 7: Double CNN (32 and 64 nodes) and 0.2 Dropout – Training and Testing Loss (Source: Personal Collection)

Finally, a last CCN model was attempted adding a third CNN 1D layer with a Max Pooling 1D layer. The architecture was set up as CNN 1D with 32 nodes, Max Pooling 1D layer, CNN 1D with 64 nodes, Max Pooling 1D layer, CNN 1D with 128 nodes, Max Pooling 1D layer, the flattening layer and a dense layer with 32 nodes. Although the simple increase from 1 CNN layer to 2 improved the accuracy and loss as well as the overfitting aspect of the model, the same could not be said for the addition of a third layer with 128 nodes. The final training accuracy was 100% and the testing accuracy was 86.67%, whereas the training loss was 0.001 and the testing loss was 1.007. These results alongside the gaps showed between the testing and training curves show a model that overfits more than both the previous models with two CNN 1ID layers.

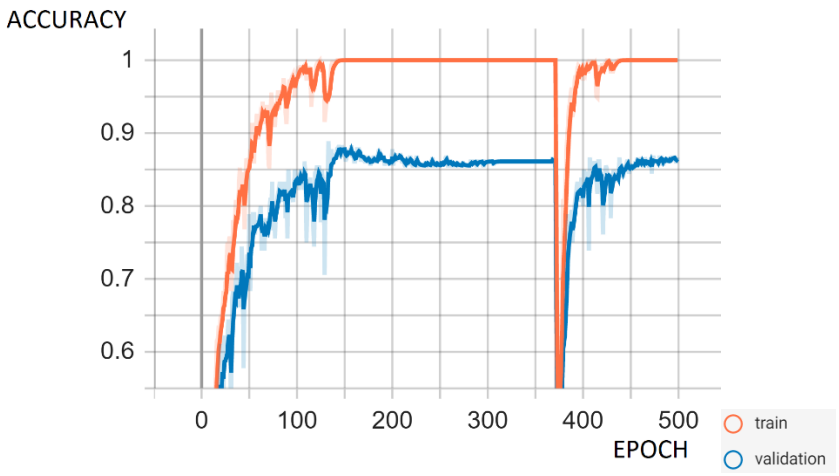

Figure 8: Triple CNN (32, 64 and 128 nodes) – Training and Testing Accuracy (Source: Personal Collection)

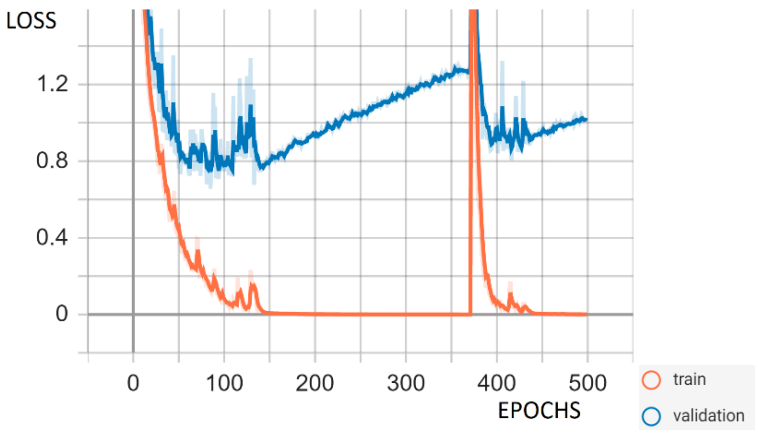

Figure 9: Triple CNN (32, 64 and 128 nodes) – Training and Testing Loss (Source: Personal Collection)

After the CNN models were analysed, the LSTM models were applied with the same training and testing data split. The first LSTM model started with a single LSTM with 32 nodes and dense layer with 32 nodes. As it can be seen in the accuracy and loss plots, as opposed to the CNN models, it clearly showed signs of underfitting with a training accuracy of 11.67% and testing accuracy of 6.11%, whereas the training loss was 3.118 and the testing loss was 3.187.

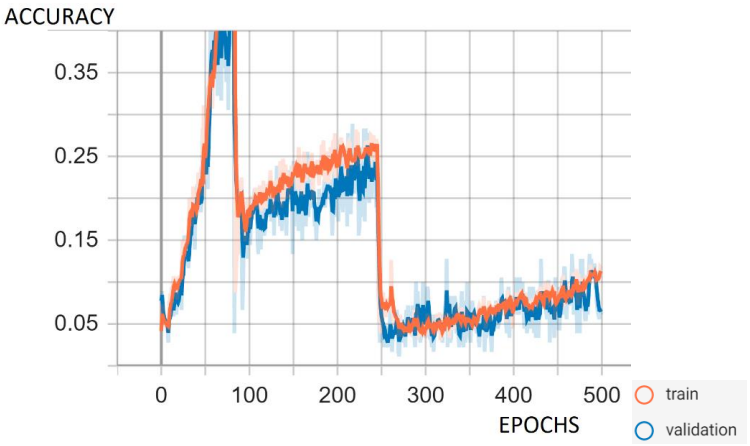

Figure 10: Single LSTM (32 nodes) – Training and Testing Accuracy (Source: Personal Collection)

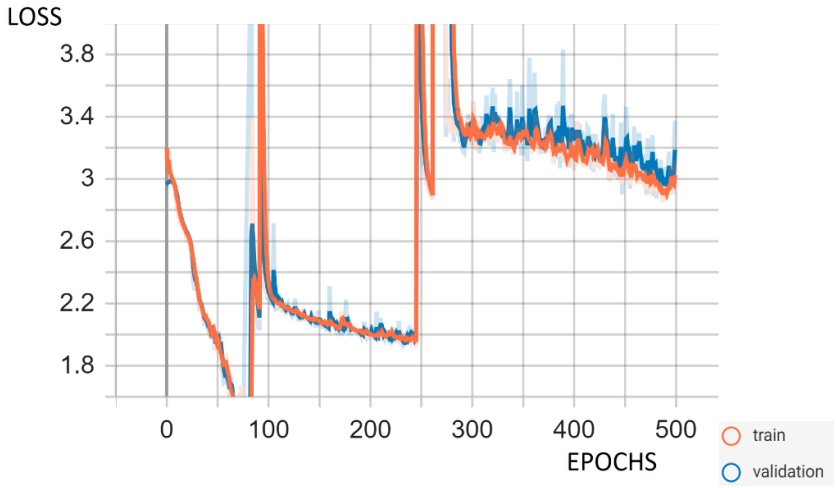

Figure 11: Single LSTM (32 nodes) – Training and Testing Loss (Source: Personal Collection)

By adding a second LSTM layer of 64 nodes to the architecture, the tight aspect between the training and testing curves from the previous model remained, but with much better results, demonstrating no signs of overfitting or underfitting. The training accuracy was 87.36% and the testing accuracy was 80.00%, whereas the training loss was 0.4069 and the testing loss was 0.6955.

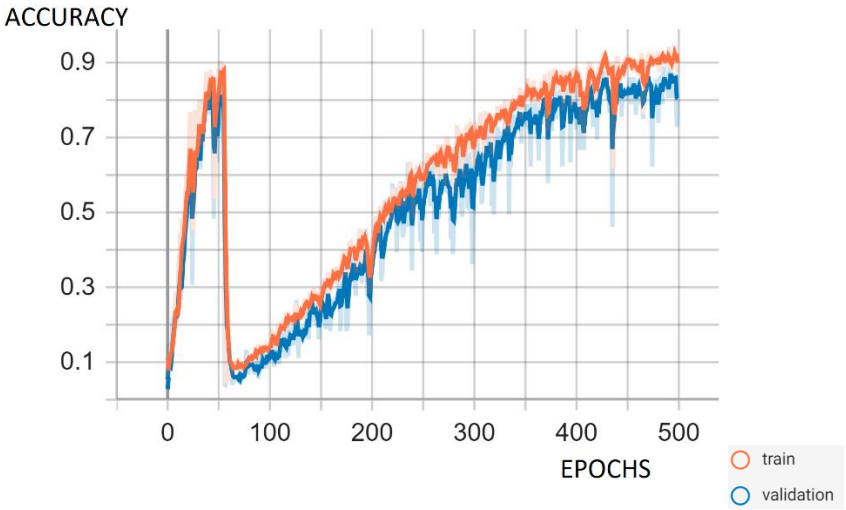

Figure 12: Double LSTM (32 and 64 nodes) – Training and Testing Accuracy (Source: Personal Collection)

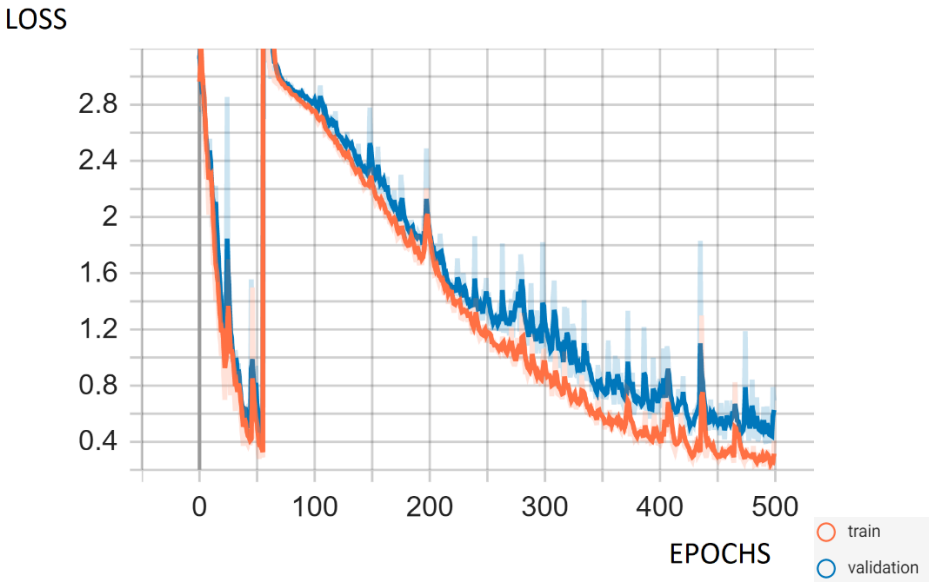

Figure 13: Double LSTM (32 and 64 nodes) – Training and Testing Loss (Source: Personal Collection)

When adding one more dense layer now with 64 nodes to the previous model, although it shows better accuracy and lower loss, the overfitting features become noticeable, mainly analysing the loss plot. The training accuracy reached 100% and testing accuracy 93.89%, whereas the training loss was 0.0025 an the testing loss was 0.3489.

