POLITECNICO DI TORINO

Master's Degree in Data Science and Engineering

Master's Degree Thesis

DEVELOPING A LIGHTWEIGHT REAL-WORLD SIGN LANGUAGE RECOGNITION MODEL USING HAND ENERGY IMAGE

Supervisors

Prof. Sarah AZIMI

Prof. Corrado DE SIO

Candidate

Ricardo NICIDA KAZAMA

JUNE 2025

Summary

This thesis presents a lightweight approach for isolated Italian Sign Language (LIS) recognition, leveraging Hand Energy Images (HEIs) and custom convolutional neural networks (CNNs) designed for efficiency and real-world adaptability. Several preprocessing strategies, including Gaussian blur, background lightening, and adaptive thresholding, were explored to enhance model robustness across controlled and unconstrained environments. Experiments conducted on a subset of the Italian Sign Language A3LIS-147 dataset and a real-world dataset recorded by the author revealed distinct trends: models trained without preprocessing achieved the highest performance on the controlled A3LIS test set, while models trained with adaptive thresholding and Gaussian blur achieved superior generalization to real-world data. The best real-world model attained a Top-1 accuracy of 48.28% and a Top-3 accuracy of 75.86%, highlighting the effectiveness of preprocessing in improving transferability. Despite limitations related to dataset size, signer diversity, and isolated sign scenarios, the proposed system demonstrated strong potential for future applications in real-time, mobile, or embedded environments.

Keywords: Sign Language Recognition, Italian Sign Language, Hand Energy Image, Convolutional Neural Networks, Gaussian Blur, Adaptive Thresholding.

Acknowledgements

First and foremost, I would like to thank God, who has blessed me throughout my entire life and especially during the more challenging recent periods. His guidance, strength, and grace sustained me through the difficult moments and made it possible for me to overcome the many obstacles along this journey.

I would like to express my deepest gratitude to my parents, Tony and Plácida, and to my brother Rafael, for their unconditional support and encouragement throughout my entire academic development. Their belief in me has been a constant source of strength and motivation. I also extend my heartfelt thanks to all my family members and friends, whose support has been fundamental in helping me reach this important milestone.

A special thanks to my girlfriend Alissa, who supported me through the many challenges of balancing student and working life. Her encouragement, patience, and positivity were crucial during the most demanding moments.

I am also sincerely grateful to all the professors from the University of São Paulo and Politecnico di Torino, whose guidance and teachings contributed significantly to my personal and academic growth. It has been an enriching experience to learn from such distinguished educators in both institutions.

Finally, I would like to thank Sarah Azimi and Corrado De Sio for their invaluable guidance and mentorship throughout the development of this thesis. Their support and understanding were essential during my experience as an international student in Italy, and I am deeply thankful for their role in this achievement.

To all of you, my sincere appreciation.

"Have I not commanded you? Be strong and courageous. Do not be afraid; do not be discouraged, for the Lord your God will be with you wherever you go."
Joshua 1:9

Table of Contents

List of Tables

List of Figures

Acronyms

AHE

Adaptive Histogram Equalization

ANN

Artificial Neural Network

ArSL

Arabic Sign Language

ASL

American Sign Language

AT

Adaptive Thresholding

ATP

Average True Probability

BSL

British Sign Language

CLAHE

Contrast Limited Adaptive Histogram Equalization

CNN

Convolutional Neural Network

DGS

Deutsche Gebärdensprache (German Sign Language)

FC-RNN

Fully Connected Recurrent Neural Network

GB

Gaussian Blur

HE

Histogram Equalization

HEI

Hand Energy Image

HMM

Hidden Markov Model

ISL

Indian Sign Language

KNN

K-Nearest Neighbors

LB

Light Background

LIBRAS

Língua Brasileira de Sinais (Brazilian Sign Language)

LIS

Lingua dei Segni (Italian Sign Language)

LRCN

Long-term Recurrent Convolutional Neural Network

LSTM

Long Short-Term Memory Network

MSE

Mean Squared Error

ND

No Dropout

ReLU

Rectified Linear Unit

RGB

Red, Green, Blue

RNN

Recurrent Neural Network

SGD

Stochastic Gradient Descent

SVM

Support Vector Machine

Chapter 1

Introduction

Hearing enables humans to perceive sounds in our environment, facilitating interaction, communication, expression of thoughts, and learning. Globally, over 1.5 billion people will experience some degree of hearing loss during their lifetime, of whom at least 430 million will need care [1]. Language development, psychosocial well-being, quality of life, educational attainment, and economic independence can be affected when hearing loss is not identified and properly addressed.

Various causes of hearing loss, including ear diseases, ear infections, and exposure to noise and chemicals, can jeopardize individuals' hearing across all age groups. The World Health Organization (WHO) estimates that over 1 billion young people risk permanent hearing loss, often unknowingly, by listening to music at high volumes for extended periods [1]. Addressing hearing loss through public health measures is essential to mitigate these risks.

Many individuals with ear conditions or hearing loss can benefit from effective interventions. In recent decades, advancements in hearing technology and medical treatments have significantly aided the hearing loss community. Medical and surgical treatments, hearing aids, cochlear implants, and rehabilitative therapy can help these individuals access education and communication, enabling them to reach their potential. However, most people can only afford to learn sign language to overcome their challenging barriers and have their potential constrained by the fact that the primary mode of communication relies on speaking and hearing.

Nevertheless, for a child who cannot receive a treatment such as a cochlear implant early in life, it is essential to teach sign language from birth. A study on language deprivation in deaf children [2] indicates that cochlear implants are not a reliable standalone intervention for a child's first language development. The priority should be fostering healthy growth across all developmental domains through a fully accessible first language foundation, such as sign language. Language deprivation, while waiting for a cochlear implant, can cause cognitive delays and mental health difficulties across the lifespan and can become a mental health

disorder, "language deprivation syndrome" in severe cases [2]. A study compared deaf babies exposed to sign language from birth to hearing ones and concluded that infants may utilize visual and auditory linguistic inputs equally [3]. The phenomenon studied in the paper was babbling, which, for many, was considered intrinsic to spoken language development. However, it was also observed in its manual form with deaf babies performing hand signs. Therefore, the authors suggest that the babies' predisposition to discover the particular patterns in the input signal (babbling) is a property of an amodal language capacity.

> *"Suppose that we had no voice or tongue, and wanted to communicate with one another, should we not, like the deaf and dumb, make signs with the hands and head and the rest of the body?"* - Socrates (469 - 399 BC)

[4] The sign language history dates from the 5th century BC represented by a Greek vase showing Philomela, whose tongue was cut by King Tereus of Thrace, using signs. Furthermore, Socrates' aforementioned statement in Plato's Cratylus was the first written evidence of sign language usage in human history. From the 5th AD, monastic communities, particularly those under vows of silence, used hand signs and finger alphabets for communication. The earliest known description of a hand alphabet is attributed to Saint Bede (5th-6th centuries), with further developments by Saint Bonaventure in the 13th century. These practices, especially prevalent in Benedictine communities, were probably the basis for the later application of manual language to the deaf in the 16th century. In 1771, Charles Michel de L'Épée established the first free school for the deaf and published syntax for sign language in 1774. In 1880, at the International Congress of Teachers of the Deaf held in Milan, the predominance of oralists resulted in anti-sign language resolutions being passed, banning sign language from schools. For most of the 20th century, speech, considered superior to sign language, was the only communication method for educating the deaf and dumb [4].

Later in the 20th century, sign language began to be recognized as the ideal method of deaf education. Initially, sign language was viewed as merely an imitation of spoken language. However, it gradually became recognized as a distinct form of communication that uses gestures to convey meaning independently [5]. According to National Deaf Children's Society [6], "Sign language is a visual language that uses hand shapes, facial expression, gestures and body language. Sign languages have their own vocabulary, construction, and grammar. Like spoken languages, sign languages are natural, which means that they're developed over the years by the people who use them". Currently, there are 138 to 300 distinct sign languages with no universal sign language shared across the globe [7]. Even countries with the same spoken language have different sign languages, such as Australia, England, and the United States of America [8]. Figures 1.1 and 1.2 show that the fingerspelling alphabet from ASL (American Sign Language) and BSL (British Sign Language)

are completely different, starting from the fact that ASL uses one hand while BSL uses both.

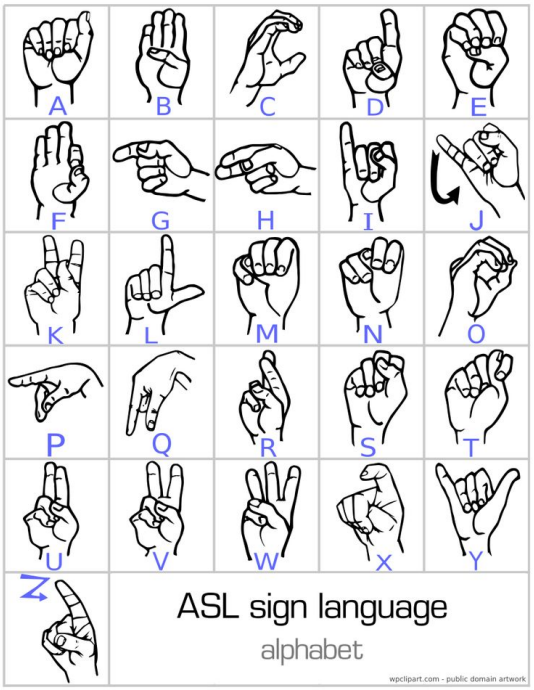

Figure 1.1: American Sign Language fingerspelling alphabet [9]

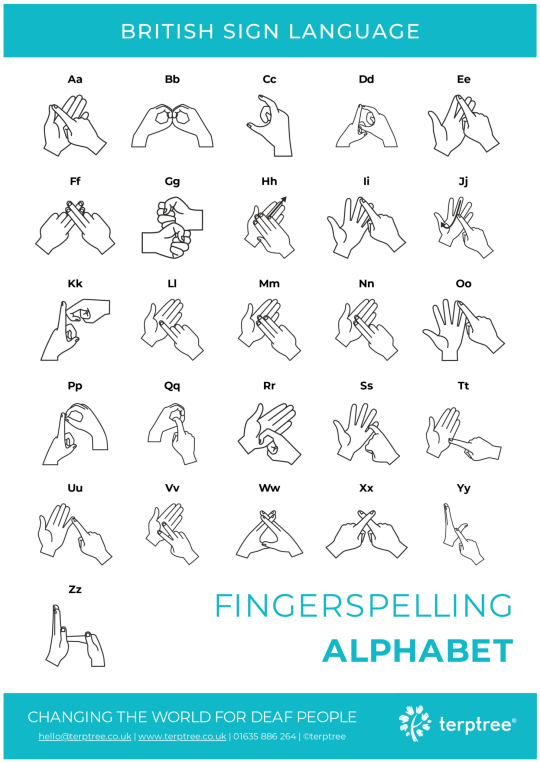

Figure 1.2: British Sign Language fingerspelling alphabet [10]

Another compelling aspect of sign language is its unique grammar, syntax, and structural characteristics. Unlike spoken languages, sign language operates with its distinct syntactical rules, where a single sign can convey an individual letter, a word, or even an entire sentence [5]. For a layperson, sign language is based only on the hands' movements, shapes, and positions, but it has many other features that most can not perceive at first glance.

[11, 12] Sign language can be divided into manual and non-manual signs. Manual signs consist of the basic movements, shapes, and positions of the hands and arms that carry explicit lexical meanings, such as letters, words, or phrases. Complementarily, non-manual signs, produced through facial expressions, tongue movements, cheek gestures, and body posture, convey morphemic information about lexical items or mark syntactic boundaries, such as the ends of phrases. In American Sign Language (ASL), an example of combining both features is the expression for "driving carelessly", which involves performing the manual sign for "driving"

while simultaneously placing the tongue between the front teeth to indicate the "carelessly" nuance [11, 12].

Sign language has many rules and nuances, just like any spoken language. However, it has only been in the spotlight since the late 20th century, and there are still many studies to be conducted so that it can continue to evolve and become more accessible to a wider audience. Even though relatively few people use sign language, they are an integral part of society and deserve to be included and empowered to reach their full potential as human beings.

1.1 Motivation and Goals

Higher education is key to the career and life development of any individual. However, for disabled people, it can become an inaccessible and challenging step in their educational journey. Disabled students have reported a lack of information about pursuing higher education, staff underdevelopment, and inadequate infrastructure to ensure equal access and support for academic formation [13]. Universities and institutions must focus on minimizing the stress caused by their infrastructure and systems, especially since the pressure of the higher education environment already increases mental health needs even among non-disabled students [14].

For deaf students, additional challenges include vulnerability to marginalization, inaccessibility of entire lectures even when interpreters are provided, and reliance on note-taking services [15]. Scottish students identified group tutorials and seminars as particularly challenging, acknowledging that creating effective access strategies for these situations remains difficult [16]. Deaf students experience exclusion not only in educational and organizational activities but also during participation in extracurricular events [15]. These difficulties demand so much energy that many deaf students report withdrawing from social activities, feeling less integrated into the “university family” compared to their hearing peers [17, 18].

[15] Hendry et al. report that the English language constitutes a major communication barrier for deaf Scottish higher education students, many of whom use British Sign Language (BSL) as their first or preferred language. A lack of educational materials available in BSL further exacerbates accessibility issues. Even for students proficient in reading English, obtaining further information about courses proves difficult due to barriers such as the inability to make phone inquiries. While a few students have access to sign language interpreters, in most cases such support is insufficient or unavailable, creating yet another communication barrier that limits active participation in academic and social spheres [15].

The aforementioned studies show that even with access to interpreters and note-taking services, deaf students continue to experience marginalization and exclusion due to limited accessibility during lectures, seminars, and extracurricular

activities. These challenges highlight the urgent need for innovative solutions that promote inclusion and accessibility. A sign language recognition model represents a promising approach to mitigating these barriers. Such a model could help bridge communication gaps by offering real-time translation of signs into written or spoken language, ensuring that deaf students can benefit from direct communication with their hearing peers. This technology would also empower students to engage more fully in academic and social environments, reducing the sense of isolation and allowing them to participate on more equitable terms.

Several engineering approaches have been proposed in the literature to address sign language recognition, including deep convolutional neural networks (CNNs), hand skeleton-based feature extraction, optical flow-based motion representation, and attention mechanisms. However, many of these approaches require large annotated datasets, highly complex architectures, or significant computational resources, limiting their practical applicability to real-world, low-resource scenarios.

In this context, this master's thesis proposes a lightweight and accessible solution by leveraging Hand Energy Images (HEIs) to represent the dynamic movement of hands during signs, combined with compact, custom-designed convolutional neural networks trained individually for each hand. Furthermore, multiple preprocessing strategies, such as background color lightening, Gaussian blur, and adaptive thresholding, were systematically evaluated to enhance model robustness across different environments. By training specialized models for left and right hands and later combining their predictions through a probabilistic strategy, this work aims to offer a simple yet effective architecture suitable for real-world deployment. The ultimate goal is to contribute a feasible tool to help deaf students overcome communication barriers within educational settings, promoting greater inclusion and participation.

Chapter 2

Literature Review

This chapter provides an overview of key studies and theoretical frameworks related to sign language recognition, highlighting the current state of knowledge, identifying gaps, and positioning this thesis within the broader academic discourse. The scope of this review lies in works that propose sign language or gesture recognition systems or contribute to the technical understanding of sign languages. This chapter will be divided into key topics: pioneering work, input modality, datasets, image preprocessing, and state-of-the-art work.

