



**INFORMATICS
INSTITUTE OF
TECHNOLOGY**

INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER

**Signergy: Sign Language Recognition with Positional
Coordinates for Aiding Video Conferencing**

A Dissertation by

Mr. Michael Nadesapillai

Supervised by

Mr. Saman Hettiarachchi

Submitted in partial fulfilment of the requirements for the BSc in Computer Science
degree at the University of Westminster.

April 2023

ABSTRACT

Whilst Video Conferencing platforms were predominantly focused on social and recreational uses in the past, the pandemic and post-pandemic shifts of life have changed this greatly. With Work-from-Home becoming more abundant in the corporate sphere across various industries, video conferencing for work purposes is a heavily utilised facet. Whilst certain solutions such as Subtitles and Closed-Captions are starting to become more prominent across daily life in a bid to ease this disparity, there are still many areas in which the AV-Impaired community struggles in, compared to those who are non-impaired.

To solve this issue, with a solution that is feasible to run on the average desktop, MediaPipe was used for feature extraction of the landmarks of the hand, whilst numpy was used to convert the array of 3D positional coordinates into a vector representing the angles between all possible pairs of landmarks, and a 2D-CNN model with 2D Max Pooling Flattening, and Padding, were used as the model tasked to infer the signs.

The above approach yielded a 70% accuracy across the classification report and confusion matrix, whilst the macro-average and micro-average ROC curves resulted 0.84 and 0.85 respectively. As such, we can conclude that the model selection for the use case has indeed been successful, and that the model will suit the needs of Signergy well.

Computing Methodologies → Artificial Intelligence → Computer Vision → Computer Vision Problems → Object Detection

Computing Methodologies → Artificial Intelligence → Computer Vision → Computer Vision Problems → Object Recognition