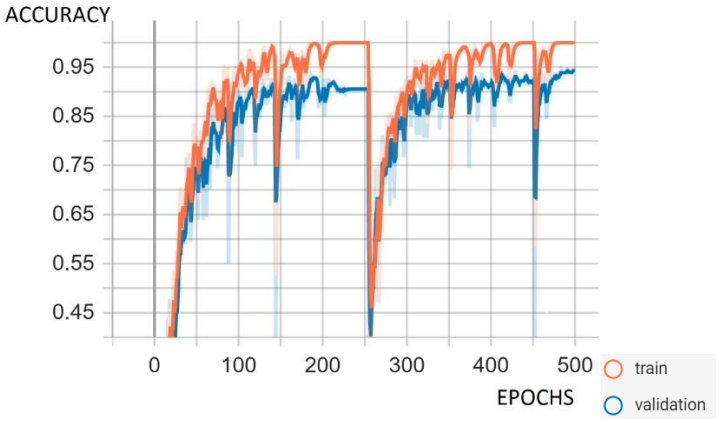

Figure 14: Double LSTM (32 and 64 nodes) and Double Dense Layer (32 and 64 nodes) – Training and Testing Accuracy (Source: Personal Collection)

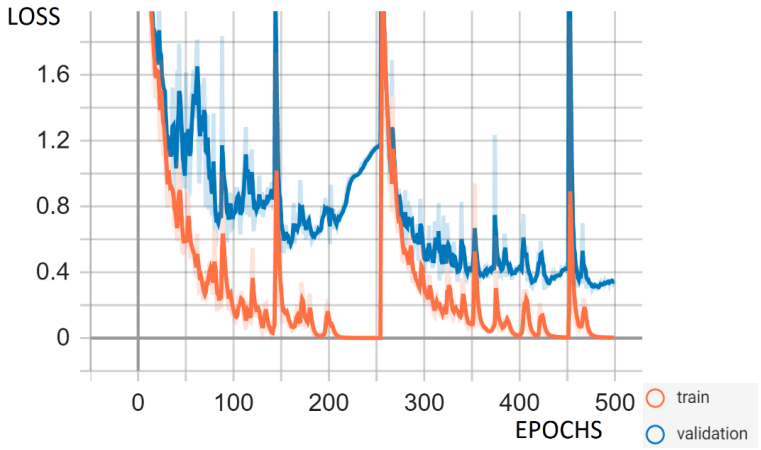

Figure 15: Double LSTM (32 and 64 nodes) and Double Dense Layer (32 and 64 nodes) – Training and Testing Loss (Source: Personal Collection)

By training a model with three LSTM layers with 32 nodes, 64 nodes and 128 nodes respectively and a dense layer of 32 nodes, interestingly the performance of accuracy and loss is even worse than the single LSTM layer previously described, demonstrating that the model became too big for the dataset in context. The training accuracy was 5.69% and the testing accuracy was 2.22%, whereas the training loss was 2.993 and the testing loss was 3.018.

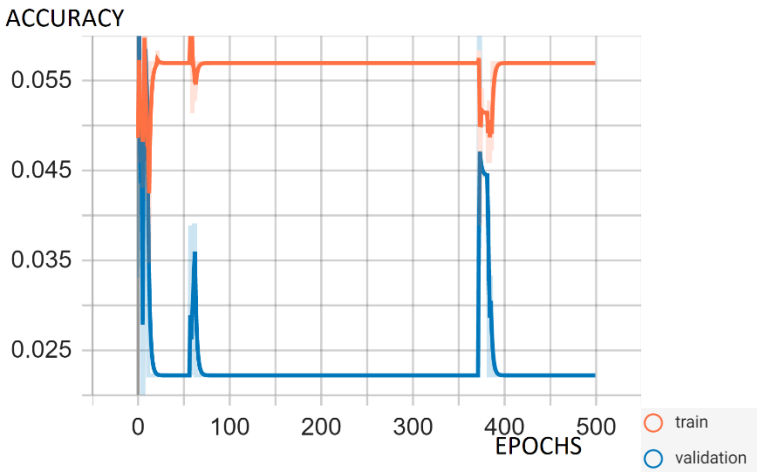

Figure 16: Triple LSTM (32, 64 and 128 nodes) – Training and Testing Accuracy (Source: Personal Collection)

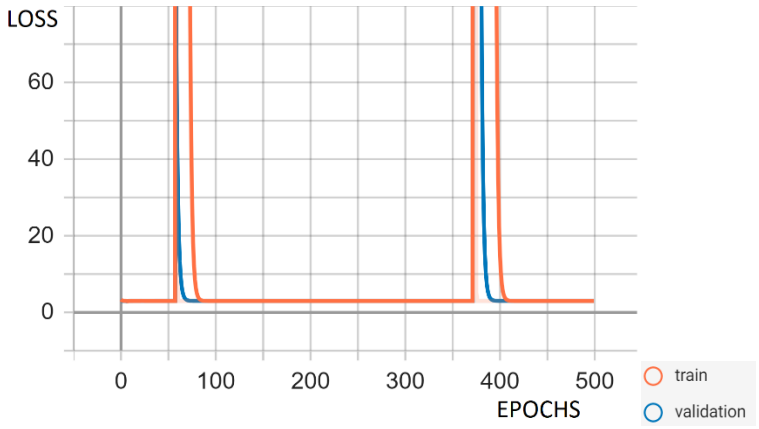

Figure 17: Triple LSTM (32, 64 and 128 nodes) – Training and Testing Loss (Source: Personal Collection)

The finds when exploring the CNN or LSTM models converge directly to the literature understanding in the field of Neural Networks and Deep Learning: large datasets require larger Neural Networks so the chosen model can perform better. This can be noticed on the fact that the CNN model with two CNN 1D layers outperformed the CNN model with a single CNN 1D layer, as well as the LSTM model with 2 LSTM layers outperformed the LSTM model with a single LSTM layer.

However, as pointed in the Literature Review in the discussions of Huang *et al.* (2017) and He *et al*. (2016), too large Neural Networks lead to an increase of the error if no solution is added to the architecture. This could be perceived, in the case of the studied dataset, when applying the third layer to the CNN model or the LSTM model. Although the CNN model with three CNN 1D layers showed better results than the model with a single CNN 1D layer, the former showed clearer signs of overfitting when compared to the model with two CNN layers, whereas the LSTM model with three LSTM layers showed sever underfitting and the LSTM model with two LSTM layers showed no sign of underfitting or overfitting, being a promising model to be chosen. This leads to a

supposition that a greater dataset could lead to an also greater architecture for a better model, also indicating that there is no definitive model or architecture in the machine learning field as a whole.

As per discussion in the literature review, the addition of the LSTM layers after the CNN layers may contribute to the overall result of the model as the LSTM would, in theory, address the temporal aspect of the data whereas the CNN layers address the spatial aspect. Therefore, different architectures involving these mathematical resources were tested. The first one was a single CNN 1D layer with 32 nodes, a Max Pooling 1D layer, a single LSTM with 32 nodes and dense layer 32 nodes. It can be noticed that the flattening layer is not required as the input of the LSTM layer requires the same dimensions as the 1D-CNN output, not requiring any vectorial transformation. It is also perceived that the model shows a configuration similar to the two layer LSTM previously applied until the epoch 200, showing close training and testing curves for both accuracy and loss, but from the 200th epoch the testing accuracy started to decreased with a training accuracy of 100% and similarly the testing loss started to increase with training loss close to zero, enhancing the overfitting behaviour of the model. The final training accuracy after 500 epochs was 100% and the testing accuracy was 95%, whereas the training loss was 0.00005 and the testing loss was 0.3162.