2.1 Pioneering Work

In 1981, Tartter and Knowlton [19] explored the possibility of using a simplified visual representation to enable sign language communication over a telephone line. They designed gloves with thirteen pieces of retro-reflective tape strategically placed on the hands and wrists to capture essential movements and hand shapes during signing. Subjects engaged in conversation while viewing only the reflected light spots on a monitor. The findings demonstrated that communication was successful at near-normal conversational speeds, although some difficulties were observed in fingerspelling. The study concluded that this reduced information system could be transmitted over a single telephone line, suggesting the potential for real-time, remote sign-language communication.

Throughout the 1990s, the majority of research efforts focused on employing Hidden Markov Models (HMMs) for sign language recognition, often utilizing DataGlove devices or colored gloves to improve hand tracking accuracy [20, 21, 22, 23, 24, 25, 26, 27]. HMMs operate under the assumption that a sequence of observations follows the Markov property, where the current state depends only on a limited history of prior states. Training HMMs involves addressing three core problems: evaluation, estimation, and decoding.

In 1995, Starner et al. [20] applied the forward-backward algorithm to evaluate the probability of observation sequences, the Baum-Welch algorithm to iteratively refine model parameters, and the Viterbi algorithm to find the most probable sequence of hidden states. Using a 40-word lexicon and a set of 99 test sentences, their system achieved a word recognition accuracy of 91.3% without relying on grammatical rules. Later, in 1998, Starner et al. [25] proposed two additional recognition systems that did not require colored gloves. One of these systems, utilizing a camera mounted on a cap worn by the signer, achieved 97% word recognition accuracy with unrestricted grammar on a 100-sentence test set, still within a 40-word lexicon.

In 2000, Bauer et al. [27] enhanced the traditional HMM-based approach by incorporating a language model during the decoding phase. Without the use of a language model, their system achieved accuracies of 94.0% and 91.8% for lexicons of 52 and 97 signs, respectively. When a Bigram language model was incorporated, the accuracies improved to 95.4% and 93.2%, respectively.

Overall, the early work on sign language recognition predominantly relied on glove-based systems to either provide direct signal measurements or improve hand-tracking accuracy before feeding the data into HMM frameworks. Additionally, training and testing datasets were typically collected under standardized and highly controlled conditions, often involving only one or a few signers. This setup, while effective in early experiments, posed limitations on model generalization and increased the risk of overfitting to specific signers or environments.

2.2 Input Modality

Generally, data input methods for sign language recognition are categorized into two main groups: glove-based and vision-based approaches. Zheng et al. [28] refer to glove-based methods as touch-based approaches, where data is captured through optical, magnetic, or acoustic sensors attached to the hands or body. In contrast, vision-based or untouched-based methods rely on video and depth data acquired from standard cameras or more advanced devices, such as the Microsoft Kinect, which additionally provides depth maps. Zheng et al. emphasize that glove-based methods tend to burden users with wearable devices and cables, whereas vision-based systems allow users to communicate naturally in front of a camera without physical constraints.

Expanding this categorization, Rastgoo et al. [29] further analyzed input modalities in vision-based systems, highlighting the use of infrared sensors and proposing an additional classification of inputs based on their temporal characteristics. According to their survey, input data can be divided into static and dynamic forms. On the one hand, many deep-learning models process static inputs, treating each

image or frame independently, focusing solely on spatial information. On the other hand, dynamic inputs incorporate temporal information, considering sequences of frames where spatial features evolve over time, which can significantly enhance recognition accuracy. Dynamic approaches also address challenges such as sentence tokenization into words, start and end detection of signs, and managing abbreviations or synonyms.

In summary, input modalities can vary depending on available resources, technological constraints, and the final objectives of the sign language recognition system. For daily applications, vision-based approaches offer a more practical and user-friendly solution, particularly considering the widespread availability of cameras in mobile devices.

2.3 Datasets

The main benchmark datasets for sign language recognition include:

- **American Sign Language (ASL)**:
  - Purdue RVL-SLLL Database [30]
  - American Sign Language Lexicon Video Dataset (ASLLVD) [31]
  - RWTH-BOSTON-104 [32]
  - RWTH-BOSTON-400 [33]
  - Massey Dataset [34]
- **Arabic Sign Language (ArSL)**:
  - SignsWorld Atlas Database [35]
  - Arabic Sign Language Database [36]
- **Brazilian Sign Language (LIBRAS)**:
  - LIBRAS-HC-RGBDS [37]
- **British Sign Language (BSL)**:
  - British Sign Language Corpus [38]
- **German Sign Language (DGS)**:
  - SIGNUM Database [39]
  - RWTH-PHOENIX-Weather 2014 [40]

The American Sign Language (ASL) datasets are among the most widely used in academic research and were among the first available benchmark datasets. Consequently, ASL is one of the most influential sign languages in computer vision and machine learning studies. In this work, the Italian Sign Language A3LIS-147 dataset [41], an uncommon but easily accessible database, was utilized.

2.3.1 American Sign Language (ASL)

In 2002, Martínez et al. [30] introduced the Purdue RVL-SLLL Database, consisting of 2,576 videos: 184 videos for each of 14 native ASL signers. The dataset was recorded under two lighting conditions: diffuse illumination to suppress shadows and directed illumination to enhance contrast. All videos are RGB AVI files with 640 × 480 resolution and 24-bit color depth.

In 2008, Athitsos et al. [31] presented the American Sign Language Lexicon Video Dataset (ASLLVD), created as part of a project to enable visual sign lookup systems. Initially covering almost 3,000 signs, each sign was performed by up to four native ASL signers and recorded from four different camera angles. Video resolution varies: most recordings are at 640 × 480 pixels and 60 fps, while some frontal views reach 1600 × 1200 pixels at 30 fps. The dataset now includes almost 9,800 videos across more than 3,300 signs.

In 2007, Dreuw et al. [32] compiled the RWTH-Boston-104 corpus from a larger Boston University dataset. It contains 201 sign sequences performed by three signers within a 104-sign vocabulary. The grayscale videos were recorded at 312 × 242 resolution at 30 fps.

In 2008, Dreuw et al. [33] introduced RWTH-Boston-400, expanding their prior work to 843 sentences performed by four signers using a 406-sign vocabulary. Videos are available in two versions: uncompressed at 648 × 648 resolution and compressed at 324 × 242 resolution.

In 2011, Barczak et al. [34] released MU_HandImages_ASL, focusing on static ASL fingerspelling (letters and numbers). Initially, it included 2,425 hand gesture images captured under different lighting conditions. The goal was to expand the dataset to 18,000 images across 20 individuals.

2.3.2 Arabic Sign Language (ArSL)

In 2015, Shohieb et al. [35] proposed the SignsWorld Atlas Database, covering both manual and non-manual signs (e.g., hand shapes, lip movement, facial expressions). Although the exact size was not specified, the dataset includes approximately 257 labels performed by 10 signers in front of a black background with 1024 × 768 image resolution.

In 2019, Ghazanfar et al. [36] released the ArSL2018 dataset, consisting of 54,049 grayscale images (64 × 64 resolution) covering the 32 Arabic alphabet letters.

2.3.3 Brazilian Sign Language (LIBRAS)

In 2013, Porfirio et al. [37] introduced LIBRAS-HC-RGBDS, a dataset with 610 videos from 5 signers covering 61 hand configurations. Each recording captured both RGB and depth information using a Kinect® sensor at 640 × 480 resolution, with videos lasting between 5 to 10 seconds.

2.3.4 British Sign Language (BSL)

In 2013, Schembri et al. [38] presented the BSL Corpus Project, gathering data from 249 participants who learned BSL before the age of seven. Participants engaged in 30-minute conversations recorded with three cameras. From this data, 7,332 lexical annotations and 25,000 conversational signs were created, yielding a lexical database of approximately 1,800 unique signs.

2.3.5 German Sign Language (DGS)

In 2007, Agris et al. [39] introduced the SIGNUM database, designed to address signer-independence challenges. It includes 15,600 sentence videos performed by 20 signers across 450 signs, with 780 × 580 pixel RGB videos recorded at 30 fps.

In 2014, Forster et al. [40] released the RWTH-PHOENIX-Weather 2014 corpus, based on German weather forecasts. The dataset contains 6,861 sentences across 1,558 signs, performed by 9 signers and recorded at 210 × 260 resolution at 25 fps.

2.3.6 Italian Sign Language (LIS)

In 2012, Fagiani et al. [41] presented A3LIS-147, one of the first Italian Sign Language datasets. It contains 147 signs performed by 10 different signers, totaling 1,470 videos. Signs are grouped into six categories (e.g., education, transportation, healthcare) and were recorded against a green background at 720 × 576 resolution and 25 fps.

2.4 Image Preprocessing

Preprocessing techniques are applied to images to enhance their quality, improving the input data of a model. Choosing a good selection of preprocessing techniques can greatly affect the accuracy of a given image-based model. There are two main classes of preprocessing techniques: image enhancement and image restoration.

Image enhancement techniques include Histogram Equalization (HE), Adaptive Histogram Equalization (AHE), Contrast Limited Adaptive Histogram Equalization (CLAHE), and logarithmic transformation. Image restoration includes a media filter, mean filter, Gaussian filter, adaptive filter, and Wiener filter. Furthermore, image preprocessing may involve operations such as image subtraction or averaging, generating additional features from the input.

2.4.1 Image Enhancement

Histogram Equalization

Histogram equalization is a preprocessing technique that strengthens colors and increases the contrast of an image (Gonzalez and Woods [42]). Verma and Dutta [43] discuss contrast enhancement techniques, pointing to histogram equalization as one of the best methods considering its easy implementation and great results. They also reviewed other extensions of the histogram equalization method, such as Adaptive Histogram Equalization and Contrast Limited Adaptive Histogram Equalization.

The Adaptive Histogram Equalization Technique differs from classical Histogram Equalization since the adaptive method computes several histograms for distinct sections of the images. According to Kumar et al. [44], this technique loses information, amplifies the noise, fails to retain the brightness, has a low signal-to-noise ratio, is very complex to implement, and takes a long computational time. They discuss how the Background Brightness Preserving Histogram Equalization performs better in all the aforementioned metrics than the classical, adaptive, brightness preserving bi, and recursive mean separate histogram equalization techniques.

These techniques are not suitable for RGB images since their methods generate histograms for grayscale images. Therefore, grayscale conversion is often applied to RGB images as a preprocessing technique that allows the application of grayscale-focused techniques.

Grayscale Conversion

According to Adeyanju et al. [45], grayscale conversion is one of the simplest image processing enhancement techniques in which the color space is converted to grayscale. Using grayscale images over RGB-colored ones simplifies the algorithms and reduces the computational requirements. Although the preprocessed image loses color information, it preserves the salient features of the colored image. The grayscale conversion equation (Biswas et al. [46]) is given in Eq. 2.1

$$GY = 0.56G + 0.33R + 0.11B \tag{2.1}$$

2.4.2 Image Restoration

According to Reeves [47], an image is a 2D representation of a 3D scene, and image restoration involves processing it to generate a more accurate representation of reality by reducing blur and noise. Various techniques, such as mean, median, Gaussian, and Wiener filters, serve different restoration purposes. The mean filter replaces the center pixel with the average value of its neighboring pixels, resulting in a smoother image and helping to reduce salt-and-pepper noise.

The Gaussian filter is a linear, non-uniform low-pass filter that blurs an image using a Gaussian function. It is widely used in image preprocessing to reduce noise and smooth edges. Common in sign language recognition research, it serves as a smoothing operator to enhance image clarity [45]. Oliveira et al. [48] applied the Gaussian filter for hand shape classification, achieving a precision of 99.65%.

The Wiener filter is used for noise removal and minimizes the mean square error (MSE) between the estimated and desired processes. It balances image smoothing and noise reduction, but may cause blurring due to its fixed filter. Its restoration function considers both the degradation function and noise characteristics, using a high-pass filter for deconvolution and a low-pass filter for noise reduction. Kaluri and Reddy [49] used this method to eliminate image noise beforehand, applying the adaptive histogram technique to enhance the hand gesture images.

2.4.3 Image Segmentation

According to Egmont-Petersen et al. [50], segmentation involves dividing an image into distinct regions based on specific criteria to ensure coherence within each part. There are two basic approaches used for image segmentation: contextual and non-contextual. The former leverages the relationships between image features, such as edges, intensity similarities, and spatial proximity, to improve segmentation accuracy. In contrast, non-contextual segmentation disregards spatial relationships and instead groups pixels based only on global attribute values. Furthermore, image segmentation techniques can be categorized into edge detection-based, thresholding-based, region-based, clustering-based, and artificial neural network-based [45].

Thresholding

Thresholding is the simplest and most commonly used technique for background removal. Global thresholding sets a single threshold value to divide the image into foreground and background. Meanwhile, adaptive thresholding partitions an image into sub-images, determining each threshold based on statistical measures, such as the mean plus the standard deviation of the pixel values within each region. Rao and Kishore [51] applied adaptive thresholding to remove the background of the videos for a continuous sign language recognition system. Dudhal et al. [52]

utilized adaptive thresholding to binarize the grayscale image into a black and white hand contour image for an isolated sign language recognition. Multilevel thresholding is applied to extract homogeneous regions in an image by defining multiple thresholds. The method performs well on images with complex or colored backgrounds where bi-level thresholding fails. Skin color segmentation, commonly used in applications like human-computer interaction, image recognition, traffic control, video surveillance, and hand segmentation, utilizes a color model to isolate skin regions in images. According to Shaik et al. [53], the RGB color space is the least favored for color-based detection and analysis, as the variation in human skin tones makes it hard to identify and distinguish skin regions.

Edge Detection

Edge detection is a fundamental technique in image processing that identifies areas in an image where intensity values change rapidly. This is typically done by locating points where the first derivative of intensity exceeds a certain threshold or where the second derivative shows zero crossings. Effective edge-based segmentation involves three key steps: identifying edges, removing unnecessary ones, and connecting the relevant edges. The most commonly used edge detection methods include Robert, Sobel, Prewitt, Laplacian of Gaussian, and Canny edge detectors. Thepade et al. [54] propose a method for recognizing sign language static gestures from images by leveraging edge detection techniques. All the aforementioned edge detection techniques are tested for extracting gradient information from the input images, generating edge maps highlighting the contours of hand shapes. The authors then compute color mean features fed into a Support Vector Machine classifier, achieving promising accuracy for static gesture recognition. Among the evaluated methods, the Sobel edge detector demonstrated superior performance in accurately capturing the relevant contour features for static sign language recognition.

2.5 State-of-the-art

Recent advancements in sign language recognition have been primarily driven by deep learning, particularly convolutional neural networks (CNNs), spatiotemporal models, and increasingly, contrastive learning techniques. The literature reveals a broad spectrum of approaches from static image classification using tailored CNNs to dynamic video modeling through 3D CNNs and recurrent neural networks (RNNs). Multimodal fusion strategies also feature prominently, leveraging combinations of RGB, depth, optical flow, skeletal data, and feature-based representations such as Hand Energy Images (HEIs) to capture richer sign descriptions. More recently, methods like SignCLIP have introduced language-agnostic, contrastive learning frameworks that align sign videos and textual inputs in a shared embedding space,

enabling scalable and flexible recognition. Across these methods, key trends include the use of extensive preprocessing, modality-specific architectures, and training optimizers such as Stochastic Gradient Descent (SGD), all aimed at enhancing robustness and generalization.