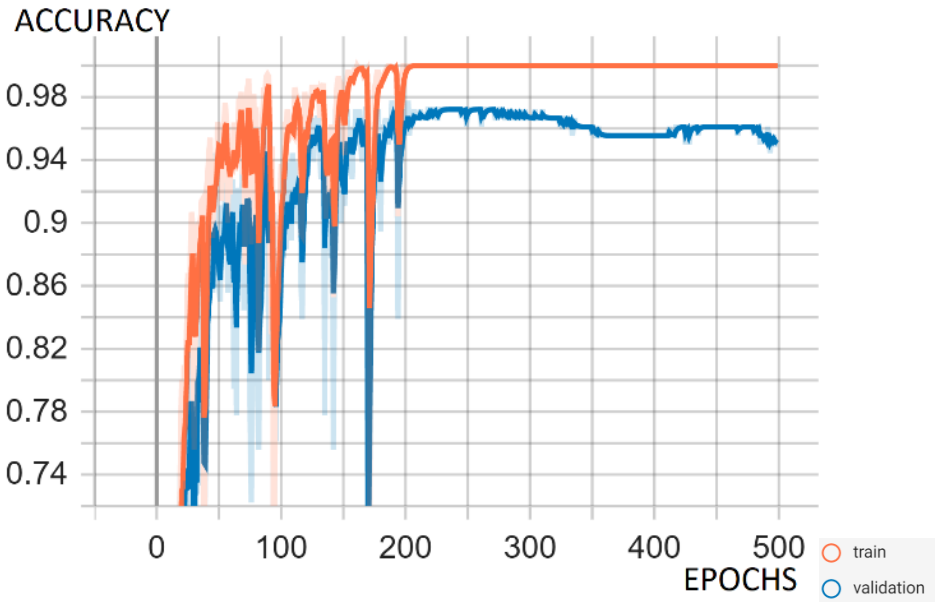

Figure 18: Single CNN (32 nodes) and Single LSTM (32 nodes) – Training and Testing Accuracy (Source: Personal Collection)

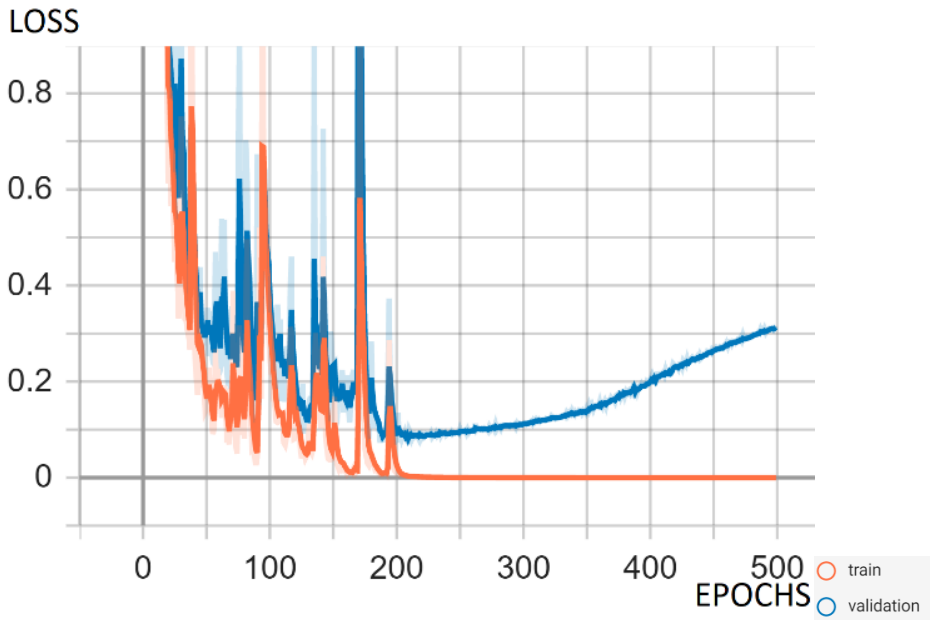

Figure 19: Single CNN (32 nodes) and Single LSTM (32 nodes) – Training and Testing Loss (Source: Personal Collection)

Considering the promising results of the previous model if the overfitting aspect is controlled, a dropout layer of 0.2 was included after the Max Pooling 1D layer and before the LSTM layer, another dropout layer of 0.2 was added after the LSTM layer and a third dropout layer of 0.2 was added after the dense layer. Although this attempt does not overfit like the single CNN with single LSTM without dropout, it still overfits and the number of dropout layers interferes in the final accuracy and loss of the model as a consequence. The final training accuracy was 79,03% and testing accuracy was 53,89%, whereas the training loss was 0.5428 and the testing loss was 2.097.

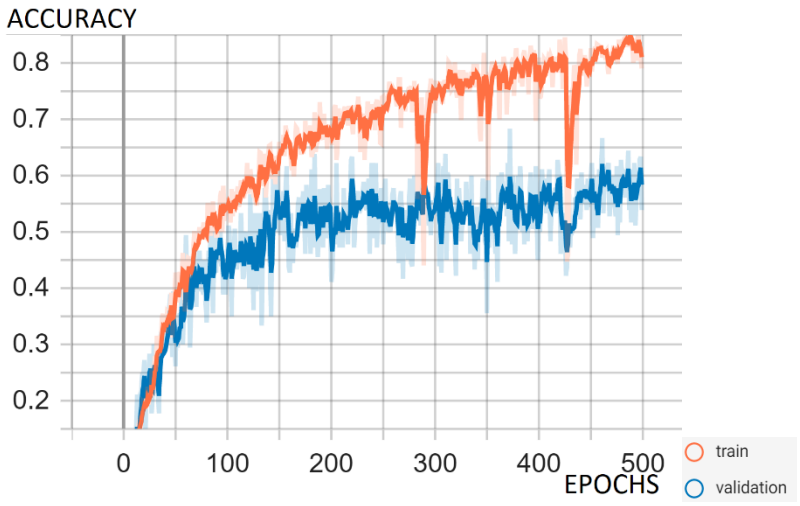

Figure 20: Single CNN (32 nodes), 0.2 Dropout, Single LSTM (32 nodes), 0.2 Dropout, Dense Layer and 0.2 Dropout – Training and Testing Accuracy (Source: Personal Collection)

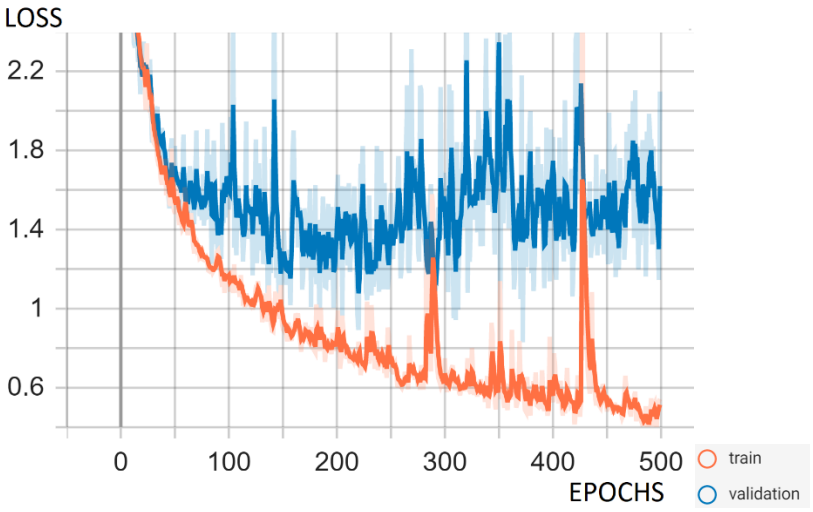

Figure 21: Single CNN (32 nodes), 0.2 Dropout, Single LSTM (32 nodes), 0.2 Dropout, Dense Layer and 0.2 Dropout – Training and Testing Loss (Source: Personal Collection)

Based on the results of the previous model that used dropout layer but they did not correct the overfitting and even compromised the accuracy and loss values, another attempt was carried out having only a single dropout layer of 0.2 after the Max Pooling 1D layer. The results, as it can be seen in the plots of the accuracy and the loss, indeed were improved and the overfitting aspect is not perceived like in the other models as the relation between training and testing did not get perceptively wider through the progression of the epochs. The training accuracy was 99.31% and the testing accuracy was 94.44%, whereas the training loss was 0.0302 and the testing loss was 0.2778.

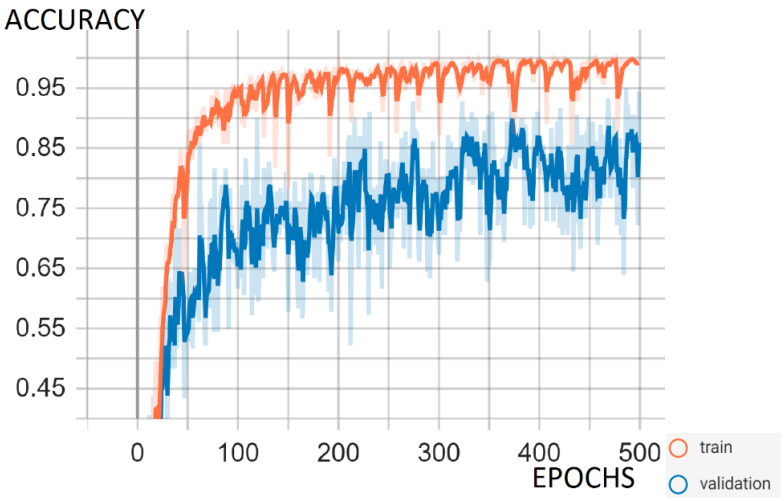

Figure 22: Single CNN (32 nodes), 0.2 Dropout and Single LSTM (32 nodes) – Training and Testing Accuracy (Source: Personal Collection)

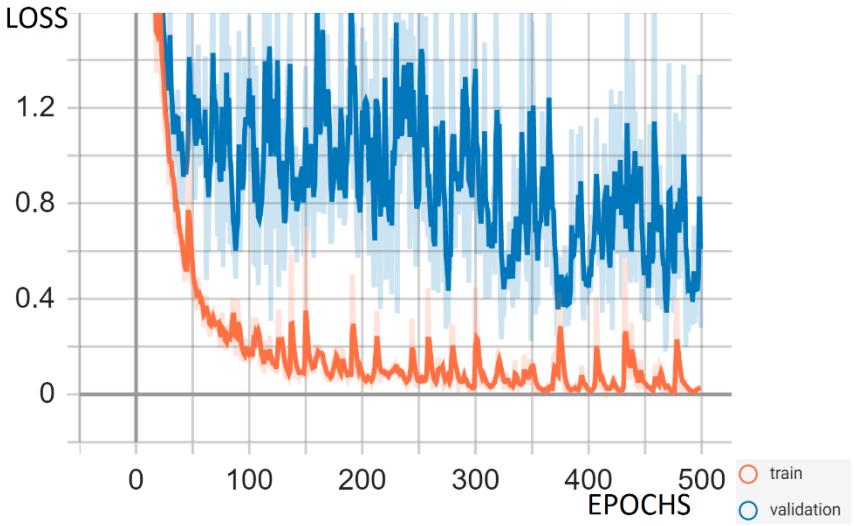

Figure 23: Single CNN (32 nodes), 0.2 Dropout and Single LSTM (32 nodes) – Training and Testing Loss (Source: Personal Collection)

Since other attempts could be carried out to achieve better results, another model was applied by adding one CNN 1D layer with 64 nodes and one LSTM 1D layer also with 64 nodes. The architecture was: two CNN 1D with 32 nodes and 64 nodes respectively and a MaxPooling 1D layer after each CNN layer, two LSTM with 32 nodes and 64 nodes

and a dense layer of 32 nodes. As it can be mainly perceived on the behaviour of the model after the 100th epoch of the loss plot, this model overfits even more than the one with a single layer of 1D-CNN with a single layer of LSTM, with training accuracy of 100% and testing accuracy of 93.89% whereas the training loss was 0.0000002 and the testing loss was 1.079.

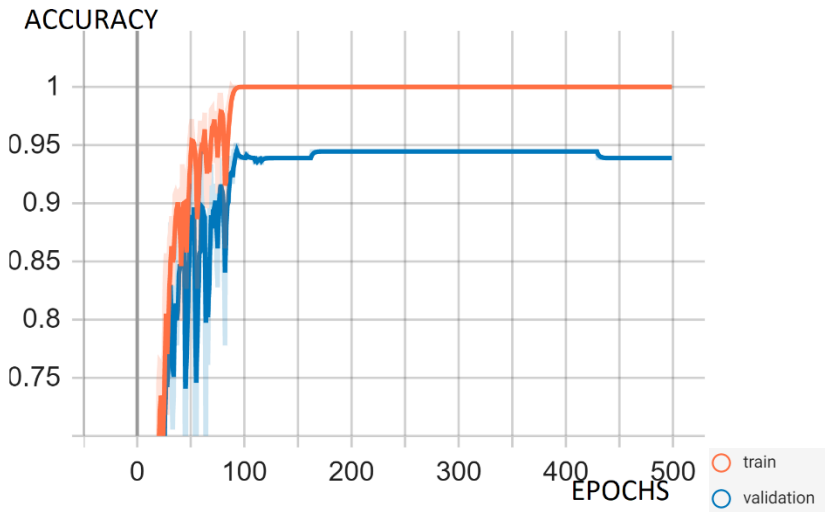

Figure 24: Double CNN (32 and 64 nodes) and Double LSTM (32 and 64 nodes) – Training and Testing Accuracy (Source: Personal Collection)

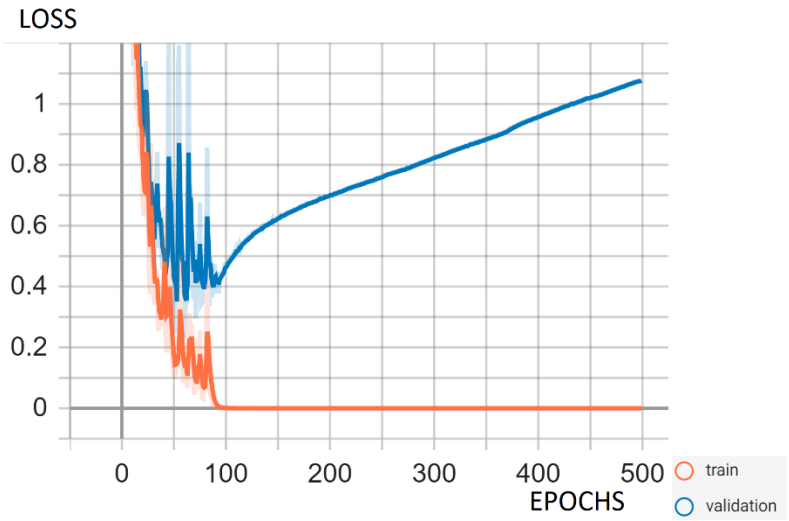

Figure 25: Double CNN (32 and 64 nodes) and Double LSTM (32 and 64 nodes) – Training and Testing Loss (Source: Personal Collection)

However, considering that a single addition of a dropout layer after the Max Pooling 1D layer was able to clearly improve the model that contained a single 1D-CNN with a single LSTM layer, the same was attempted in the architecture with two CNN layers and two LSTM layers by adding a single dropout of 0.2 between the last Max Pooling 1D Layer and the first LSTM. The final plots show balanced accuracy and loss graphs with a gap between training and testing that narrows down with the progression of the epochs, becoming a promising model to be chosen for the studied dataset. The training accuracy was 99.86% and the testing accuracy was 93.33%, whereas the training loss was 0.0035 and the testing loss was 0.1791. The results are better than another promising model, the LSTM model with two LSTM layers only, but the latter still may be considered more generalised as the relation of training and testing is even tighter.

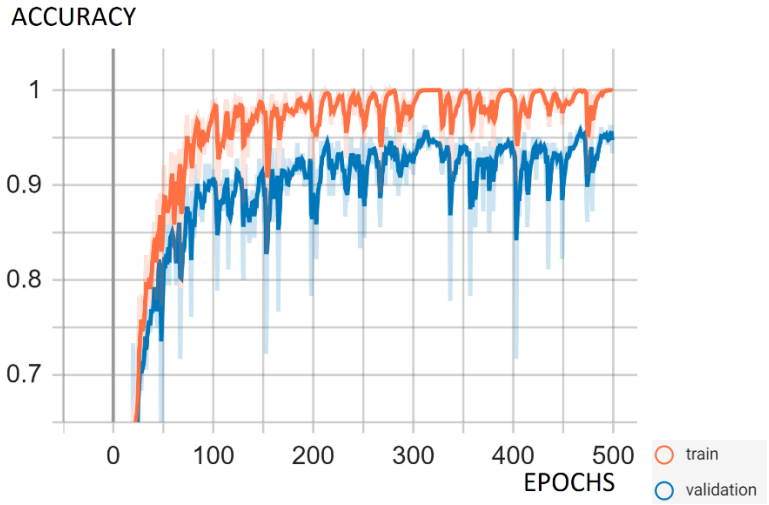

Figure 26: Double CNN (32 and 64 nodes), 0.2 Dropout and Double LSTM (32 and 64 nodes) – Training and Testing Accuracy (Source: Personal Collection)

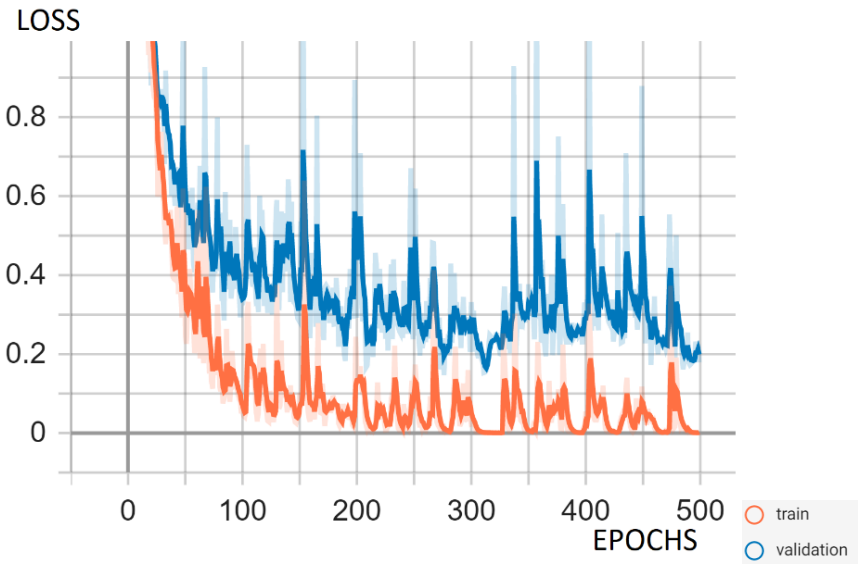

Figure 27: Double CNN (32 and 64 nodes), 0.2 Dropout and Double LSTM (32 and 64 nodes) – Training and Testing Loss (Source: Personal Collection)

After assessing different Neural Network architectures and models, Support Vector Machine was applied to the dataset after the vectorial transformation described in the Primary Research, Methodology and Ethics chapter. From all the SVM hyperparameters applied in the cross validation of 10 folds, the graphs enhance an overfitting for all the cases of poly kernel since the relation between training and testing is constant with the progression of C, being the training accuracy constant in 100% and the testing in 96.8%. The results of the sigmoid kernel show clear underfitting, with results below 10% for training and testing. In the case of the RBF kernel, most of the plots with different gamma values demonstrate an overfitting with the increase of C variable, since there is a considerable discrepancy between the training curve and the testing curve – even greater than the poly kernel ones. However, one of the cases of RBF kernel, with gamma equal 0.001, could be considered a strong candidate as one of the final models, since the mean accuracy of the cross validation narrows down as C increases, going from 0.964 of

training and 0.894 of testing with C equal 1 to 100% of training and 96.5% of testing for C equal 10, the latter values remain constant for C equal 100. This behaviour between training and testing show less overfitting features than the poly kernel accuracies, which remained constant as 0.968 for testing and 100% for training. However, this also points to a change of approach, since a very low width (gamma) for RBF points to a linear behaviour. Therefore, another cross validation was applied specifically for the SVM with linear kernel with 10 folds. The results of this cross validation are similar to the polynomial kernel: a constant training score in 100% and a constant testing score of 95.6%. It may be concluded that, although the RBF function with low gamma is similar to a the linear kernel, the plots of mean accuracy of the training and testing for both kernels showed different behaviours, being that the RBF showed less overfitting features than the SVM with linear kernel.