2.5.1 Recognizing American Sign Language Gestures from within Continuous Videos

In 2018, Ye et al. [55] proposed a 3DRCNN hybrid model that combines 3D Convolutional Neural Networks (3DCNNs) with Fully Connected Recurrent Neural Networks (FC-RNNs) to recognize American Sign Language (ASL) gestures in continuous video streams. The model is designed to both classify signs and localize their temporal boundaries within the video. The 3DCNN component captures spatiotemporal features from RGB, motion (optical flow), and depth data, while the FC-RNN captures sequential dependencies between video clips. To further enhance performance, features from the three modalities are concatenated before being passed to the RNN. A greedy linking method is used to merge video clips with the same label into coherent segments.

The authors also introduce a new multi-modal ASL dataset containing RGB, depth, and optical flow channels, along with full temporal annotations for individual signs within continuous sequences. Experimental results demonstrate that the 3DRCNN outperforms baseline models like C3D, LRCN, and Two-Stream CNNs, achieving 69.2% accuracy (person-dependent) and 65.8% (person-independent) on a subset of 27 ASL signs. Their results highlight the importance of combining both spatial and temporal information across multiple data modalities for improved recognition and segmentation performance in real-world ASL applications.

2.5.2 Deep convolutional neural networks for sign language

Also in 2018, Rao et al. [56] presented a deep learning-based system for recognizing Indian Sign Language (ISL) gestures using a selfie video input approach. The authors constructed a custom dataset containing 200 ISL signs, performed by five native signers in five different orientations, resulting in 300,000 video frames. The proposed method uses a deep convolutional neural network (CNN) architecture with four convolutional layers, two stochastic pooling layers, and a SoftMax classifier. Various pooling strategies were tested, and stochastic pooling provided the best results in terms of accuracy and generalization.

Training was conducted in three batches using one, two, and three user sets, respectively, with performance evaluated on unseen signer data. The system achieved a maximum recognition accuracy of 92.88%, outperforming traditional classifiers like Mahalanobis Distance, AdaBoost, and both shallow and deep ANNs.

The architecture was optimized for mobile platforms, making it suitable for real-time sign recognition on smartphones. The study highlights the effectiveness of CNNs combined with stochastic pooling for robust and efficient sign language recognition in mobile applications.

2.5.3 Isolated sign language recognition using Convolutional Neural Network hand modelling and Hand Energy Image

In 2019, Lim et al. [57] presented an isolated sign language recognition framework centered on a two-phase process: hand tracking and hand representation. The hand tracking is performed using a particle filter, which combines hand motion information and CNN-based pre-trained hand models to accurately detect hand positions across frames. Once detected, the hand regions are used to generate a Hand Energy Image (HEI), a compact representation obtained by averaging segmented hand regions over time.

The system was evaluated on two benchmark datasets, RWTH-BOSTON-50 and ASLLVD, demonstrating that the CNN-based hand tracking (CNNT) significantly outperformed traditional methods such as Kalman filters, Dynamic Programming, and color-based tracking in terms of accuracy and robustness. The Hand Energy Image (HEI) representation further enhanced performance by capturing the temporal dynamics of hand movements. Sign recognition was performed using a nearest-neighbor approach, where gestures were classified based on the minimum distance between the vectorized HEIs of the test and training samples. This technique achieved up to 89.33% accuracy on RWTH-BOSTON-50 and showed strong resilience to signer variability and occlusion, indicating its suitability for signer-independent recognition tasks.

2.5.4 On the role of multimodal learning in the recognition of sign language

In 2019, Ferreira et al. [58] proposed a novel multimodal learning approach for static sign language recognition by combining data from Kinect (color and depth) and Leap Motion sensors. It introduces EENReg (End-to-End Network with Regularization), a deep neural network that jointly learns modality-specific and shared feature representations. The architecture comprises separate private and shared convolutional streams for each modality, trained using a custom loss function that encourages feature orthogonality and alignment across modalities. A robust hand detection pipeline is also implemented using YCbCr-based skin segmentation and depth filtering, followed by a background suppression step and extensive data augmentation.

The model was trained and evaluated on a 10-class American Sign Language dataset comprising 1400 samples with 5-fold signer-independent cross-validation. Results show that multimodal fusion methods consistently outperform single-modality ones, with EENReg achieving 97.66% accuracy, surpassing previous state-of-the-art methods. Notably, the study reveals that combining Leap Motion's structural hand data with Kinect's visual modalities leads to strong complementary information, improving recognition performance. This work demonstrates the potential of deep multimodal learning in enhancing the generalization and robustness of sign language recognition systems.

2.5.5 Deep learning-based sign language recognition system for static signs

In 2020, Wadhawan and Kumar [59] introduced a robust sign language recognition system for 100 static signs of Indian Sign Language (ISL), utilizing a custom-built dataset of 35,000 RGB images captured under various environmental conditions. The authors design and evaluate nearly 50 different CNN models, fine-tuning hyperparameters such as the number of convolutional layers, filters, and optimizers. The architecture includes standard components, such as convolutional layers, ReLU, max-pooling, dropout, and fully connected layers, culminating in a SoftMax classifier for multi-class recognition. Preprocessing steps include image resizing and normalization. Experimental results show that reducing the CNN depth to four layers while using Stochastic Gradient Descent (SGD) leads to optimal performance, achieving a training accuracy of 99.90% and validation accuracy of 98.70% on grayscale images, and 99.72% training accuracy and 98.56% validation accuracy on colored images.

The system is evaluated not only by accuracy but also by precision, recall, and F1-score, and it significantly outperforms several existing ISL recognition methods based on machine learning (e.g., KNN, SVM, ANN). The authors also demonstrate the system's effectiveness across different optimizers (Adam, RMSProp, Adagrad), concluding that SGD provides the best generalization. One of the notable contributions of this work is the large-scale static ISL dataset and extensive comparative experimentation, which establish this CNN-based approach as a state-of-the-art benchmark for static sign recognition. Future work includes extending the model to dynamic sign recognition and deploying it in real-time applications on mobile devices.

2.5.6 SignCLIP: Connecting Text and Sign Language by Contrastive Learning

In 2024, Jiang et al. [60] presented SignCLIP, a contrastive learning framework designed to bridge the gap between spoken language text and sign language videos by projecting both into a shared latent space. Unlike most sign language processing models that rely on labeled datasets and gloss annotations, SignCLIP leverages video-text pairs from the large-scale multilingual Spreadthesign dictionary (∼500k videos across 41 sign languages). By adapting the CLIP (Contrastive Language–Image Pretraining) architecture, the authors train a dual-encoder model with a 3D-CNN-based video encoder and a BERT-based text encoder. Notably, they explore both raw video inputs and pose-based representations using MediaPipe Holistic, finding the latter to be more efficient and interpretable for downstream tasks. A smaller variant called FingerCLIP is also tested on isolated fingerspelling recognition, achieving perfect retrieval accuracy using dominant hand keypoints and augmentation strategies.

SignCLIP shows excellent in-domain performance on isolated sign recognition tasks, especially when trained on pose data, achieving top-1 accuracy up to 0.40 and top-10 accuracy up to 0.83 on 4,531 unique signs. However, its zero-shot performance on out-of-domain datasets (like PopSign and ASL Citizen) is limited due to domain shifts in both visual and textual modalities. The model excels in few-shot and fine-tuned scenarios, sometimes outperforming previous state-of-the-art methods, particularly on datasets such as PopSign ASL and ASL Citizen. The paper concludes that multilingual pretraining, pose-based inputs, and contrastive objectives offer a promising path toward scalable, language-agnostic sign language recognition systems. Moreover, it emphasizes the potential of SignCLIP as a universal embedding model for sign languages, with implications for recognition, translation, and retrieval tasks.

2.5.7 Considerations

A noteworthy pattern across the reviewed studies is the influence of dataset choice on reported performance. Systems trained on custom-built or controlled datasets often demonstrate high accuracy, likely due to domain-specific optimizations and reduced variability. Conversely, models evaluated on public or more diverse datasets tend to report lower but more realistic performance, highlighting the challenge of generalization in real-world scenarios. This trade-off becomes especially apparent in cross-domain evaluations, as seen in SignCLIP, where zero-shot performance drops significantly despite strong in-domain results. These findings underscore the need for standardized, large-scale benchmarks and transparent reporting to enable fair comparisons, encourage robust model development, and advance sign language

recognition systems that are inclusive, scalable, and truly deployable across diverse settings.

Chapter 3

Methodology

3.1 Overview

This chapter presents the methodology adopted for building a sign language recognition system focused on the classification of isolated signs in the Italian Sign Language (LIS). The proposed pipeline combines computer vision techniques for hand localization and segmentation with deep learning-based classification, aiming to create a lightweight and effective model suitable for real-world deployment.

The system is developed and evaluated using the A3LIS-147 dataset, which contains video recordings of isolated LIS signs performed by multiple signers. Each video is first processed using MediaPipe [61], a real-time pose estimation framework, to detect and track the hand regions. Once the hand bounding boxes are extracted, a sequence of preprocessing operations is applied to enhance the input quality. The first step is adaptive thresholding, which removes the green screen background and simplifies the image by preserving only the most salient contour information of the hands. This transformation reduces the visual complexity of each frame, highlighting the key motion and shape features relevant for classification. After thresholding, a Gaussian blur is applied to smooth the result and reduce high-frequency noise, leading to cleaner and more consistent representations for generating Hand Energy Images.

The preprocessed hand crops are then used to generate a Hand Energy Image (HEI) for each hand. This representation captures temporal dynamics and spatial movement patterns by averaging the segmented hand frames into a single composite image. Each HEI, left and right, is then passed through a separate Convolutional Neural Network (CNN) trained to classify the corresponding sign. The final prediction is obtained by averaging the confidence scores from both hands, enabling effective bi-manual integration.

This methodology emphasizes the use of pose-guided cropping and temporal

feature condensation while incorporating targeted preprocessing steps to improve segmentation quality and model robustness. The pipeline is designed to maintain low computational complexity while delivering strong classification performance across a representative set of isolated signs. To evaluate the generalization capability of the model, a real-world video was recorded by the author performing the target signs in a natural environment, without the standardized green screen background, allowing for testing under less controlled conditions.

3.2 Database

Figure 3.1: Sample frame (*inviare SMS*) from the A3LIS-147 dataset [41].

Figure 3.2: Sample frame (*inviare SMS*) from the real-world test dataset recorded by the author.

This work utilizes a subset of the A3LIS-147 dataset, a publicly available video corpus of Italian Sign Language (Lingua Italiana dei Segni – LIS). The dataset was developed by Fagiani et al. [41] in collaboration with the Ente Nazionale dei Sordi (ENS) and contains 147 distinct isolated signs, organized into six thematic categories relevant to daily life. All signs are recorded under repeatable and controlled conditions, specifically designed for automatic sign recognition and synthesis research.

The full corpus includes recordings from 10 native LIS signers (7 males and 3 females), aged between 18 and 43 years with an average age of 29 years, and heights ranging from 156 cm to 190 cm (average of 172 cm). Each signer performed

all 147 signs individually. Each video clip begins and ends with a standardized "silence" pose, recorded for consistency across samples. Videos were recorded in a controlled environment with uniform lighting, green chroma-key background, and a 25 fps, 720×576 pixel resolution, using a frontal commercial camera setup.

For this thesis, a subset of 148 videos of 14 signs from the Common Life category was selected, comprising signs frequently used in everyday interactions: *abitare* (to live), *acqua* (water), *affitto* (rent), *banca* (bank), *caldo* (hot), *casa* (house), *cibo* (food), *data* (date), *freddo* (cold), *interprete* (interpreter), *inviare SMS* (send SMS), *lingua dei segni* (sign language), *litro* (liter), and the idle or silence position used as a background class. This reduced vocabulary allows for focused development and evaluation of the proposed recognition system on semantically meaningful and visually distinct signs. The Figure 3.1 shows a frame of the *inviare SMS* (send SMS) sign performed by one male sign.

To ensure that the classification model focuses on the meaningful portion of each sign, the original videos from the A3LIS-147 dataset were manually trimmed to remove the initial and final silence poses, leaving only the active signing segment. However, selected samples of the silence position were still retained and explicitly added to the training data as a separate class. This allows the model to also identify idle or transition periods, improving its applicability in continuous or real-time settings.

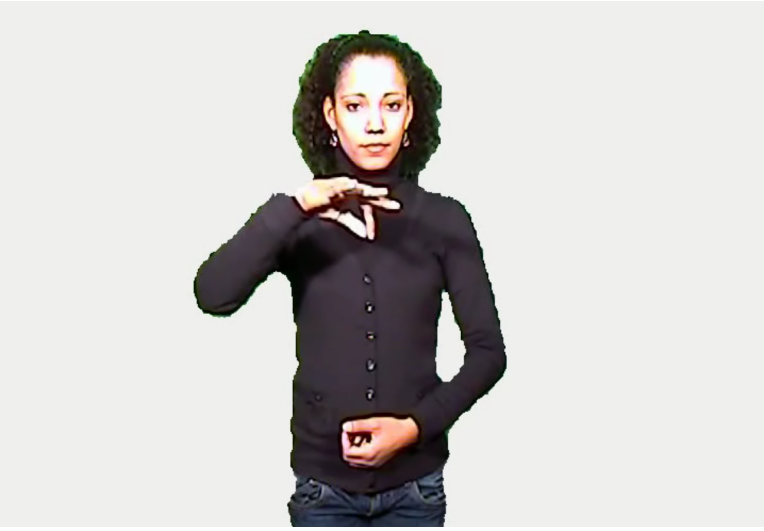

Figure 3.3: Sample frame (*affitto*) from the modified A3LIS-147 dataset with light background color.

In addition to the original dataset, an alternative version of the A3LIS-147 subset was generated by replacing the green screen background with a light color close to white (Figure 3.3). This modification aimed to simulate a more realistic visual

environment by reducing the stark contrast between the signer and the background, thereby better approximating real-world recording conditions. The purpose of this adapted dataset was to evaluate the model's robustness when transitioning from highly controlled settings to more naturalistic visual contexts, bridging the gap between laboratory-trained systems and their deployment in everyday scenarios.

In addition to using the A3LIS-147 dataset, a separate set of real-world test videos was recorded by the author to evaluate the model's generalization capability outside of the controlled dataset conditions. 61 videos were captured in a bedroom setting using a smartphone at 720×1280 resolution and 29 frames per second, without a green screen background. The recorded signs include: *abitare* (4 samples), *acqua* (4), *affitto* (4), *banca* (4), *caldo* (5), *casa* (4), *cibo* (3), *data* (5), *freddo* (5), *idle* (5), *interprete* (5), *inviare* (6), *lingua* (3), and *litro* (4). Although the author is not a native LIS signer, each sign was carefully reproduced by referencing standard LIS materials to simulate realistic usage and test the system in an uncontrolled environment. The Figure 3.2 shows a frame of the *inviare SMS* (send SMS) sign performed by the author. A clear contrast can be observed between Figures 3.1 and 3.2, highlighting the difference between the controlled, studio-like conditions of the A3LIS-147 dataset and the more variable, real-world environment of the author-recorded videos.

3.3 Image Preprocessing

To ensure high-quality input for classification and to emphasize relevant visual features, a preprocessing pipeline was designed to operate on each video frame prior to feature extraction and classification performed by the convolutional neural network. The goal of this stage is to isolate and enhance the hand regions responsible for conveying sign information while suppressing irrelevant background noise and visual clutter. The preprocessing procedure is composed of four main components: hand tracking, which localizes and crops the hand areas using pose estimation; adaptive thresholding, which removes the background and retains essential contours; Gaussian filtering, which smooths the segmented images and reduces noise; and the generation of Hand Energy Images (HEI), which condense motion and spatial features across the video sequence into a single composite representation. Each of these components plays a critical role in shaping the input to the classification model and improving its robustness and performance.