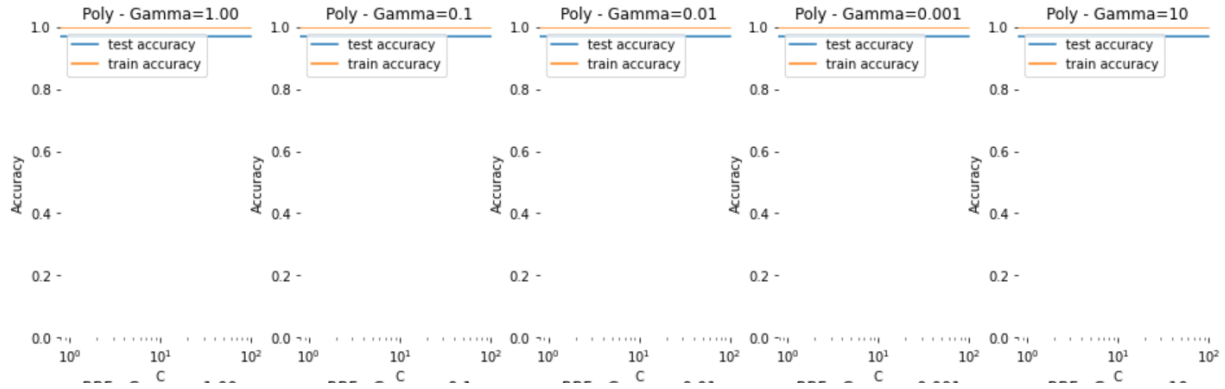

Figure 28: Poly SVM with Different Gamma Plots – Training and Testing Accuracy (Source: Personal Collection)

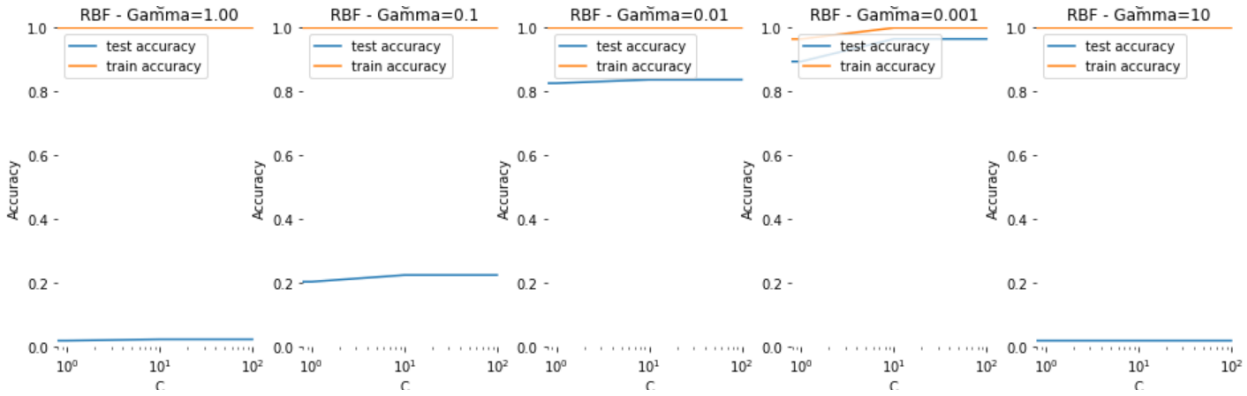

Figure 29: RBF SVM with Different Gamma Plots – Training and Testing Accuracy (Source: Personal Collection)

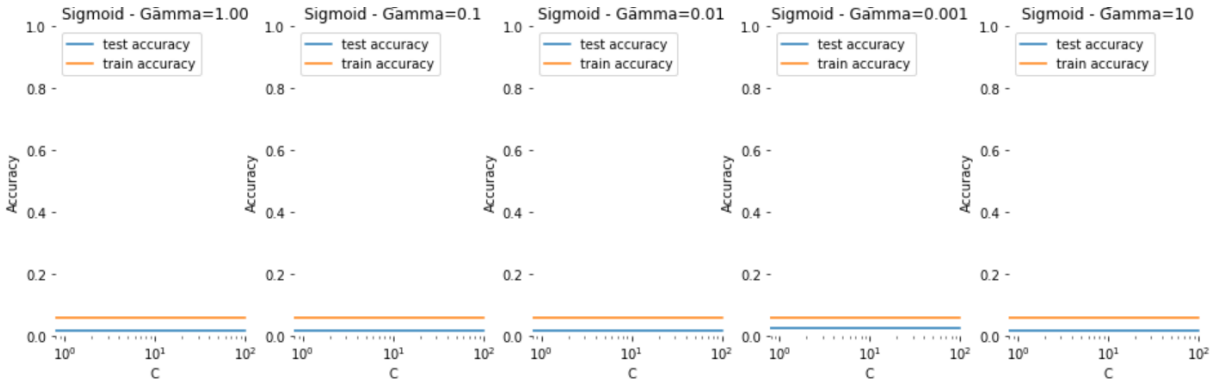

Figure 30: Sigmoid SVM with Different Gamma Plots – Training and Testing Accuracy (Source: Personal Collection)

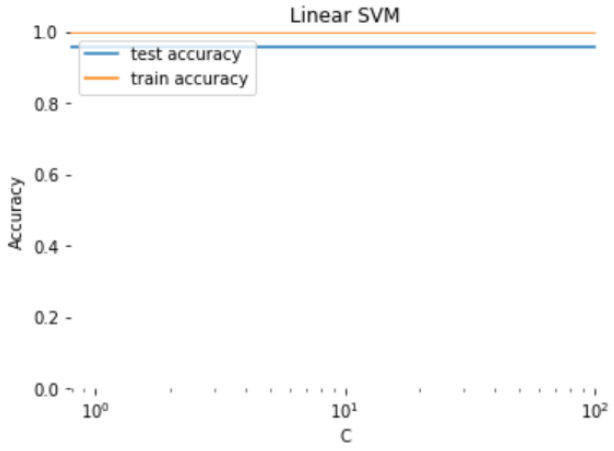

Figure 31: Linear SVM – Training and Testing Accuracy (Source: Personal Collection)

Considering all the models assessed, the SVM with ‘rbf’ kernel and gamma equal 0.001 was the model amongst the SVM ones that showed less overfitting attributes since the relation between training and testing accuracy gets narrower as the variable C increases. Therefore, this was the SVM model chosen to be compared with the Neural Networks. In regards to the Neural Networks, the model with 2 LSTM layers was the ones that demonstrated less overfitting features, the model with two CNN layers, a dropout layer and two LSTM layers also did not demonstrate overfitting as the other models and generated higher accuracy and lower loss when comparing to the 2 LSTM layer model. Therefore, both were selected amongst the Neural Network layers.

Table 2: Accuracy and Loss of the selected Machine Learning Models

Model	Accuracy Training	Accuracy Testing	Loss Training	Loss Testing
Double CNN + Dropout + Double LSTM	99.86%	93.33%	0.0035	0.1791
Double LSTM	87.36%	80.00%	0.4069	0.6955
SVM RBF Gamma = 0.001 C = 10	100%	96.5%	-	-

(Source: Personal collection)

Once all the models were analysed, the final chosen models were applied in a real time Open CV interface to detect how well the signs would be classified. For the final artifact, the libraries gTTS and playsound were also integrated to create and play an mp3 file that corresponds to the detected word. The tables 3, 4 and 5 show the analysis of each sign for the three selected models, whilst table 6 shows a summary of the number of signs split into the three categories described in the Primary Research, Methology and Ethics chapter: ‘Detected’, ‘Detected with reservations’ and ‘Not detected’.

It can be noticed that the model with 2 LSTM layers had lower number of different sign detections comparing to the SVM ‘rbf’ and the model with 2 CCN layers, dropout and 2 LSTM layers, it could be a consequence of the former being the model with lower accuracy of training and testing amongst the three, reflecting in the precision of the classifications in real time when comparing to the other two models with higher accuracy. Between the other two models, that had the same amount of signs precisely detected (8 signs), the model with 2 CCN layers, dropout and 2 LSTM layers had more signs detected with reservations, so it would tend to be the best model amongst the three final models, although none of them were able to detect all signs precisely.

There are other relevant points that these analyses raise. The first one is the fact that the signs ‘to want’, ‘jigsaw’ and ‘girl’ were not correctly classified by any of the models in the real time detection. ‘To want’ is a sign that is visually very similar to ‘to sit’ – both of them place the right fingers crossways on the left fingers in front of the chest, the difference is in the direction of the movement. Even the Neural Network models with LSTM were not able to differ the movements and tended to pick ‘to sit’. Similarly, the other words that could not be identified by any model, ‘jigsaw’ and ‘girl’ were considered similar to ‘to play’ and ‘hello’ respectively. Differently from ‘to want’ and ‘to sit’, ‘jigsaw’ and ‘to play’ are not similar beside the fact that they are generated with two hands in front of the chest. ‘Girl’ and ‘hello’ are also not similar if fully compared, although the sign of ‘girl’ starts with the right hand close to the forehead, like the sign ‘hello’. For these three cases – ‘to want’, ‘jigsaw’ and ‘girl’ – additional data would be beneficial to help the models to differentiate them of other signs that the models are struggling to distinguish them from.