3.3.1 Hand Tracking with MediaPipe

Hand tracking is a crucial first step in the preprocessing pipeline, responsible for localizing and cropping the hand regions used to build the Hand Energy Images (HEIs). In this work, the hand tracking task is performed using the *MediaPipe Hands*

solution, a real-time hand perception framework developed by Google. MediaPipe detects and tracks up to two hands per frame, returning 21 3D landmarks for each hand, including positions for the wrist, finger joints, and fingertips.

To define a stable and consistent bounding box around each hand, two specific landmarks are used: the wrist (landmark ID 0) and the metacarpophalangeal joint of the middle finger (landmark ID 9). The Euclidean distance between these two points serves as a reference for estimating the size of the square bounding box centered at the middle finger base. Let (x_0, y_0) and (x_9, y_9) denote the image coordinates of the wrist and the middle finger base, respectively. The distance d between these two landmarks is computed as:

$$d = \sqrt{(x_9 - x_0)^2 + (y_9 - y_0)^2} \tag{3.1}$$

To ensure the box fully captures the spatial extent of the hand, this value is scaled by a factor of 1.25:

$$s = 1.25 \cdot d \tag{3.2}$$

The bounding box is then defined as a square of side length $2s$, centered at (x_9, y_9). The top-left and bottom-right corners of the square are computed as:

$$\text{start} = (x_9 - s, y_9 - s), \quad \text{end} = (x_9 + s, y_9 + s) \tag{3.3}$$

The computed bounding box is used to crop the hand region from the original frame. These cropped images are stored in temporal buffers separately for the left and right hands. The buffered sequences are then used to generate Hand Energy Images by averaging the preprocessed cropped hand frames over time.

This approach ensures that each HEI captures the spatial and motion characteristics of the sign, while reducing the influence of background or non-manual features. The use of landmark-based dynamic cropping leads to consistent localization even in the presence of signer movement or hand shape variation.

3.3.2 Adaptive Thresholding

Following hand tracking and cropping, each hand region undergoes an image segmentation step to remove background noise and emphasize the hand's structural contours. This is achieved using `cv2.adaptiveThreshold`, a local thresholding method provided by OpenCV, which dynamically adjusts the threshold value across different regions of the image based on local intensity distributions. Unlike global thresholding techniques that apply a single fixed value, adaptive thresholding is more robust to illumination changes and shadows, features often present in real-world video frames.

In this work, the cropped hand image is first converted to grayscale before undergoing adaptive Gaussian thresholding. This technique dynamically computes a local threshold for each pixel based on the intensity distribution in its surrounding neighborhood. The binarization process follows the rule:

$$\mathrm{dst}(x,y) = \begin{cases} 255 & \text{if } I(x,y) > T(x,y) \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

Where $I(x,y)$ is the grayscale intensity at pixel location (x,y), and $T(x,y)$ represents the adaptive threshold calculated from the weighted sum of intensities in a local neighborhood around that point.

In particular, the method applies a Gaussian-weighted window to the neighborhood, assigning greater importance to pixels closer to the center. This weighting is defined by the two-dimensional Gaussian function:

$$w(i,j) = \exp\left(-\frac{i^2+j^2}{2\sigma^2}\right) \tag{3.5}$$

Here, (i,j) are the offsets from the center pixel and σ is the standard deviation, which determines the spread of the Gaussian distribution. The threshold $T(x,y)$ is then computed as the sum of the weighted intensity values within the window, and a constant C is subtracted to control the sensitivity of the binarization:

$$T(x,y) = \left(\sum_{i,j} w(i,j) \cdot I(x+i, y+j)\right) - C \tag{3.6}$$

This approach improves robustness to local lighting variations and ensures that meaningful edges and hand contours are preserved. By simplifying the image content and removing background noise, adaptive thresholding contributes to a more focused and informative input for the subsequent generation of Hand Energy Images.

This technique, particularly the Gaussian-weighted neighborhood strategy, is a well-established method in image processing for edge-preserving binarization [62].

Figure 3.4 illustrates the result of this process. The left image shows a raw frame from the A3LIS-147 dataset with its original green background, while the right image shows the corresponding output after adaptive thresholding. As observed, the hand contours are preserved clearly, and the background is removed effectively, leaving a simplified yet informative representation that enhances the downstream Hand Energy Image (HEI) computation.

(a) Raw frame (b) After adaptive thresholding

Figure 3.4: Effect of adaptive thresholding for background removal on a frame from A3LIS-147 dataset.

3.3.3 Gaussian Blur

After adaptive thresholding is applied to remove the background and emphasize hand contours, a Gaussian blur is introduced as a complementary preprocessing step to further smooth the image and suppress residual noise. This operation uses a Gaussian kernel structurally identical to the one employed in the adaptive thresholding method for computing local threshold values (see Equation 3.5). In both cases, the kernel assigns greater weight to pixels nearer the center, promoting spatial coherence.

When applied as a blurring filter, the Gaussian kernel performs a convolution across the image, replacing each pixel value with a weighted average of its neighbors. This process reduces high-frequency noise and softens sharp edges, particularly around segmented hand regions. The resulting images exhibit improved visual consistency, which is crucial for generating Hand Energy Images (HEIs) through temporal averaging.

The effects of this procedure are illustrated in Figures 3.5 and 3.6. Figure 3.5 shows an HEI constructed without Gaussian blur, where aliasing and abrupt edge transitions are evident. In contrast, Figure 3.6 demonstrates how the blur mitigates pixel-level artifacts and produces a smoother, more coherent representation of hand motion. This enhancement improves visual clarity and supports more stable and discriminative feature extraction during classification.

Figure 3.5: HEI without Gaussian blur

Figure 3.6: HEI with Gaussian blur

Further benefits of applying Gaussian blur after adaptive thresholding are evident in Figures 3.7 and 3.8. Both HEIs were generated using adaptive thresholding to isolate hand contours and remove background content. However, the image in Figure 3.7 exhibits sharper edges and greater visual noise due to residual high-frequency components and aliasing. In contrast, the blurred version in Figure 3.8 displays smoother contours and fewer artifacts, resulting in a more continuous and compact visual representation of motion. This refinement enhances sample consistency and reduces the likelihood of overfitting to spurious edge patterns, making it advantageous for neural network-based classification.

Figure 3.7: HEI with adaptive thresholding (no blur)

Figure 3.8: HEI with adaptive thresholding and Gaussian blur

In this work, Gaussian blur is implemented using OpenCV's standard library. The kernel size and standard deviation are empirically tuned to balance noise

suppression with contour preservation. When applied consistently across all frames, this filter enhances the spatial stability of hand regions while maintaining essential structural features, ultimately contributing to more robust and reliable sign classification.

3.3.4 Hand Energy Image (HEI)

Inspired by the work of Lim et al. [57], the Hand Energy Image (HEI) is adopted in this project as a compact representation of temporal and spatial hand movement patterns in isolated sign videos. Originally proposed for summarizing a sequence of hand gestures into a single image, HEI encodes the spatiotemporal characteristics of hand motion by averaging segmented hand regions across all video frames. This approach transforms temporal dynamics into a static but information-rich representation, enabling efficient sign classification using image-based models.

3.3.5 HEI Generation

To construct the HEI, the signer's hand is first localized in each frame using bounding boxes obtained via hand tracking (see Section 3.3.1). To ensure uniformity in the dimensions and proportions of the cropped hand regions across the entire sequence, the hand is cropped using the largest bounding box detected among all frames of the sequence. This guarantees that all hand crops used for HEI generation share the same spatial size and proportion, preventing issues that could arise if each frame were resized individually based on its own bounding box dimensions. Such independent resizing could introduce distortions or scale inconsistencies, especially when the bounding boxes vary significantly. By anchoring the crop size to the maximal bounding box, no part of the hand is clipped, and all frames are properly aligned for averaging. This strategy contributes to the visual consistency and semantic integrity of the resulting HEI. Each hand crop is then preprocessed using adaptive thresholding and Gaussian blur to improve the consistency and clarity of shape contours. Let S_t represent the segmented hand region at time t, and T be the total number of frames in the video. The HEI is computed as:

$$H = \frac{1}{T} \sum_{t=1}^{T} S_t \tag{3.7}$$

This averaging operation emphasizes spatial regions that are most consistently occupied by the hand across time, effectively encoding motion patterns and structural consistency. In this work, separate HEIs are generated for the left and right hands, following the bilateral tracking performed in the preprocessing stage.

The resulting HEI images serve as inputs to dedicated Convolutional Neural Networks (CNNs), which are trained to classify isolated signs based on the spatial-temporal hand signature. This representation allows the network to focus on overall hand movement and articulation characteristics while being robust to minor frame-wise variations.

Figure 3.9 (adapted from [57]) illustrates an example of HEI generated from a sequence of hand images. The cumulative intensity in the HEI highlights frequent motion zones and preserves essential gesture features for accurate classification.

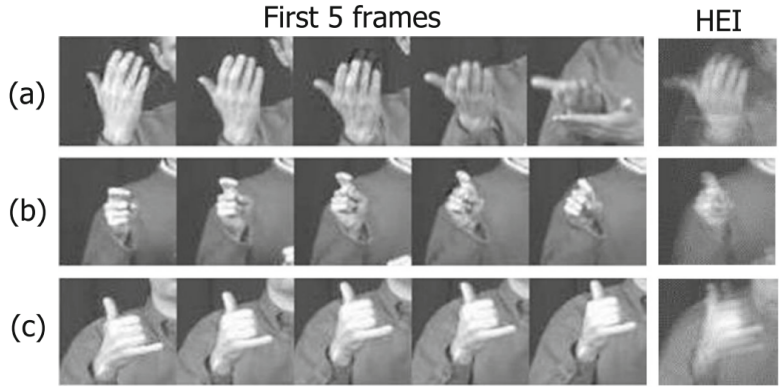

Figure 3.9: Illustration of Hand Energy Image (HEI) adapted from [57]. The sequence shows several segmented frames of a hand gesture, and the right image shows the resulting HEI obtained by averaging them.

Figure 3.10 illustrates the visual process of generating the Hand Energy Image (HEI) for the sign *inviare SMS*. A sequence of cropped right-hand frames is shown on the left, demonstrating the temporal evolution of the sign. These frames are first spatially aligned and then averaged to form the HEIs shown on the right. HEI 1 is produced by averaging the original cropped images after applying Gaussian blur, which helps suppress noise and stabilize local variations across frames. HEI 2, on the other hand, is generated by first applying adaptive thresholding to highlight hand contours and remove background content, followed by Gaussian blurring to smooth the resulting edges. This combination produces a sharper, contour-focused representation of hand motion, potentially more informative for neural network classification. Both approaches benefit from temporal condensation, but HEI 2 emphasizes the structural outlines, while HEI 1 captures a softer, intensity-based motion trace.

Figure 3.10: Sequence of frames and two resulting HEIs for the right hand performing the sign *inviare SMS*. HEI 1 is obtained from blurred RGB frames; HEI 2 is constructed from thresholded and then blurred frames.

3.3.6 HEI Datasets

To evaluate the impact of different preprocessing techniques on the quality and performance of the HEI-based recognition system, several variants of the HEI dataset were generated throughout this work. Each variant corresponds to a specific combination of preprocessing steps, such as background modification, adaptive thresholding, and Gaussian blur, and is constructed by applying a consistent HEI generation pipeline to differently processed video inputs.

The datasets are grouped into two main sources: the original A3LIS-147 dataset and the author-recorded real-world videos. For the A3LIS-147 data, variants were created using both the original green-screen background and a modified version with a lighter tone to better simulate real-world lighting conditions. Within each background category, HEIs were generated under different configurations, including raw input, adaptive thresholded input, and with or without Gaussian blurring.

Importantly, a distinction exists between the left and right hand datasets. Several of the signs in the chosen vocabulary, namely acqua, cibo, affitto, and lingua dei segni, are typically performed using only the right hand. As a result, the left hand remains idle during these signs. In this work, HEIs corresponding to the left hand for these signs were assigned to the "idle" class, reducing the number of meaningful gesture labels for the left hand channel from 14 to 10. This asymmetry is reflected in the distribution of samples across the two channels and has implications for training and evaluation, especially in terms of class balance and recognition accuracy.

The following table (Table 3.1) summarizes all processed datasets used in this study, indicating the preprocessing applied to each variant and the number of samples used for training and testing in both left and right hand channels.

Table 3.1: Summary of HEI datasets generated for experimentation

Dataset	Background	Adaptive thresholding	Gaussian Blur
A3LIS	Green	No	No
A3LIS-GB	Green	No	Yes
A3LIS-LB	Light	No	No
A3LIS-LB-GB	Light	No	Yes
A3LIS-AT-GB	Green	Yes	Yes
Real-world	Real	No	No
Real-world-GB	Real	No	Yes
Real-world-AT-GB	Real	Yes	Yes

Note: **LB** = Light Background; **GB** = Gaussian Blur; **AT** = Adaptive Thresholding.

Additionally, for consistency between training and evaluation conditions, the real-world test datasets were selected to mirror the preprocessing strategies applied during model training, as described below.

For models trained on the A3LIS and A3LIS-LB datasets, the real-world evaluation was conducted using the **Real-world** dataset without additional preprocessing. For models trained on A3LIS-GB and A3LIS-LB-GB, the real-world test set used was the **Real-world-GB** dataset, applying Gaussian blur to match the training conditions. Finally, models trained on A3LIS-AT-GB were evaluated using the **Real-world-AT-GB** dataset, which underwent both adaptive thresholding and Gaussian blur. This approach ensured coherence between the training and testing environments, allowing for a more accurate assessment of the generalization capabilities of each model.

3.4 Classification Model

The proposed sign language recognition system prioritizes a lightweight yet effective convolutional neural network (CNN) architecture suitable for real-world deployment. Given the need for low computational complexity, especially for scenarios like mobile or embedded systems, we adopt compact custom-designed CNNs instead of relying on large pre-trained models.

3.4.1 CNN Architectures

Two versions of the CNN architecture were explored: **model version v0** (Figure 3.11) and **model version v1** (Figure 3.12). Both versions begin with a rescaling layer to normalize pixel values, followed by a series of convolutional layers and max-pooling operations to extract spatial features from the Hand Energy Image (HEI) input.

Model v0 uses three consecutive standard convolutional layers with 32 filters each, followed by a fully connected layer of 128 units. Model v1 introduces separable convolutions, gradually increasing the number of filters (32, 64, and 128) across layers to improve representational capacity while keeping the parameter count low. Both models conclude with a dense classification layer using a SoftMax activation function to output class probabilities, while ReLU was utilized as activation function for the other layers.

In addition, each architecture was tested with and without a *dropout* layer before the final classification layer, enabling evaluation of its effect on model generalization. These configurations are referred to in the experimental analysis by appending the suffix "-dropout" where applicable. Both models are compiled with the Adam optimizer and trained using sparse categorical cross-entropy at a learning rate of 0.001.

3.4.2 Combined Hand Model Integration

Given the bi-manual nature of many signs in LIS, predictions from the left and right hand CNNs are combined to enhance recognition performance. This is particularly relevant in signs involving both hands, while also accommodating one-handed signs in which the non-dominant hand remains idle.

The final classification is computed using a combined model class, which merges predictions from the independently trained right-hand and left-hand networks. The model outputs a weighted combination of the respective probability distributions P_R (right hand) and P_L (left hand):

$$P_{\text{combined}} = \frac{w_R \cdot P_R + w_L \cdot P'_L}{Z} \tag{3.8}$$

Where:

- P'_L represents a potentially scaled version of P_L,
- w_R and w_L are weight coefficients for the right and left hand models,
- Z is a normalization constant ensuring $\sum P_{\text{combined}} = 1$.

This combination follows a series of logical conditions:

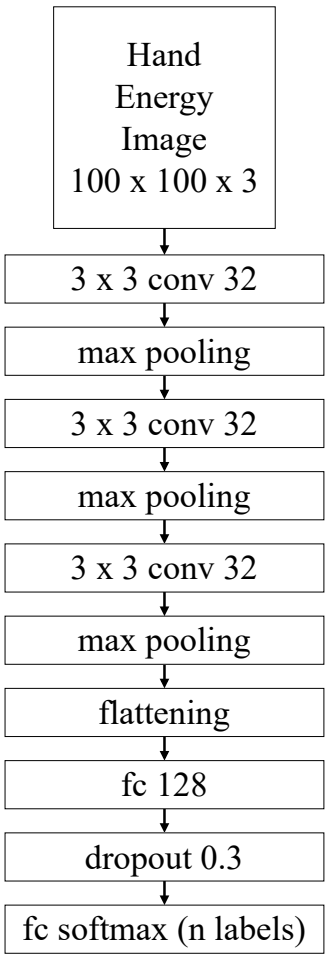

Figure 3.11: Architecture of the CNN model v0.

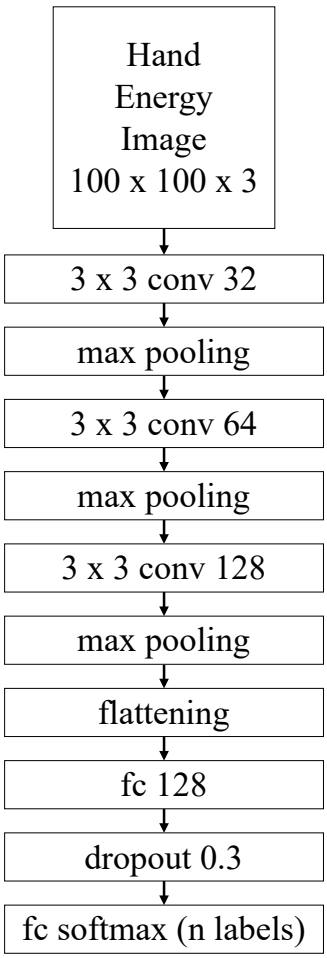

Figure 3.12: Architecture of the CNN model v1.

1. If the input to the right-hand model is missing (e.g., no HEI generation caused by occlusion), P_R is excluded from the final prediction.

2. If the left-hand model's top-1 prediction is the *idle* label or its input is missing, P_L is also excluded.

3. If the top-1 prediction of the left-hand model appears within the top-N_R predictions from the right-hand model, its output probabilities are multiplied by a boosting factor w'_L. This condition promotes mutual agreement across hands. N_R and w'_L model parameters that can be tuned like w_R and w_L.

These rules account for the structural asymmetry present in the dataset: signs such as *acqua*, *cibo*, *affitto*, and *lingua dei segni* are typically performed with only the right hand. The left hand remains idle, and its corresponding HEIs are assigned to the *idle* class, effectively reducing the number of active sign classes for the left hand to 10. By introducing logic to suppress idle or uninformative contributions, the model maintains robustness while effectively integrating information from both hands.

3.4.3 Model Design Considerations

Using compact, custom CNN architectures, rather than large-scale pre-trained models, was a deliberate choice to balance accuracy with computational efficiency. The system is designed for real-time scenarios and constrained hardware environments like mobile devices or embedded platforms.

The independent hand models allow for flexibility in sign interpretation, while the combined model mechanism merges their outputs using contextual rules, improving overall robustness without adding significant computational overhead. By accounting for asymmetric gesture involvement and optimizing for lightweight deployment, the architecture offers a practical and scalable solution for isolated sign language recognition.

3.5 Evaluation Metrics

The evaluation of the sign language recognition models developed in this work relies on a set of metrics specifically selected to align with the nature of the task. Since the models perform isolated sign classification without access to linguistic context or language model priors, it is critical to assess their predictive confidence and ranking ability, rather than relying solely on hard top-1 decisions.

The primary evaluation metrics used are:

- **Top-N Accuracy**: Top-N accuracy measures the percentage of test samples where the true label appears among the model's N most confident predictions. In this work, top-1, top-2, and top-3 accuracies are reported:
  - **Top-1 Accuracy** (overall classification accuracy): The model is considered correct only if the true label is the one with the highest predicted probability.
  - **Top-2 Accuracy**: The true label is considered correctly predicted if it appears among the two classes with the highest predicted probabilities.
  - **Top-3 Accuracy**: Similarly, the prediction is deemed correct if the true label appears within the top three most probable classes.

This family of metrics provides a richer understanding of the model's behavior, particularly useful for applications where secondary suggestions (e.g., autocorrection or user feedback systems) could be incorporated in real-world deployments.

- **Average True Probability**: In addition to discrete accuracy metrics, the average predicted probability assigned to the true label across all test samples is computed. Formally, if p_i is the predicted probability for the true label of sample i, and N is the number of samples, the average true probability is:

$$\text{Average True Probability} = \frac{1}{N} \sum_{i=1}^{N} p_i \tag{3.9}$$

  This metric captures the model's confidence in its correct predictions, offering a continuous evaluation complementary to top-N accuracies. It is particularly meaningful when high-confidence correct predictions are critical, even if the true label occasionally ranks lower among the top predictions.

- **Confusion Matrix**: To provide a more detailed analysis of model performance across different sign classes, confusion matrices were generated for each evaluation setting. The confusion matrix illustrates how predictions are distributed across all classes, highlighting common misclassifications, per-class performance, and the overall confusion patterns of the model. Visualizing these matrices enables qualitative assessment of the recognition strengths and weaknesses, particularly in distinguishing between visually similar signs or handling idle gestures.

The use of top-N accuracy metrics is justified given the nature of isolated sign recognition, where the model cannot rely on sequential dependencies or grammatical cues to refine its prediction. Unlike continuous sign language recognition systems that can leverage temporal context or language models to disambiguate uncertain predictions, isolated sign classification requires each prediction to be made purely based on the visual features captured from the input.

Moreover, reporting multiple levels of top-N accuracies acknowledges the inherent visual similarity between some signs and reflects more practical real-world usage scenarios. For instance, in assistive technologies or educational applications, offering the top-2 or top-3 suggestions could significantly enhance user experience and usability.

Similarly, the average true probability serves as a valuable measure of the system's reliability by quantifying how strongly the model "believes" in its correct predictions. High average true probability values imply that correct predictions are made with high confidence, which is desirable for downstream decision-making and building trust in the system.

3.6 Baseline Model

In order to provide a reference point for evaluating the impact of various preprocessing techniques and model enhancements, baseline models were established in this study. These baseline models serve as the fundamental performance benchmark against which all subsequent experiments and improvements are compared.

The baseline models are defined as follows:

- They are based on the two CNN architectures described previously, namely **model version v0** and **model version v1**.
- No *dropout* layers were applied in the baseline versions, allowing for the evaluation of the core network architecture without any explicit regularization.
- The models were trained on the original A3LIS-147 dataset without any additional preprocessing steps beyond the basic HEI generation pipeline. Specifically, no Gaussian blur filtering, no background color modification (green screen was maintained), and no adaptive thresholding were applied to the input videos.
- The Hand Energy Images (HEIs) used were generated directly from the raw green background videos, preserving the controlled recording conditions of the dataset without introducing any artificial variations.

By utilizing the simplest data preparation and architectural setup, these baseline models offer a clean and controlled environment for quantifying the benefits of later preprocessing steps, network regularization (such as dropout), and combined model strategies. The performance of these baseline models, measured using the evaluation metrics described earlier, provides a fundamental reference for understanding the improvements achieved by the more advanced configurations explored in subsequent sections.

3.7 Experimental Setup

The experimental protocol followed a structured multi-phase approach to maximize model generalization while ensuring comparability across different preprocessing experiments.

First, the available labeled videos for each signer and sign were split into **80% for training** (118 videos) and **20% for testing** (30 videos). This first split was performed **at the video level before generating any Hand Energy Images (HEIs)**, and a fixed random seed was used to guarantee reproducibility. This strategy ensures that all HEI datasets generated from different preprocessing

techniques (e.g., raw images, Gaussian-blurred, adaptive-thresholded) share the exact same video samples for training and testing. Consequently, the impact of preprocessing methods could be compared fairly, as all models are evaluated on consistent data splits.

After the 80%-20% split at the video level, the HEIs were generated independently for each hand (left and right) according to the selected preprocessing technique. Then, within the training set (80% of the total), a secondary split was performed at the HEI level: **82% for training** and **18% for validation**. These proportions correspond to **66% for training** and **14% for validation** relative to the full dataset.

The CNN models for the left and right hands were trained separately using this internal train-validation split. Validation was monitored during training to select the optimal number of epochs based on the best validation accuracy. After identifying the best epoch, the final models were **retrained from scratch** using the entire 80% training portion (i.e., combining previous training and validation sets) for the chosen number of epochs.

Following separate training, the **CombinedModel** was constructed. This model merges the outputs of the left-hand and right-hand classifiers by combining their probability distributions while applying specific decision rules to handle asymmetries:

- If the right-hand model output is invalid (sum equals zero), its prediction is disregarded.
- If the left-hand model output is invalid (sum equals zero) or if the top prediction corresponds to the *idle* class, its prediction is disregarded.
- If the top-1 prediction of the left-hand model appears within the top-N_R predictions of the right-hand model, the left-hand probabilities are multiplied by an additional weight factor (w'_L).

Two hyperparameters, w'_L and N_R, were tuned during this stage. Multiple configurations were tested to maximize performance on the 20% held-out "test set," which in this context served as a **secondary validation set** for hyperparameter tuning of the combined model.

Once the optimal combination strategy was determined, the final model was evaluated on the independent real-world test set recorded by the author. This final test set was designed to assess the generalization capability of the system in uncontrolled, non-studio environments.

All training and evaluation experiments were conducted on a **Dell G3-3590-A20P laptop** equipped with an **Intel Core i5 9th generation processor** and **8 GB RAM**, **without GPU acceleration**. The entire training and inference process relied solely on CPU computation.

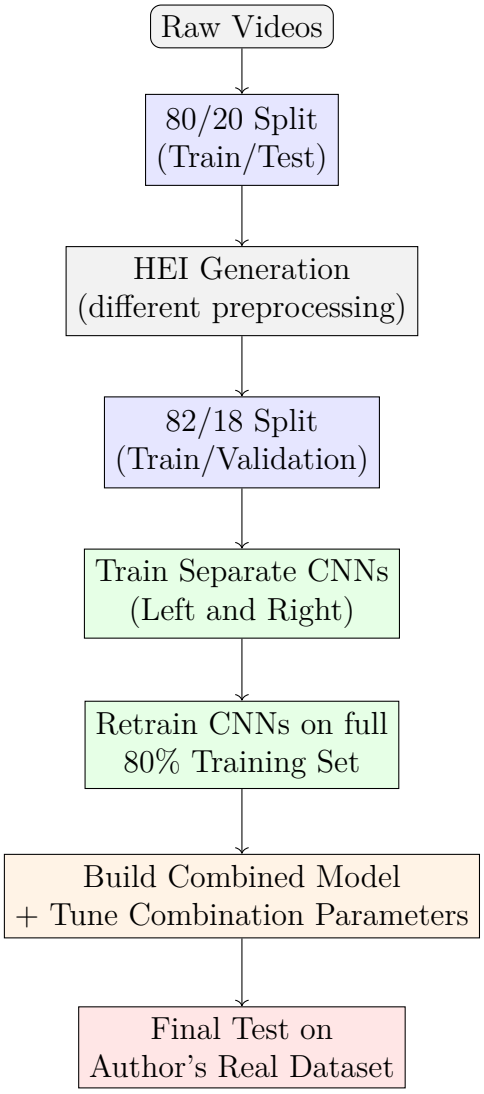

Figure 3.13: Workflow of data preprocessing, model training, and evaluation.

3.7.1 Summary of Experimental Procedure

1. **First split:** 80% train / 20% test at the **video level** using a fixed random seed (before HEI generation).
2. **HEI generation:** preprocessing applied individually to generate datasets.
3. **Second split:** 82% train / 18% validation at the **HEI level** for each hand.
4. **Model training:** separate CNNs trained for left and right hands, selecting best epoch based on validation accuracy.

5. **Model retraining:** retrain using full 80% training set with optimal number of epochs.
6. **Model combination:** build CombinedModel, tune w'_L for left-hand predictions and the N_R top labels to allow the extra weight on 20% test set (as validation).
7. **Final evaluation:** perform testing on the real-world, author-recorded dataset.

The overall workflow of the data preparation, model training, and evaluation process is summarized in Figure 3.13. This flowchart illustrates the key steps, starting from the initial 80/20 split of the original video dataset, the generation of HEI datasets with different preprocessing configurations, the separate training and validation of the left and right hand CNN models, the retraining on the full training data, the combination and fine-tuning of the ensemble model parameters, and finally the evaluation of the combined system on the real-world dataset recorded by the author.

3.8 Challenges

While the proposed methodology demonstrates promising results for isolated sign recognition, several challenges must be acknowledged regarding data, model generalization, and system performance.

First, the subset of the A3LIS-147 dataset selected for this work consists of only 148 training videos across 14 classes. Although balanced across signs, the relatively small number of samples limits the model's ability to fully capture intra-class variability, such as signer-specific differences in hand movement styles or minor variations in gesture execution. Additionally, the A3LIS-147 videos were recorded in a highly constrained environment: a green-screen background, frontal camera positioning, stable lighting conditions, and standardized clothing. These factors simplify the visual complexity of the task compared to real-world scenarios but restrict the model's exposure to more diverse signing conditions.

To partially address this, a real-world evaluation dataset was recorded by the author. However, while careful effort was made to replicate the LIS signs accurately, the signer is not a native LIS user. Therefore, the produced gestures may not fully align with authentic LIS sign execution, introducing another potential source of variability and limiting conclusions regarding model performance with expert signers. Moreover, the background, lighting, and camera angle in the author's recordings differ from the A3LIS environment, adding realism but also posing additional recognition challenges.

Another limitation arises from the hand tracking stage. Although MediaPipe provides strong real-time hand localization, it is sensitive to occlusions, motion blur,

and lighting inconsistencies. Consequently, there were occasional frames where hands were poorly localized or not detected at all, especially in the real-world videos. This affects the quality of the generated HEI representations and can degrade classification performance. Specifically, in the author-recorded test set comprising 61 videos, three instances (two for the *banca* sign and one for the *freddo* sign) resulted in failure to generate usable HEI inputs for either hand, thus excluding these examples from final testing.

Furthermore, due to the natural asymmetry between the left and right hand usage in some signs, the left-hand model received less variability for certain classes, particularly for one-handed signs. Although a strategy was adopted to assign idle positions to the left-hand HEIs for such cases, this reduces the diversity of informative samples for the left-hand classifier.

Finally, no external linguistic model or post-processing was employed to enhance predictions based on language model priors. Since the classification task treats each isolated sign independently, the model cannot rely on contextual cues that would normally be present in connected sign language communication. This isolates the evaluation purely to visual and spatial recognition, but may also limit the practical deployment of the system in continuous signing scenarios.

Despite these limitations, the results achieved suggest that the proposed system serves as a solid foundation for lightweight, real-time isolated sign recognition and highlights multiple directions for future improvement and expansion.

Chapter 4

Experiments and Results

4.1 Overview

This chapter presents the experimental evaluations conducted to assess the performance of the proposed sign language recognition system based on Hand Energy Images (HEIs). The experiments were designed to investigate the influence of different preprocessing techniques, CNN model architectures, and combination strategies on classification accuracy.