Another point that can be noticed between the two models that performed better in the real time scenario is that the SVM struggled to identify several signs with two hands, classifying some of them as ‘to play’, whereas the Neural Network with CNN and LSTM struggled with one hand signs that are generated on the same area of the body with the right hand, ‘you’ and ‘good’, and these were easily detected by the SVM. Further studies would be required to determine whether this is a coincidence or there are intrinsic features of each model that helps them to manage certain skeletal limbs better than others.

Table 3: Two LSTM layers – Detection Status of each sign

Sign	Detection Status	Comments
To play	Detected with reservations	Model also picks ‘to sit’, which is not a similar sign, although both signs use the two hands.
To look	Detected with reservations	Model tends to pick ‘hello’, which is a sign in the same area of the body, using the right hand close to the right side of the face.
To sit	Detected	
To go	Detected with reservations	The model eventually struggles to distinguish ‘to go’ and ‘hello’ .
You	Detected with reservations	The model eventually struggles to distinguish ‘you’ and ‘please’ when the right hand is too close to the chest.
Good	Not detected	The model picks ‘you’ or ‘please’, which are other one-hand signs close to the chest
What	Detected with reservations	Rarely selected, model tends to pick ‘to play’ or ‘book’ depending on the angle of the camera
Time	Detected with reservations	Selected in a specific position when the landmarks of both hands are clearly captured by MediaPipe
Thank you	Detected with reservations	Sometimes the probability of ‘thank you’ increases, but the model tends to pick the signs ‘you’ and ‘please’, which are not very similar to ‘thank you’
Book	Detected with reservations	Easily detected when the webcam angle is greater than 45 degrees. When the laptop angle is 45 degrees or less, the model can identify book, although it also selects other two-hand signs like ‘jigsaw’ and ‘to play’
Hello	Detected	
To want	Not detected	The model picks either ‘to sit’, which is very similar, or ‘you, which is not similar
Me	Not detected	The model picks other one-hand signs at chest level like ‘you’ or ‘please’
Girl	Not detected	The model can only detect ‘hello’

Box	Not detected	Other two-hand signs picked, like ‘to play’ or ‘book’
To show	Not detected	The model tends to pick other two-hand signs, like ‘jigsaw’ or ‘book’
Table	Not detected	The model detects a very different sign: ‘you’
Jigsaw	Detected with reservations	Similarly to ‘thank you’, the probability of ‘jigsaw’ increases when the sign is carried out, but the detection never fulfills the threshold requirements, and the model may select other two-hand signs like ‘to play’ and ‘book’
Game	Detected with reservations	Game is a sign similar to ‘to play’ and, probably because of that, shows the same problem of ‘to play’, in which the model may also pick ‘to sit’. However, game is more complex than ‘to play’, with more gesture variation, so game never fulfills the threshold to be selected. Therefore, the model tends to pick either ‘to play’ or ‘to sit’
Please	Detected	

(Source: Personal collection)

Table 4: Two 1-D CNN + 0.2 Dropout + Two LSTM layers – Detection Status of each sign

Sign	Detection Status	Comments
To play	Detected	
To look	Detected with reservations	Detected easily when the angle of the webcam is greater than 45 degrees, if the angle is 45 degrees or less, the model picks the word ‘hello’
To sit	Detected	
To go	Detected	
You	Not detected	The model tends to pick either ‘please’ or ‘to want’, which are not similar models
Good	Not detected	The model tends to pick either ‘please’ or ‘to want’, which are not similar models
What	Not detected	The model chooses other two-hand signs, ‘to play’ or ‘to show’
Time	Detected with reservations	Detected in a specific position. Until this position is found, other two-hand signs are detected, like ‘to play’, ‘to show’ or ‘box’
Thank you	Detected	
Book	Detected	
Hello	Detected	
To want	Not detected	The model’s probabilities are distributed between ‘to sit’, which is a very similar sign, and ‘to play’ another two-hand sign
Me	Detected with reservations	Since the model tends to pick ‘me’ when there is no movement of the hands at chest level, this detection may be compromised
Girl	Not detected	The model tends to pick the sign ‘hello’

Box	Detected with reservations	Detected when the laptop angle is greater than 45 degrees
To show	Detected	
Table	Detected with reservations	Detected when the laptop angle is equal or less than 45 degrees
Jigsaw	Not Detected	Other two-hand signs chosen like ‘to show’ or ‘to play’
Game	Detected with reservations	Detected when the angle of the laptop is equal or less than 45 degrees
Please	Detected	

(Source: Personal collection)

Table 5: SVM RBF (GAMMA = 0.001) – Detection Status of each sign

Sign	Detection Status	Comments
To play	Detected	
To look	Detected with reservations	Detected easily when the signer is close to the left side of the frame, otherwise the model detects ‘hello’
To sit	Detected	
To go	Detected	
You	Detected	
Good	Detected with reservations	Detected easily when the signer is close to the right side of the frame, otherwise the model detects ‘hello’
What	Not detected	The model chooses other two-hand signs, ‘to play’ or ‘to show’
Time	Detected	
Thank you	Detected with reservations	Sometimes the model detects the word ‘you’
Book	Not detected	Other two-hand signs are detected instead, ‘time’ and ‘to play’
Hello	Detected	
To want	Not detected	The model tends to pick either ‘to sit’ or ‘to play’
Me	Detected	
Girl	Not detected	The model tends to pick the sign ‘hello’
Box	Not detected	The model picks ‘to play’ or ‘book’
To show	Not detected	The model picks ‘to play’
Table	Not detected	The model picks ‘to play’
Jigsaw	Not detected	The model picks ‘to play’
Game	Not detected	The model picks ‘to play’
Please	Detected	

(Source: Personal collection)

Table 6: Summary of detection statuses per machine learning model

Model	Detected	Detected with reservations	Not Detected
Double CNN + Dropout + Double LSTM	8	6	6
Double LSTM	3	10	7

SVM RBF Gamma = 0.001 C = 10	8	3	9

(Source: Personal collection)

In relation to the behaviour of the models when no action was carried out by the signer, the two layer LSTM picked the word ‘girl’ in the still position when only arms, chest and face were showed. If the hips were also showed in still position, the model picked ‘to want’. If the hips and the hands are showed with the elbows, the model picked the word ‘box’. If just the face and the chest landmarks are showed, the model does not pick a final word, but the probabilities for ‘to want’ and ‘to show’ tend to increase.

The Neural Network model with 2 CNN layers, dropout and two LSTM layers picked tended to pick the word ‘me’ and eventually ‘girl’ when the signer was in still position not showing hands or hips tends to pick the word ‘me’ and eventually ‘girl’. If the hips were also showed in still position, the model ‘table’ or ‘hello’. If the hips and the hands were showed with the elbows, it picked the word ‘jigsaw’ or ‘to sit’. If just the face and the chest landmarks were showed, the model picked the word ‘me’, while the probability of ‘girl’ could also increase.

Finally, the SVM model picked ‘to want’ when the signer was in a still position not showing hands or hips. If the hips were also showed in still position, the word ‘to want’ was still selected. If the hips and the hands were showed with the elbows, the model picked several words, like ‘to sit’, ‘table’, ‘game’ . If just the face and the chest landmarks are showed, the model picks either the sign ‘hello’ or the sign ‘to want’.

It can be concluded then that all three models had unstable classifications when no actual sign was generated by the signer, requiring workarounds not to generate

mistaken classifications. One alternative for that would be the creation of a separate status in the dataset called ‘no action’ for example. A new assessment would need to be made with all the studied models to confirm whether the addition of this new action would stabilize the classifications when no action is being performed or if this addition could affect the classification of the other signs. In relation to all the signs, none of the three models were able to classify precisely the three signs, being the Neural Network with 2 CNN layers, dropout and 2 LSTM layers the one that achieved better classifications – fourteen different signs if the signs detected and the signs detected with reservations are considered, being 8 signs precisely detected. As it could be seen above, some signs were not detected by any model – ‘jigsaw’, ‘to want’ and ‘girl’ – and would require a special attention during the management of the motion capture. Other signs that this final model could not classify, ‘you’ and ‘good’ were also not classified, although the SVM model detected these two signs in real time, which may indicate that the Neural Network may require additional data for these two cases to be able to properly distinguish them.

However, in general, the most problematic matter was the specific laptop angle required to detect some words and the distance that the signer needed to be from the laptop camera to generate accurate signs – in the real time detection, all final models had no accuracy to classify the signs when the signer was distant enough to show the hip landmark or even the leg landmarks. These elements enhance the need of additional data augmentation. Lu *et al.* (2016) and Meng *et al.* (2019) propose skeletal offset as a data augmentation strategy and, considering that the angle of the camera and the depth of the signer interfered in the final classification of some words and also that the MediaPipe numpy arrays are mainly composed of x, y and z coordinates, a new experiment could be

carried out applying a programmed function that should multiply the existing data but changing all the x coordinates and/or all the y coordinates and/or all z coordinates in small offsets between 0.05 and 0.2, for example. Since the MediaPipe data is alredy normalized, a special attention would need to be given to the logic of the skeletal landmarks that go beyond the value 1 or below 0, which are the limits of the MediaPipe normalization for the coordinates.