First, baseline models were trained and evaluated using HEIs generated from the A3LIS-147 dataset without any preprocessing, serving as reference points for subsequent comparisons. Then, the impact of specific preprocessing steps, including Gaussian blur, background color modification, and adaptive thresholding, was analyzed through systematic experiments across multiple dataset variants.

The models were trained separately for each hand and subsequently integrated using the CombinedModel strategy. Different hyperparameter configurations for the combined prediction, namely w'_L and N_R, were explored to optimize system performance.

Finally, the best-performing models were tested on a real-world dataset recorded by the author under uncontrolled environmental conditions, providing insights into the system's generalization capability beyond the constrained laboratory environment of the original dataset.

The following sections detail the experimental setup, baseline performance, evaluations across different dataset variants, and results obtained on the real-world dataset.

4.2 Experimental Setup

The experiments were conducted following the training, validation, and evaluation procedures described in Chapter 3. Separate CNN models were trained for left and right-hand Hand Energy Images (HEIs) and later combined into a unified classifier using the CombinedModel approach. Hyperparameters specific to the model combination were tuned on the validation set, and the final evaluation was carried out on a real-world dataset recorded by the author.

4.3 Baseline Results

The baseline models were evaluated using the Hand Energy Images (HEIs) generated from the original A3LIS dataset, without applying any additional preprocessing such as Gaussian blur, background color modification, or adaptive thresholding. Both model architectures (Model v0 and Model v1) were tested with and without the use of a dropout layer before the final classification layer.

Performance was assessed separately on two different datasets:

- The internal A3LIS test set, composed of 20% of the original videos, serving as a validation benchmark for isolated sign recognition under controlled conditions.

- The real-world dataset recorded by the author, used to assess the generalization capabilities of the models to unconstrained environments.

The evaluation metrics considered include Top-1, Top-2, and Top-3 accuracy, along with the average true probability, the mean predicted probability assigned to the correct class.

Table 4.1: Baseline model performance trained on A3LIS dataset without preprocessing

Metric / Parameter	**v0 ND**	**v0 30%D**	**v1 ND**	**v1 30%D**
$w'_L \mid N_R$	4 | 3	2 | 5	4 | 5	2 | 3
Test Top-1 Accuracy (%)	90.00	86.67	70.00	86.67
Test Top-2 Accuracy (%)	90.00	93.33	86.67	96.67
Test Top-3 Accuracy (%)	100.00	93.33	86.67	100.00
Test ATP (%)	63.50	67.74	62.51	57.23
Real Top-1 Accuracy (%)	18.97	41.38	18.97	22.41
Real Top-2 Accuracy (%)	32.76	44.83	20.69	37.93
Real Top-3 Accuracy (%)	39.66	51.72	31.03	44.83
Real ATP (%)	17.47	30.16	17.72	19.00

Note: w'_L and N_R = combined model parameters; **vN** = model version vN; **ND** = No Dropout; **30%D** = 30% Dropout; **ATP** = Average True Probability.

Table 4.1 presents the baseline model performances trained on the A3LIS dataset without any preprocessing. Among the configurations, model version v0 without dropout achieved the highest Test Top-1 accuracy at 90.00%, while model version v1 without dropout yielded the lowest Test Top-1 accuracy at 70.00%. In the real-world evaluation using the author's dataset, the best Real Top-1 accuracy was 41.38%, obtained by model v0 with a 30% dropout rate. Across all models, performance on the real-world dataset was consistently lower than on the A3LIS test set. Notably, models incorporating dropout outperformed their non-dropout counterparts in real-world conditions. Furthermore, Top-2 and Top-3 accuracies showed an expected trend of improvement as the prediction window widened. These baseline results serve as a critical reference point for assessing the impact of preprocessing strategies and model combination techniques discussed in the following sections.

4.4 Evaluation on Preprocessed HEI Datasets

This section presents the evaluation results of the proposed system when trained on various versions of the HEI A3LIS-based datasets generated with different preprocessing techniques. The aim is to assess the impact of each preprocessing strategy, Gaussian Blur (GB), Light Background modification (LB), and Adaptive Thresholding (AT), on model performance.

Following the dataset definitions summarized earlier in Table 3.1, the experiments reported here cover the following processed datasets:

- **A3LIS-GB**: applying Gaussian blur over the original green background videos.
- **A3LIS-LB**: modifying the green background to a lighter color without additional filtering.
- **A3LIS-LB-GB**: combining light background modification with Gaussian blur.
- **A3LIS-AT-GB**: applying adaptive thresholding followed by Gaussian blur for background simplification and contour emphasis.

For consistency, the same evaluation metrics used for baseline models are adopted here, including Top-1, Top-2, Top-3 accuracies, and Average True Probability (ATP). Results are compared directly against the baseline established in Section 4.3, providing insight into how different preprocessing steps affect both controlled (A3LIS test set) and real-world performance.

4.4.1 Results on A3LIS-GB

Table 4.2 presents the model performance when trained and evaluated on the **A3LIS-GB** dataset, where a Gaussian blur was applied to the original videos without altering the green background. As in the baseline evaluation, both model versions (v0 and v1), with and without dropout, were tested.

Compared to the baseline results, the application of Gaussian blur alone generally led to a slight decrease in Test Top-1 accuracy for all models. The best Test Top-1 accuracy for A3LIS-GB was 86.67%, achieved by model v0 without dropout, which is slightly lower than its corresponding baseline performance (90.00%). In the real-world evaluation, a modest improvement was observed for models with dropout: model v0 with dropout reached 34.45% Real Top-1 accuracy, compared to 41.38% in the baseline, showing some instability in generalization.

Top-2 and Top-3 accuracies remained relatively high across test evaluations, suggesting that even when the top prediction was incorrect, the true label often remained among the top candidate predictions. However, the Average True Probability (ATP) showed a slight decrease compared to baseline models.

Overall, applying Gaussian blur alone did not consistently improve the recognition performance and, in some cases, introduced minor degradation, particularly on real-world data.

Table 4.2: Model performance trained on **A3LIS-GB**

Metric / Parameter	v0 ND	v0 30%D	v1 ND	v1 30%D
$w'_L \mid N_R$	2 | 3	2 | 5	3 | 4	2 | 3
Test Top-1 Accuracy (%)	86.67	80.00	76.67	80.00
Test Top-2 Accuracy (%)	93.33	86.67	83.33	86.67
Test Top-3 Accuracy (%)	96.67	100.00	90.00	93.33
Test ATP (%)	69.01	66.46	61.21	57.99
Real Top-1 Accuracy (%)	22.41	34.45	20.69	22.41
Real Top-2 Accuracy (%)	31.03	36.21	27.59	32.76
Real Top-3 Accuracy (%)	48.28	44.82	36.21	43.10
Real ATP (%)	20.36	26.23	17.95	19.36

Note: w'_L and N_R = combined model parameters; **vN** = model version vN; **ND** = No Dropout; **30%D** = 30% Dropout; **ATP** = Average True Probability.

4.4.2 Results on A3LIS-LB

Table 4.3 presents the model performance when trained and evaluated on the **A3LIS-LB** dataset, where the green background was replaced by a light color background without applying Gaussian blur. As before, both CNN model versions (v0 and v1) were evaluated with and without a dropout layer.

Compared to the baseline (A3LIS), modifying the background color slightly decreased the Test Top-1 accuracy for most configurations. The best Test Top-1 accuracy was 83.33%, achieved by model v0 without dropout, which is lower than the 90.00% recorded for the same model in the baseline. However, on the real-world dataset, a noticeable improvement in Real Top-1 accuracy was observed for most configurations. The best Real Top-1 accuracy was 41.38%, achieved by model v1 without dropout.

The Top-2 and Top-3 accuracies also remained relatively high on the A3LIS test set, showing that the models maintained reasonable ranking of correct labels despite the background modification. Real ATP values also improved compared to the baseline, indicating better average confidence on the real-world dataset, especially for models trained with the lighter background.

Thus, replacing the green screen background with a lighter color appears to have contributed to slight gains in generalization, while maintaining comparable test performance.

Table 4.3: Model performance trained on **A3LIS-LB**

Metric / Parameter	**v0 ND**	**v0 30%D**	**v1 ND**	**v1 30%D**
$w'_L \mid N_R$	2 | 3	2 | 2	2 | 3	3 | 2
Test Top-1 Accuracy (%)	83.33	80.00	80.00	76.67
Test Top-2 Accuracy (%)	90.00	86.67	90.00	80.00
Test Top-3 Accuracy (%)	90.00	90.00	93.33	90.00
Test ATP (%)	67.14	53.24	65.62	58.17
Real Top-1 Accuracy (%)	32.76	39.66	41.38	36.21
Real Top-2 Accuracy (%)	53.45	53.45	55.17	48.28
Real Top-3 Accuracy (%)	60.34	62.72	56.90	60.34
Real ATP (%)	27.55	30.09	31.54	26.10

Note: w'_L and N_R = combined model parameters; **vN** = model version vN; **ND** = No Dropout; **30%D** = 30% Dropout; **ATP** = Average True Probability.

4.4.3 Results on A3LIS-LB-GB

Table 4.4 presents the model performance when trained and evaluated on the **A3LIS-LB-GB** dataset, where the background was modified to a light color and Gaussian blur was applied as a preprocessing step.

Compared to the baseline results, the application of Gaussian blur after background modification led to mixed effects. On the A3LIS test set, model v1 without dropout achieved the highest Test Top-1 accuracy at 86.67%, matching the top baseline performances. However, Real Top-1 accuracies showed notable improvements, particularly for model v1 with dropout, which achieved the best Real Top-1 accuracy of 41.38%.

Top-2 and Top-3 accuracy scores remained consistently high across all models, indicating that correct labels were often among the top predictions even when Top-1 accuracy was lower. The Real ATP metric also improved, especially for the dropout configurations, suggesting better confidence calibration in real-world conditions.

Overall, the combination of light background replacement and Gaussian blur appeared to enhance generalization to the real-world dataset without severely compromising performance on the original A3LIS test set.

Table 4.4: Model performance trained on **A3LIS-LB-GB**

Metric / Parameter	v0 ND	v0 30%D	v1 ND	v1 30%D
$w'_L \mid N_R$	1 | 2	2 | 2	3 | 2	3 | 3
Test Top-1 Accuracy (%)	73.33	83.33	86.67	76.67
Test Top-2 Accuracy (%)	83.33	90.00	90.00	83.33
Test Top-3 Accuracy (%)	93.33	93.33	93.33	93.33
Test ATP (%)	64.79	62.55	69.86	69.83
Real Top-1 Accuracy (%)	39.66	39.66	25.86	41.38
Real Top-2 Accuracy (%)	53.45	50.00	43.10	58.62
Real Top-3 Accuracy (%)	67.24	62.07	51.72	63.79
Real ATP (%)	31.44	31.82	26.14	36.52

Note: w'_L and N_R = combined model parameters; **vN** = model version vN; **ND** = No Dropout; **30%D** = 30% Dropout; **ATP** = Average True Probability.

4.4.4 Results on A3LIS-AT-GB

Table 4.5 presents the model performance when trained and evaluated on the **A3LIS-AT-GB** dataset, where adaptive thresholding was applied to segment the hands and Gaussian blur was used for noise reduction.

Compared to the baseline results, the application of adaptive thresholding combined with Gaussian blur produced mixed outcomes. On the A3LIS test set, Test Top-1 accuracies were noticeably lower across all models, with values ranging from 63.33% to 70.00%. This indicates that the preprocessing steps may have removed some visual information necessary for classification under controlled conditions.

However, the models demonstrated interesting behavior when evaluated on the real-world dataset. Model v1 without dropout achieved the highest Real Top-1 accuracy at 48.28%, surpassing the best real-world results obtained in previous configurations. Furthermore, Real Top-2 and Top-3 accuracies showed substantial improvements, particularly for model v1 without dropout, which reached 75.86% Top-3 accuracy.

These findings suggest that although adaptive thresholding may impair performance on clean studio-like data, it significantly improves generalization to more complex, unconstrained environments by focusing on hand contours and suppressing irrelevant background features.

Table 4.5: Model performance trained on **A3LIS-AT-GB**

Metric / Parameter	v0 ND	v0 30%D	v1 ND	v1 30%D
$w'_L \mid N_R$	5 | 3	3 | 4	3 | 3	2 | 3
Test Top-1 Accuracy (%)	63.33	70.00	63.33	70.00
Test Top-2 Accuracy (%)	83.33	80.00	80.00	90.00
Test Top-3 Accuracy (%)	93.33	90.00	90.00	93.33
Test ATP (%)	54.13	61.48	52.40	59.93
Real Top-1 Accuracy (%)	39.66	41.38	48.28	39.66
Real Top-2 Accuracy (%)	48.28	51.72	60.34	56.90
Real Top-3 Accuracy (%)	56.90	62.07	75.86	68.97
Real ATP (%)	26.92	34.96	25.29	29.73

Note: w'_L and N_R = combined model parameters; **vN** = model version vN; **ND** = No Dropout; **30%D** = 30% Dropout; **ATP** = Average True Probability.

4.5 Summary of Experimental Results

This section consolidates the results obtained across different preprocessing configurations, training datasets, and model versions. The evaluation covers Top-1, Top-2, and Top-3 accuracies, as well as Average True Probability (ATP) for both the A3LIS internal test set and the real-world test set recorded by the author. Tables 4.6–4.13 summarize the results, with the best performance values for each column highlighted in blue and underlined, and the lowest performances marked in red.

In terms of Top-1 accuracy on the A3LIS test set (Table 4.6), model version v0 without dropout trained on the A3LIS dataset achieved the highest accuracy of 90.00%, while the lowest value of 63.33% was observed for models trained on A3LIS-AT-GB. When evaluated on the real-world dataset (Table 4.7), the highest Top-1 accuracy of 48.28% was achieved by model v1 without dropout trained on A3LIS-AT-GB, while the lowest was 18.97% for models v0 and v1 without dropout trained on A3LIS.

Table 4.6: Top-1 Accuracy (%) on A3LIS Test Set across Different Preprocessing Variants

Train Dataset	v0 ND	v0 30%D	v1 ND	v1 30%D
A3LIS	90.00	86.67	70.00	86.67
A3LIS-GB	86.67	80.00	76.67	80.00
A3LIS-LB	83.33	80.00	80.00	76.67
A3LIS-LB-GB	73.33	83.33	86.67	76.67
A3LIS-AT-GB	63.33	70.00	63.33	70.00

Table 4.7: Top-1 Accuracy (%) on Real-World Test Set across Different Preprocessing Variants

Train Dataset	v0 ND	v0 30%D	v1 ND	v1 30%D
A3LIS	18.97	41.38	18.97	22.41
A3LIS-GB	22.41	34.45	20.69	22.41
A3LIS-LB	32.76	39.66	41.38	36.21
A3LIS-LB-GB	39.66	39.66	25.86	41.38
A3LIS-AT-GB	39.66	41.38	48.28	39.66

For Top-2 accuracy (Tables 4.8 and 4.9), the best performance on the A3LIS test set was achieved by model v1 with dropout trained on A3LIS, reaching 96.67%. In the real-world test set, the highest Top-2 accuracy was 60.34%, obtained by model v1 without dropout trained on A3LIS-AT-GB. Models trained on A3LIS-AT-GB consistently recorded lower Top-2 accuracies on the internal test set compared to other preprocessing variants.

Table 4.8: Top-2 Accuracy (%) on A3LIS Test Set across Different Preprocessing Variants

Train Dataset	v0 ND	v0 30%D	v1 ND	v1 30%D
A3LIS	90.00	93.33	86.67	96.67
A3LIS-GB	93.33	86.67	83.33	86.67
A3LIS-LB	90.00	86.67	90.00	80.00
A3LIS-LB-GB	83.33	90.00	90.00	83.33
A3LIS-AT-GB	83.33	80.00	80.00	90.00

Table 4.9: Top-2 Accuracy (%) on Real-World Test Set across Different Preprocessing Variants

Train Dataset	v0 ND	v0 30%D	v1 ND	v1 30%D
A3LIS	32.76	44.83	20.69	37.93
A3LIS-GB	31.03	36.21	27.59	32.76
A3LIS-LB	53.45	53.45	55.17	48.28
A3LIS-LB-GB	53.45	50.00	43.10	58.62
A3LIS-AT-GB	48.28	51.72	60.34	56.90

Regarding Top-3 accuracy (Tables 4.10 and 4.11), perfect classification (100.00%) was achieved by model v0 without dropout and model v1 with dropout, both trained on A3LIS. Model v0 with dropout also achieved 100% accuracy when trained on A3LIS-GB. The best real-world Top-3 accuracy, 75.86%, was achieved by model v1 without dropout trained on A3LIS-AT-GB.

Table 4.10: Top-3 Accuracy (%) on A3LIS Test Set across Different Preprocessing Variants

Train Dataset	v0 ND	v0 30%D	v1 ND	v1 30%D
A3LIS	100.00	93.33	86.67	100.00
A3LIS-GB	96.67	100.00	90.00	93.33
A3LIS-LB	90.00	90.00	93.33	90.00
A3LIS-LB-GB	93.33	93.33	93.33	93.33
A3LIS-AT-GB	93.33	90.00	90.00	93.33

Table 4.11: Top-3 Accuracy (%) on Real-World Test Set across Different Preprocessing Variants

Train Dataset	v0 ND	v0 30%D	v1 ND	v1 30%D
A3LIS	39.66	51.72	31.03	44.83
A3LIS-GB	48.28	44.82	36.21	43.10
A3LIS-LB	60.34	62.72	56.90	60.34
A3LIS-LB-GB	67.24	62.07	51.72	63.79
A3LIS-AT-GB	56.90	62.07	75.86	68.97

In terms of Average True Probability (Tables 4.12 and 4.13), the best performance

on the A3LIS test set was 69.86%, achieved by model v1 without dropout trained on A3LIS-LB-GB. In the real-world evaluation, the highest ATP value was 36.52%, recorded by model v1 with dropout trained on A3LIS-LB-GB.

Table 4.12: Average True Probability (%) on A3LIS Test Set across Different Preprocessing Variants

Train Dataset	v0 ND	v0 30%D	v1 ND	v1 30%D
A3LIS	63.50	67.74	62.51	57.23
A3LIS-GB	69.01	66.46	61.21	57.99
A3LIS-LB	67.14	53.24	65.62	58.17
A3LIS-LB-GB	64.79	62.55	69.86	69.83
A3LIS-AT-GB	54.13	61.48	52.40	59.93

Table 4.13: Average True Probability (%) on Real-World Test Set across Different Preprocessing Variants

Train Dataset	v0 ND	v0 30%D	v1 ND	v1 30%D
A3LIS	17.47	30.16	17.72	19.00
A3LIS-GB	20.36	26.23	17.95	19.36
A3LIS-LB	27.55	30.09	31.54	26.10
A3LIS-LB-GB	31.44	31.82	26.14	36.52
A3LIS-AT-GB	26.92	34.96	25.29	29.73

4.5.1 Summary of Best Performing Models

Table 4.14 and Table 4.15 present a consolidated view of the best performing models across different evaluation metrics for both the A3LIS test set and the real-world dataset. On the controlled A3LIS test set, models trained without any preprocessing (A3LIS) consistently achieved the highest Top-1, Top-2, and Top-3 accuracies, confirming that the original green background dataset remains highly effective under laboratory-like conditions. In contrast, for the real-world dataset, the best performances were achieved by models trained with adaptive thresholding combined with Gaussian blur (A3LIS-AT-GB), highlighting the positive impact of this preprocessing strategy for generalizing to unconstrained environments.

Table 4.14: Best Performing Models on A3LIS Test Set

Metric	Best Model	Training Dataset
Top-1 Accuracy (%)	v0 ND	A3LIS
Top-2 Accuracy (%)	v1 30%D	A3LIS
Top-3 Accuracy (%)	v0 ND, v1 30%D | v0 30%D	A3LIS | A3LIS-GB
ATP (%)	v1 ND	A3LIS-LB-GB

Table 4.15: Best Performing Models on Real-World Test Set

Metric	Best Model	Training Dataset
Top-1 Accuracy (%)	v1 ND	A3LIS-AT-GB
Top-2 Accuracy (%)	v1 ND	A3LIS-AT-GB
Top-3 Accuracy (%)	v1 ND	A3LIS-AT-GB
ATP (%)	v1 30%D	A3LIS-LB-GB

Overall, models trained with preprocessing techniques involving background modification and/or Gaussian blur generally achieved better generalization performance on the real-world dataset compared to models trained on the unaltered green background dataset. Notably, the models trained on the adaptive thresholding variant (A3LIS-AT-GB) showed a significant advantage for real-world testing, achieving the highest Top-1, Top-2, and Top-3 accuracies despite demonstrating lower performance on the A3LIS controlled test set. These findings reinforce the importance of preprocessing choices to bridge the gap between controlled datasets and real-world applications.

To provide further insights into model behavior, confusion matrices for the best-performing models on each test set are presented in Figures 4.1, 4.2, 4.3, and 4.4. These matrices illustrate the distribution of predicted versus true classes, highlighting common misclassifications and offering a detailed view of the models' prediction patterns.

Figures 4.1 and 4.2 show the results on the A3LIS test set for, respectively, model v0 ND trained on A3LIS and model v1 ND trained on A3LIS-AT-GB. Figures 4.3 and 4.4 display the corresponding results on the real-world test set.

Figure 4.1 highlights the strong performance of model v0 ND on the controlled A3LIS test set. However, the same model struggles significantly when evaluated on the real-world dataset (Figure 4.3), often misclassifying most inputs as either the *interprete* or idle classes. In contrast, model v1 ND trained on the adaptive thresholded and blurred dataset (A3LIS-AT-GB) exhibits a more consistent and

balanced performance across both controlled and real-world conditions, as shown in Figures 4.2 and 4.4.

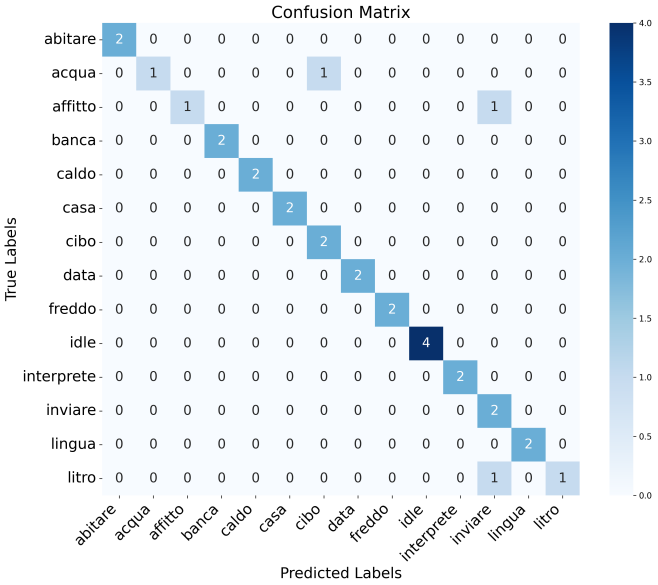

Figure 4.1: Confusion matrix for A3LIS test set predictions of model v0 ND trained on A3LIS.

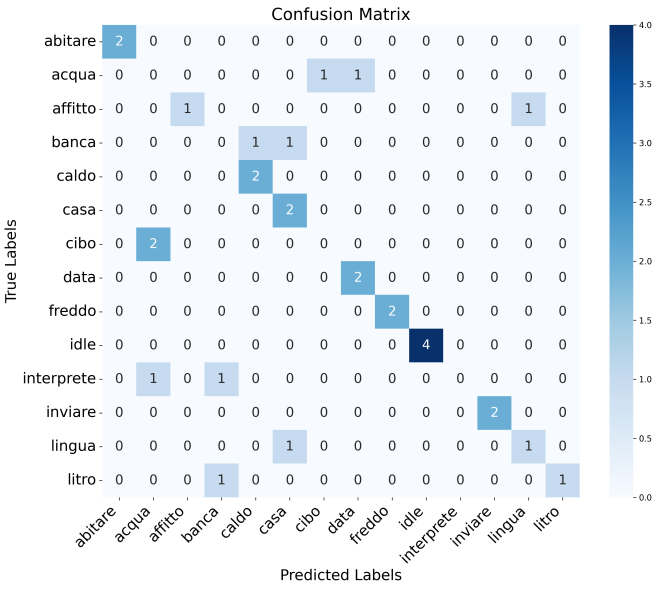

Figure 4.2: Confusion matrix for A3LIS test set predictions of model v1 ND trained on A3LIS-AT-GB.

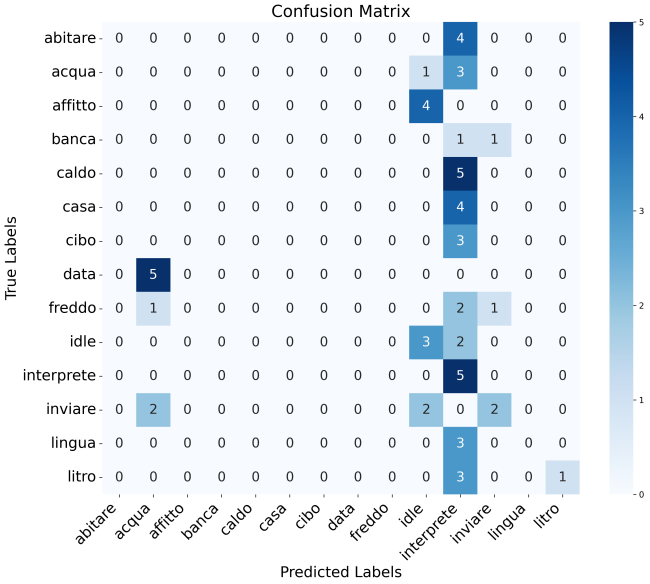

Figure 4.3: Confusion matrix for real-world test set predictions of model v0 ND trained on A3LIS.

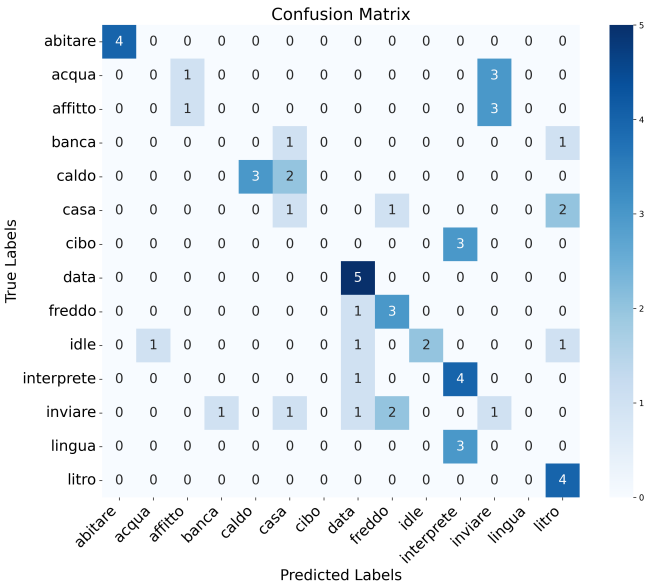

Figure 4.4: Confusion matrix for real-world test set predictions of model v1 ND trained on A3LIS-AT-GB.

Chapter 5

Discussion

5.1 Overview

This chapter discusses and interprets the experimental results presented in Chapter 4. Beyond reporting numerical outcomes, the analysis focuses on identifying underlying trends, evaluating the impact of different preprocessing techniques, and assessing the overall effectiveness of the proposed methodology.

The discussion also highlights the strengths and limitations of the developed system, emphasizing both the successes achieved and the challenges encountered. In addition, suggestions for future research directions are provided to guide further development and potential improvements.

The goal of this chapter is to critically assess the work carried out, placing the results into context and exploring how different design decisions influenced the system's performance in both controlled and real-world scenarios.

5.2 Interpretation of Results

The experiments conducted in this work revealed distinct patterns regarding the impact of preprocessing techniques, model architectures, and dropout regularization on isolated sign recognition performance.

5.2.1 Performance on A3LIS Test Set

On the controlled A3LIS test set, the best overall performance was achieved when training models using the original dataset without additional preprocessing. In particular, the model version v0 without dropout reached the highest Top-1 accuracy of 90% (Table 4.6). Models trained with only Gaussian blur (A3LIS-GB) or light background modification (A3LIS-LB) showed slight drops in performance compared

to the original dataset, although they remained competitive. In contrast, adaptive thresholding preprocessing (A3LIS-AT-GB) consistently led to lower accuracies on the A3LIS test set across all metrics, suggesting that important spatial information was lost when hand segmentation was overly simplified.

These results indicate that models trained directly on the raw A3LIS dataset may have overfitted to specific characteristics of the controlled acquisition environment. The green screen background, standardized lighting, and consistent clothing worn by signers provided strong visual cues that persisted across all samples, potentially biasing the models towards exploiting these non-essential features. Preprocessing techniques such as Gaussian blur, light background substitution, or adaptive thresholding acted to remove or attenuate some of these cues, thereby eliminating shortcuts the models might have learned. Consequently, while preprocessing slightly degraded performance on the highly constrained test set, it encouraged the models to focus more on essential hand structures and motion patterns, at the cost of reduced reliance on background and environment artifacts.

5.2.2 Performance on Real-World Test Set

In contrast to the controlled test set, models trained with preprocessing involving adaptive thresholding and Gaussian blur (A3LIS-AT-GB) demonstrated the best results on the real-world test set recorded by the author. Specifically, the model version v1 without dropout achieved the highest Top-1 accuracy of 48.28% and a Top-3 accuracy of 75.86% (Tables 4.7 and 4.11), which represent strong results for real-world evaluation. This highlights that adaptive thresholding, while detrimental to performance on highly controlled backgrounds, improved the system's robustness under real-world conditions, where lighting, background clutter, and visual noise differed significantly from the original training environment.

The application of preprocessing techniques, particularly adaptive thresholding, appears to have forced the models to generalize beyond superficial background and environmental features. By simplifying the hand region to its most salient contours and discarding background information, adaptive thresholding encouraged the models to focus primarily on hand shape, motion, and relative positioning, which are critical features for sign recognition. As a result, even though models trained with adaptive thresholding underperformed on the A3LIS dataset, they proved more resilient when facing the natural variability present in real-world scenarios.

Moreover, models incorporating Gaussian blur or light background adjustments also showed improved generalization to real-world data compared to models trained purely on the original green background. These techniques introduced mild distortions or background variations during training, reducing the models' dependency on constrained conditions and making them more tolerant to imperfect and noisy inputs at inference time.

Further insights can be drawn from the confusion matrices presented in Figures 4.1 and 4.3. The model version v0 without dropout trained on A3LIS with no preprocessing, which performed best on the A3LIS test set, struggled significantly when tested on the real-world dataset. As shown in Figure 4.3, this model tended to predict most inputs as either the *interprete* or idle classes. A plausible explanation is that, during training, these signs were performed with the hands positioned closer to the body, partially covering the green background and reducing background variability in the HEIs. Consequently, the model might have learned to associate certain spatial patterns, less influenced by the background, with these classes. When exposed to real-world HEIs with different background characteristics, the model defaulted to these familiar and safer predictions, reflecting its overdependence on constrained training aspects rather than robust hand gesture features.