11. Conclusions and Future Research

This study was able to achieve the recognition of different Lámh language signs by applying machine learning models and deep neural network architectures on a database generated by capturing landmarks of a signer using the MediaPipe python library. Amongst all the applied models, although the model of two LSTM layers (with 32 and 64 nodes respectively and a 32 nodes dense layer) and the rbf SVM with width 0.001 and C=10 demonstrated a good behaviour between the increase of training and testing alongside high accuracy (90% and 100% of training accuracy respectively), they did not generate an actual real time detection tool that could accurately predict most of the signs in real time. The neural network model with 2 layers of convolutional neural network (with 32 and 64 nodes respectively), 2 layers of long short term memory (with 32 and 64 nodes respectively) and a 32 nodes dense layer equally had a good performance 99,86% of training accuracy and 93,33% of testing accuracy with lower signs of overfitting when comparing to previous models. Furthermore, the model showed a better performance than the other two when executed in an actual real time detection, being able to correctly classify 11 of the 20 words in different circumstances.

Thus, the secondary and primary objectives could be fulfilled since it is possible to generate a tool that will recognise Lámh language signs. As a consequence, the research question is also answered once this machine learning tool created an opportunity to be explored by communication partners. This study does not propose the ending of the human to human interactions, but the training and improvement of the communication partners – being professionals of the social sector or family and friends – so they can remember or expand their Lámh vocabulary in an attempt to promote a better

communication with the Lámh users. However, there are points that must be corrected and improved in the final model which lead to future work endeavours.

The final chosen model is not ideal as the words that are properly recognised (‘to play', 'to look', 'to sit', 'to go', ‘time’, ‘thank you’, ‘book’, 'hello', ‘me’, 'box', 'to show', 'table', ‘game’, 'please',) are better recognised with certain angles of the webcam and within certain distances from the webcam. When the user is too distant from the webcam and the landmarks of the hips or legs are included in the numerical array, the classifications of the real time detection get unstable and drastically inaccurate when compared to other angles. Moreover, the model should have satisfactorily predicted all the proposed 20 signs. These elements lead to three corrections in the methodology.

The first element is the size of the dataset. Similarly to the issue faced by Bagby *et al.* (2021) when creating a model to classify ASL signs, the greater the variety of signs, the greater the dataset must be so the machine learning model has more content to distinguish each sign. This would be a workaround for the signs that were too similar and therefore the model struggled to classify them, like the signs ‘to play’, ‘what’ and ‘game’, or ‘to want’ and ‘to sit’, as examples of similar signs with two hands that the final model did not perform well tending to choose the sign ‘to play’, and ‘you’, ‘me’ and ‘good’ as examples of signs with one hand that the model would tend to choose ‘me’. The increase of the dataset with more data should also contemplate more inputs with different poses of the body, including the hips or the legs present in more inputs since the final model performed badly when other parts of the body below the chest also appeared and the ideal model should aim for generalised scenarios.

The second element is a work that could benefit not only this computer vision project, but the computer vision area as a whole. Authors like Lu *et al.* (2016) and Meng *et al.* (2019) discuss offset and data augmentation alternatives of skeletal data aiming to improve the accuracy of their studied machine learning model based on well-known RBGD datasets like NTU RGB+D. However, the offset of skeletal data should also be explored in cases like the one of this research which generated its own dataset: although there were models that performed well when classifying the existing Lámh signs, the created dataset was not comprehensive enough for the final user experience. Thus, the augmentation of the MediaPipe Skeletal data through controlled offsets in the normalised coordinates would enhance the generalisation of the dataset and of the final machine learning model as a consequence. This methodology correction could be a potential correction to the inaccuracies of the chosen model when the webcam was set in an angle that was too different from the angle used to build the trained dataset. However, this would need to be experimented in further studies.

The third element that could be corrected in the proposed methodology of this study is to address the *softmax* probability distribution amongst the classes in the Neural Network models when there is no action being performed. When there was no action during the actual experimentation of the models, the Support Vector Machine model tended to not generate any classification because of the imposed threshold of 0.9 and requirement of 20 consecutive repeated classifications. However, the same could not be observed in the Neural Network models as they tended to pick one of the classifications even when the user was not performing any action. A suggested workaround that can be experimented is to add a condition in which the classification of the model will be replaced

by a new “no action” status if no landmarks of the hands can be perceived. Other solutions should be assed as the previous one requires that the user hides their hand to not stimulate a classification from the Neural Network models. A more comprehensive dataset could also correct the mentioned issue.

If the three methodology corrections above are executed and potentially generate a better real time Lámh language classifier, two further branches would be crucial for the consolidation of this tool in the Lámh sector: a first approach should aim for the accessibility of the machine learning tool, exploring the integration of the tool with web and software interfaces in order to turn them into actual alternatives for a variety of users; the second approach should assess the user experience impact, in which the machine learning tool, adapted to an accessible user interface, would be tested with professionals of the Lámh and speech therapy sectors so their opinions, suggestions an concerns can be raised and potentially generate corrections to the real time Lámh language detector.

In a social perspective, the initiative of this research by itself is a great contribution to the Lámh language sector in Ireland in terms of the promotion of technologies and alternatives that can assist the area. During the development of the project, the methodologies and objectives of the research were displayed and explained to the representatives of the language, special needs assistants and speech therapists in order to stimulate their participation. This intersection between the data analytics sector and the social care sector brought curiosity and widened the involved professionals’ awareness of how machine learning can contribute and develop unexpected solutions even to an AAC mechanism like Lámh, which is in essence an unassisted communication tool. Such integration and knowledge exchange between sectors was most certainly the best

contribution of this research above all technical achievements that were fulfilled through the project.

All in all, this research was able to conciliate the data analytics background with the social concern by integrating the most relevant academic literature related to Lámh language and AAC with the state of the art in regards to sign language detection and classification in the machine learning researches of other sign languages and motion recognition. This intends to challenge the current computer vision studies in a positive way, specially the areas towards Alternative and Augmented Communication. The problem definition must ultimately be user oriented and, although an outstanding progress has been perceived in the existing machine learning models and solutions within the last two decades, very little attention is given to the integration of these advanced technologies with a greater variety of languages as there are several existing KWS languages, like Lámh, which contain an expressive number of users that could benefit from the machine learning advances, but at the moment there are no studies of proposed tools to help the promotion of these languages and the user experience. The research carried out by Alexanderson and Beskow (2015) must be mentioned as they promoted a methodology for Swedish KWS motion capture with a final intention to create a game avatar. In relation to researches that are more focused in the machine learning aspect, improving and outperforming the existing state of the art will always be an important validation and scientific pursuit, but the final intention of this research is to open the discussion and highlight the importance of putting the practical social benefits as the major priority of the computer vision studies and experiments.

References

Alexanderson, S. and Beskow, J. (2015). Towards Fully Automated Motion Capture of Signs--Development and Evaluation of a Key Word Signing Avatar. *ACM Transactions on Accessible Computing (TACCESS)*, *7*(2), pp.1-17.

Alzubaidi, L., Zhang, J., Humaidi, A., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M., Al-Amidie, M. and Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1).

Bagby, B., Gray, D., Hughes, R., Langford, Z. and Stonner, R. (2021). Simplifying sign language detection for smart home devices using google mediapipe.

Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F. and Grundmann, M. (2020). Blazepose: On-device real-time body pose tracking. *arXiv preprint arXiv:2006.10204*.

Bradski, G. and Kaehler, A. (2008). *Learning OpenCV*. Beijing: O'Reilly.

Byrne, Á., Pyne, J. and Sheehan, V. (2019). Use of key word signing for children and adults with intellectual disability in an Irish context. *Tizard Learning Disability Review*, 24(3), pp.113–120.

Chan, Joseph O. (2013). An Architecture for Big Data Analytics. *Communications of the IIMA*, 13 (2).

Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Du, Y., Wang, W. and Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1110-1118).

Dolly, A. and Noble, E. (2018). ‘Lámh Signs Combined’ – Investigating a Whole School Approach to Augmentative and Alternative Communication (AAC) Intervention Through Research in Practice. *REACH: Journal of Inclusive Education in Ireland*, 31(1), pp. 53–68.

Ercolano, G. and Rossi, S. (2021). Combining CNN and LSTM for activity of daily living recognition with a 3D matrix skeleton representation. *Intelligent Service Robotics*, *14*(2), pp.175-185.

Etikan, I., Abubakar Musa, S. and Sunusi Alkassim, R. (2016). Comparison of Convenience Sampling and Purposive Sampling. *American Journal of Theoretical and Applied Statistics*, 5(1), p.1.

Everingham, M., Zisserman, A., Williams, C., Van Gool, L., Allan, M., Bishop, C., Chapelle, O., Dalal, N., Deselaers, T., Dorkó, G., Duffner, S., Eichhorn, J., Farquhar, J., Fritz, M., Garcia, C., Griffiths, T., Jurie, F., Keysers, D., Koskela, M., Laaksonen, J., Larlus, D., Leibe, B., Meng, H., Ney, H., Schiele, B., Schmid, C., Seemann, E., Shawe-Taylor, J., Storkey, A., Szedmak, S., Triggs, B., Ulusoy, I., Viitaniemi, V. and Zhang, J. (2006). The 2005 PASCAL Visual Object Classes Challenge. *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pp.117-176.

Frizelle, P. and Lyons, C. (2022). The development of a core key word signing vocabulary (Lámh) to facilitate communication with children with down syndrome in the first year of mainstream primary school in Ireland. *Augmentative and Alternative Communication*, pp.1-14.

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*.

Girshick, R. (2015). Fast R-CNN. *2015 IEEE International Conference on Computer Vision (ICCV)*.

Grandini, M., Bagli, E. and Visani, G. (2020). Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*.

Graves, A. (2011). Practical variational inference for neural networks. Advances in neural information processing systems, 24.

Halder, A. and Tayade, A. (2021). Real-time vernacular sign language recognition using mediapipe and machine learning. *Journal homepage: www. ijrpr. com ISSN*, *2582*, p.7421.

He, K., Gkioxari, G., Dollar, P. and Girshick, R. (2017). Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp.2980-2988.

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778.

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, 9(8), pp.1735-1780.

Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q. (2017). Densely connected convolutional networks*. In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700-4708.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *In International conference on machine learning*, pp. 448-456.

Jiang, D., Li, G., Sun, Y., Kong, J. and Tao, B. (2018). Gesture recognition based on skeletonization algorithm and CNN with ASL database. *Multimedia Tools and Applications*, 78(21), pp.29953-29970.

Jim, K.C., Giles, C.L. and Horne, B.G. (1996). An analysis of noise in recurrent neural networks: convergence and generalization. *IEEE Transactions on neural networks*, 7(6), pp.1424-1438.

Karim, M., Pujari, P. and Sewak, M. (2018). *Practical Convolutional Neural Networks*. Birmingham: Packt Publishing.

Kent-Walsh, J. and Mcnaughton, D. (2005). Communication Partner Instruction in AAC: Present Practices and Future Directions. *Augmentative and Alternative Communication*, 21(3), pp.195–204.

Koller, O., Zargaran, S., Ney, H. and Bowden, R. (2018). Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs. *International Journal of Computer Vision*, *126*(12), pp.1311-1325.

Kim, H., Jeon, J., Han, Y., Joo, Y., Lee, J., Lee, S. and Im, S. (2020). Convolutional Neural Network Classifies Pathological Voice Change in Laryngeal Cancer with High Accuracy. J*ournal of Clinical Medicine*, 9(11), p.3415.

Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kosmidou, V. and Hadjileontiadis, L. (2009). Sign Language Recognition Using Intrinsic-Mode Sample Entropy on sEMG and Accelerometer Data. *IEEE Transactions on Biomedical Engineering*, 56(12), pp.2879-2890.

Krizhevsky, A., Sutskever, I. and Hinton, G. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), pp.84-90.

Lefebvre, G., Berlemont, S., Mamalet, F. and Garcia, C. (2013). BLSTM-RNN based 3D gesture classification. In *International conference on artificial neural networks* (pp. 381-388). Springer, Berlin, Heidelberg.

Li, C., Wang, P., Wang, S., Hou, Y. and Li, W. (2017). Skeleton-based action recognition using LSTM and CNN. In 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) (pp. 585-590). IEEE.

Liao, Y., Xiong, P., Min, W., Min, W. and Lu, J. (2019). Dynamic Sign Language Recognition Based on Video Sequence With BLSTM-3D Residual Networks. *IEEE Access*, 7, pp.38044-38054.

Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C. (2014). Microsoft COCO: Common Objects in Context. *Computer Vision – ECCV 2014*, pp.740-755.

Liu, J., Wang, G., Duan, L.Y., Abdiyeva, K. and Kot, A.C. (2017). Skeleton-based human action recognition with global context-aware attention LSTM networks. *IEEE Transactions on Image Processing*, *27*(4), pp.1586-1599.

Liu, J., Shahroudy, A., Xu, D. and Wang, G. (2016). Spatio-temporal lstm with trust gates for 3d human action recognition. In *European conference on computer vision* (pp. 816-833). Springer, Cham.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. and Berg, A. (2016). SSD: Single Shot MultiBox Detector. *Computer Vision – ECCV 2016*, pp.21-37.

Lu, L. (2020). Dying ReLU and Initialization: Theory and Numerical Examples. *Communications in Computational Physics*, 28(5), pp.1671-1706.

Lu, G., Zhou, Y., Li, X. and Kudo, M. (2016). Efficient action recognition via local position offset of 3D skeletal body joints. *Multimedia Tools and Applications*, *75*(6), pp.3479-3494.

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.L., Yong, M.G., Lee, J. and Chang, W.T. (2019). Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*.

Martínez, A.M., Wilbur, R.B., Shay, R. and Kak, A.C. (2002). Purdue RVL-SLLL ASL database for automatic recognition of American Sign Language. In *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces* (pp. 167-172). IEEE.

Mehdi, S.A. and Khan, Y.N. (2002). Sign language recognition using sensor gloves. In *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02.* (Vol. 5, pp. 2204-2206). IEEE.

Meng, F., Liu, H., Liang, Y., Tu, J. and Liu, M. (2019). Sample fusion network: An end-to-end data augmentation network for skeleton-based human action recognition. *IEEE Transactions on Image Processing*, *28*(11), pp.5281-5295.

Nunnari, F., España-Bonet, C. and Avramidis, E. (2021). A data augmentation approach for sign-language-to-text translation in-the-wild. In *3rd Conference on Language,*

Data and Knowledge (LDK 2021). Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Prior, S. (2011). *Towards the full inclusion of people with severe speech and physical impairments in the design of Augmentative and Alternative Communication software* (Doctoral dissertation, University of Dundee).

Rabiner, L.R. and Juang, B.-H. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1), pp. 4–16.

Raheja, J.L., Mishra, A. and Chaudhary, A., 2016. Indian sign language recognition using SVM. *Pattern Recognition and Image Analysis*, *26*(2), pp.434-441.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ren, S., He, K., Girshick, R. and Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), pp.1137-1149.

Salaün, A., Petetin, Y. and Desbouvries, F. (2019). Comparing the modeling powers of RNN and HMM. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)* (pp. 1496-1499). IEEE.

Sanchez, S., Romero, H. and Morales, A. (2020). A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework. *IOP Conference Series: Materials Science and Engineering*, 844, p.012024.

Schlosser, R. and Wendt, O. (2008). Effects of Augmentative and Alternative Communication Intervention on Speech Production in Children With Autism: A

Systematic Review. *American Journal of Speech-Language Pathology,* 17(3), pp.212–230.

Sennott, S., Akagi, L., Lee, M. and Rhodes, A. (2021). AAC and Artificial Intelligence (AI). *Topics in language disorders*, 39(4), pp.389–403.

Sharma, N., Jain, V. and Mishra, A. (2018). An Analysis Of Convolutional Neural Networks For Image Classification. *Procedia Computer Science*, 132, pp.377-384.

Shrestha, A. and Mahmood, A. (2019). Review of Deep Learning Algorithms and Architectures. *IEEE Access*, 7, pp.53040-53065.

Shorten, C. and Khoshgoftaar, T. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1).

Sigafoos, J., O'Reilly, M., Lancioni, G. and Sutherland, D. (2014). Augmentative and Alternative Communication for Individuals with Autism Spectrum Disorder and Intellectual Disability. *Current Developmental Disorders Reports*, 1(2), pp.51–57.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Srivastava, S., Divekar, A., Anilkumar, C., Naik, I., Kulkarni, V. and Pattabiraman, V. (2021). Comparative analysis of deep learning image detection algorithms. *Journal of Big Data*, 8(1).

Stolcke, A. (2002). SRILM - An extensible language modeling toolkit. *In Proceedings on international conference on spoken language processing (ICSLP), Denver, Colorado*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). Going deeper with convolutions. *In*

Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9.

Tan, X., Trembath, D., Bloomberg, K., Iacono, T. and Caithness, T. (2014). Acquisition and generalization of key word signing by three children with autism. *Developmental Neurorehabilitation*, 17(2), pp.125-136.

Vo, D., Huynh, H., Doan, P. and Meunier, J. (2017). Dynamic Gesture Classification for Vietnamese Sign Language Recognition. *International Journal of Advanced Computer Science and Applications*, 8(3).

Wang, T., Chen, Y., Zhang, M., Chen, J. and Snoussi, H. (2017). Internal transfer learning for improving performance in human action recognition for small datasets. *IEEE Access*, 5, pp.17627-17633.

Wilkinson, K. and Hennig, S. (2007). The state of research and practice in augmentative and alternative communication for children with developmental/intellectual disabilities. *Mental Retardation and Developmental Disabilities Research Reviews*, 13(1), pp.58-69.

Xue-Wen Chen and Xiaotong Lin. (2014). Big Data Deep Learning: Challenges and Perspectives. *IEEE Access*, 2, pp.514-525.

Zhang, E., Xue, B., Cao, F., Duan, J., Lin, G. and Lei, Y. (2019). Fusion of 2D CNN and 3D DenseNet for Dynamic Gesture Recognition. *Electronics*, 8(12), p.1511.

Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.L. and Grundmann, M. (2020). Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*.

Zhang, W. and Wang, J. (2019). Dynamic Hand Gesture Recognition Based on 3D Convolutional Neural Network Models. *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*.

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE multimedia*, *19*(2), pp.4-10.

Appendix A

Top twenty words of the core school-based vocabulary built by Frizelle and Lyons (2022):

Table 7**:** Vocabulary based on study carried out by Frizelle and Lyons (2022) in alphabetical order from the left to the right

Vocabulary
PLAY, TO
LOOK, TO
SIT, TO
GO, TO
YOU
GOOD
WHAT?
TIME
THANK YOU
BOOK
HELLO/HOW ARE YOU?
WANT, TO
I/ME
GIRL
BOX
SHOW, TO
TABLE
JIGSAW
GAME
PLEASE

(Modified from source: Frizelle and Lyons 2022, p. 8)