Overall, the findings suggest that preprocessing strategies that intentionally remove or reduce dataset-specific artifacts can play a pivotal role in improving model transferability from laboratory environments to real-world applications, even at the expense of some degradation in performance under controlled settings.

5.2.3 Model Architectures and Dropout Effects

Comparing model versions, the simpler model v0 often performed better on the controlled A3LIS test set, while the deeper model v1, based on separable convolutions and a larger representational capacity, achieved competitive or superior results on the real-world dataset when combined with appropriate preprocessing techniques. This suggests that while simpler architectures may excel in clean, standardized environments, more complex models are advantageous when the input data presents greater variability.

Regarding dropout, its impact was particularly evident when evaluating real-world performance. Models trained without dropout tended to overfit more severely to the constrained conditions of the A3LIS dataset, as evidenced by their sharp drop in accuracy when tested on the real-world dataset (Table 4.7). In contrast, adding a dropout layer improved robustness, especially for models trained on datasets without extensive preprocessing (e.g., A3LIS and A3LIS-GB). In these cases, dropout acted as an effective regularizer, mitigating overfitting to background color, signer clothing, and other controlled environment artifacts.

However, it is important to note that the combination of heavy preprocessing (such as adaptive thresholding) and dropout did not always lead to further improvements. Since preprocessing techniques like adaptive thresholding already promote generalization by simplifying the input data and removing environment-specific biases, the additional application of dropout sometimes reduced performance. This suggests that, in cases where the input is already highly generalized, dropout may

remove useful discriminative features that are still necessary for reliable classification. Therefore, the benefit of dropout appears to be more pronounced for models trained on minimally preprocessed or raw datasets, while its effect becomes less clear, or even slightly detrimental, when strong preprocessing is already applied.

These results corroborate the broader observation that models trained on constrained datasets without regularization tend to capture false correlations, and that dropout, even in small lightweight architectures, plays a crucial role in encouraging better feature abstraction and generalization, especially when raw or lightly preprocessed data is used.

5.2.4 Average True Probability Trends

The Average True Probability (ATP) metric, which captures the model's confidence in its correct predictions, largely followed the patterns observed in Top-N accuracies across different datasets and preprocessing strategies. On the controlled A3LIS test set, models trained without heavy preprocessing typically achieved higher ATP values, indicating strong confidence when operating in a familiar, constrained environment.

However, when evaluated on the real-world dataset, ATP values dropped substantially for models trained solely on raw A3LIS data, reflecting their limited ability to maintain high confidence under unseen and variable conditions. In contrast, models trained on datasets incorporating light background modifications (A3LIS-LB), Gaussian blur (A3LIS-LB-GB), and particularly adaptive thresholding combined with Gaussian blur (A3LIS-AT-GB) demonstrated higher ATP scores on real-world data. This improvement suggests that preprocessing strategies that simplify the input distribution, by reducing background variability and emphasizing hand shapes, help the models produce more confident and reliable predictions outside the original training domain.

Moreover, ATP trends reinforced the earlier observations about the nuanced role of dropout. Models with dropout trained on lightly or non-preprocessed datasets (e.g., A3LIS) displayed noticeable ATP improvements on real-world evaluations, further supporting the idea that dropout effectively counters overfitting to specific backgrounds or signer artifacts. Conversely, in models trained on heavily preprocessed datasets, the additional regularization imposed by dropout sometimes led to minor reductions in ATP, likely because the available discriminative information had already been compressed through preprocessing.

Overall, the ATP results strengthen the conclusion that targeted preprocessing enhances model robustness and that the appropriate balance between data simplification and architectural regularization is critical for achieving strong, confident performance in real-world sign recognition tasks.

5.2.5 Summary of Interpretation of Results

In summary, the experimental findings highlight the critical interplay between preprocessing choices, model architecture, and regularization strategies in isolated sign language recognition. While raw data training preserved maximum information for constrained environments, preprocessing techniques such as adaptive thresholding and Gaussian blur proved essential for real-world generalization. Dropout played a complementary role, especially for models trained on less processed data, improving robustness by mitigating overfitting to environment-specific artifacts. These insights set the stage for a broader reflection on the strengths and contributions of the proposed system, discussed next.

5.3 Strengths of the Proposed Approach

The methodology developed in this work demonstrated several strengths, particularly considering the challenges posed by isolated sign language recognition under constrained data conditions.

5.3.1 Lightweight and Efficient Architecture

A major strength of the proposed system is its reliance on simple, lightweight Convolutional Neural Networks (CNNs) without the need for pretraining on large external datasets. Both model versions (v0 and v1) were specifically designed to balance computational efficiency and classification performance, making them highly suitable for real-world deployment scenarios, including potential use on mobile or embedded devices with limited resources.

5.3.2 Effective Utilization of HEI Representations

The Hand Energy Image (HEI) strategy proved to be an effective way to condense temporal motion and spatial information from video sequences into a compact, static input format. By averaging the sequence of segmented hand frames into a single image, the system was able to capture meaningful gesture dynamics without requiring complex sequential modeling. This greatly simplified the learning task while maintaining competitive classification results.

5.3.3 Robustness to Real-World Variability

Despite being trained on a relatively small and controlled dataset, the system exhibited encouraging generalization capabilities when applied to real-world videos. Preprocessing strategies such as adaptive thresholding and Gaussian blur played a

critical role in this achievement, enabling the models to better handle variations in background, lighting, signer clothing, and camera quality. Notably, models trained with these techniques outperformed baseline models in real-world conditions, demonstrating the practical adaptability of the approach.

5.3.4 Independence from Large Datasets

Unlike many modern deep learning approaches that depend heavily on massive annotated datasets, the proposed system was able to achieve reasonable performance using only 148 training videos. This emphasizes the system's efficiency and its potential for deployment in contexts where large-scale labeled datasets for sign language are not readily available—a common limitation in minority languages or resource-constrained settings.

5.3.5 Flexible and Modular Design

The training pipeline, including independent hand models and a combination strategy with tunable hyperparameters (e.g., *extra_weight2* and *top_for_extra*), offers flexibility for further improvements. The modular nature of the system allows easy substitution of preprocessing techniques, hand tracking modules, or classification backbones, making it a good foundation for future research or real-world adaptations.

5.4 Challenges

While the proposed approach demonstrated several promising strengths, some important limitations and challenges were encountered throughout the project. A critical evaluation of these aspects is essential for understanding the scope of the results and identifying areas for future improvement.

5.4.1 Limited Dataset Size

One of the main limitations of this work was the relatively small size of the training dataset. The A3LIS subset used for training comprised only 148 videos across 14 sign classes (including the idle class). Although the system achieved reasonable performance under these conditions, the limited amount of data likely constrained the model's ability to learn highly robust and generalized representations, particularly for signs with subtle inter-class differences.

5.4.2 Environment Constraints in Original Dataset

The A3LIS videos were recorded under highly standardized conditions: green screen background, controlled lighting, and consistent signer positioning. While beneficial for initial experiments, these constraints introduced a bias that made the models sensitive to environmental features not representative of real-world signing contexts. As observed, models trained solely on this data often overfitted to the controlled background and lighting artifacts.

5.4.3 Quality of Real-World Evaluation Data

Although the real-world dataset recorded by the author served as an important benchmark for generalization, it presents its own limitations. The videos were recorded by a single individual who is not a native LIS signer. Despite efforts to replicate the signs accurately, subtle inaccuracies in sign execution may have influenced model performance and do not fully capture the variability expected from a broader signer population.

5.4.4 Tracking and Preprocessing Challenges

Hand tracking errors, including missed detections, poor bounding box estimations, or occlusions, occasionally affected the quality of the generated HEIs. In some cases, tracking failures were significant enough that no usable HEIs could be extracted from particular video samples, as observed for two instances of the sign *banca* and one instance of the sign *freddo* in the real-world dataset.

Although preprocessing methods like adaptive thresholding improved generalization, they sometimes oversimplified the hand representations, potentially discarding subtle but discriminative features necessary for fine-grained classification.

5.4.5 Scope of Recognition: Isolated Signs Only

The current system was designed and evaluated exclusively for isolated sign recognition, meaning that each video contained a single, well-defined sign. In real-world scenarios, continuous signing, including transitions, coarticulations, and sentence-level structures, presents a much greater challenge. The methodology developed here does not yet address sequential modeling, temporal segmentation, or the recognition of continuous sign streams.

5.4.6 Computational Constraints

All training and testing were conducted without GPU acceleration, relying solely on CPU computation. While this setup reinforces the lightweight nature of the proposed approach, it also imposed practical constraints on experimentation speed and the possibility of testing more complex architectures or larger-scale hyperparameter tuning strategies.

5.5 Future Work

While the results obtained in this work demonstrate the potential of Hand Energy Images (HEIs) combined with lightweight CNN architectures for isolated sign recognition, several paths for future improvements and extensions have been identified.

5.5.1 Increasing Dataset Size and Diversity

One of the primary limitations of the current study is the restricted dataset size, particularly regarding the diversity of training and real-world evaluation data. Expanding the dataset with more native LIS signers, a wider range of sign variations, and different environmental conditions (such as varied backgrounds, lighting setups, and camera angles) would be crucial to enhancing the system's robustness and generalization capability. Collecting data from multiple individuals with diverse signing styles would not only improve the model's ability to generalize across different users but also help mitigate signer-dependent biases. A more varied and representative dataset would support both the training of more resilient models and a more rigorous evaluation of their performance in realistic usage scenarios.

5.5.2 Improving Hand Tracking Robustness

Although the hand-tracking solution used in this work was generally effective, occasional failures in detecting or accurately tracking hands, due to occlusions, non-standard poses, or subtle hand movements, highlight the need for even more robust solutions. Future work could focus on integrating more advanced or adaptive hand-tracking algorithms that better handle occlusions, hand overlaps, or challenging lighting conditions. Improved tracking stability would directly enhance the quality of the generated Hand Energy Images (HEIs), resulting in better input representations for recognition models.

5.5.3 Exploring Sequence Models for Continuous Sign Recognition

The current system addresses isolated sign recognition without considering the temporal dynamics between consecutive signs. Future research could extend the approach to continuous sign language recognition by incorporating sequence modeling techniques, such as recurrent neural networks (RNNs), Long Short-Term Memory networks (LSTMs), or attention-based temporal models. Capturing inter-sign transitions and context could significantly increase the system's applicability for real-world communication, where signs are performed naturally in sequences rather than in isolation.

5.5.4 Testing Alternative Architectures

While lightweight CNN architectures proved effective in this work, exploring alternative model designs could further improve performance and efficiency. Attention-based CNNs, lightweight transformer models, or architectures optimized for mobile deployment (such as MobileNet variants) represent promising directions. These approaches could provide a better balance between computational cost and accuracy, especially for scenarios requiring real-time performance on resource-constrained devices.

5.5.5 Real-Time Mobile Implementation

Given the compact nature of the proposed models and the low computational requirements of the HEI representation, a future goal is to implement a real-time sign recognition system on mobile or embedded platforms. This would involve optimizing the full pipeline, from hand tracking to inference, to ensure low-latency processing, possibly leveraging hardware accelerators like GPUs or NPUs available on modern smartphones. A real-time system would significantly enhance the practicality and accessibility of sign language recognition technology for daily use.

5.6 Discussion Summary

This chapter provided a critical interpretation of the experimental results, highlighting the role of different preprocessing strategies, model architectures, and regularization techniques in shaping system performance. The strengths and limitations of the proposed methodology were discussed, along with potential avenues for future research. The next chapter presents the final conclusions of this work, summarizing the main contributions and outlining broader implications for sign language recognition.

Chapter 6

Conclusion

6.1 Summary of Contributions

This thesis proposed a lightweight and effective system for isolated Italian Sign Language (LIS) recognition based on Hand Energy Images (HEIs) and custom convolutional neural networks (CNNs). By systematically evaluating different preprocessing strategies, such as Gaussian blur, light background modification, and adaptive thresholding, the study demonstrated how preprocessing impacts both performance on controlled datasets and generalization to real-world scenarios.

The system was built with simplicity and computational efficiency in mind, avoiding the need for large pre-trained models or extensive data augmentation. Despite the limited size of the available datasets, the approach achieved promising results, particularly when preprocessing methods that emphasized the hand's structural features were applied.

Furthermore, the thesis introduced a combined model approach, merging independent left- and right-hand classifiers to improve recognition performance, especially in more complex signs involving both hands.

6.2 Key Findings

The experiments conducted revealed several important trends:

- Models trained on the original A3LIS dataset without preprocessing achieved the highest accuracies on the controlled A3LIS test set, benefiting from dataset-specific biases like uniform backgrounds and standardized signing conditions.

- Preprocessing techniques that simplified the hand region, especially adaptive thresholding combined with Gaussian blur, significantly improved model

generalization to real-world test conditions, where variability in lighting, backgrounds, and signer styles is prominent.

- Applying dropout regularization improved real-world performance, especially for models trained on less preprocessed, more constrained datasets, by mitigating overfitting to background artifacts.
- Hand Energy Images (HEIs) proved to be a compact and effective representation for isolated sign recognition, particularly when combined with lightweight CNN architectures.

6.3 Challenges

Despite the promising results, several challenges were identified:

- The dataset used for training remained relatively small, limiting the system's exposure to signer variability and environmental diversity.
- The real-world test set, although helpful in simulating more practical conditions, was still limited in size and signer representation.
- Hand tracking errors occasionally affected the quality of HEIs, especially in cases of occlusions or rapid hand movements.
- The study focused exclusively on isolated sign recognition without modeling temporal dynamics or continuous signing sequences.

6.4 Future Directions

Building on the findings of this work, several avenues for future research are recommended:

- Expanding the dataset to include more signers, environments, and spontaneous signing to improve model robustness.
- Enhancing the hand tracking pipeline to better handle occlusions and partial hand visibility.
- Exploring sequential models, such as Recurrent Neural Networks (RNNs) or Transformers, to address continuous sign language recognition.
- Investigating lightweight attention-based architectures that could offer better feature extraction while maintaining efficiency.
- Pursuing real-time deployment strategies on mobile or embedded devices to make the system accessible in everyday assistive applications.

6.5 Final Remarks

This thesis demonstrated that even with limited resources and without relying on massive pre-trained models, it is possible to design an effective isolated sign language recognition system by carefully crafting preprocessing pipelines and model architectures. The results encourage further exploration into lightweight, real-world deployable solutions for sign language recognition, aiming to promote more inclusive technologies for the Deaf and hard-of-hearing communities.

Bibliography

[1] *World report on hearing*. Tech. rep. Geneva: World Health Organization, 2021 (cit. on p. 1).

[2] Wyatte C. Hall. «What You Don't Know Can Hurt You: The Risk of Language Deprivation by Impairing Sign Language Development in Deaf Children». In: *Maternal and Child Health Journal* 21 (5 May 2017), pp. 961–965. ISSN: 15736628. DOI: 10.1007/s10995-017-2287-y (cit. on pp. 1, 2).

[3] Laura Ann Petitto and Paula F. Marentette. «Babbling in the Manual Mode: Evidence for the Ontogeny of Language». In: *Science* 251.5000 (1991), pp. 1493–1496. DOI: 10.1126/science.2006424. eprint: https://www.science.org/doi/pdf/10.1126/science.2006424. URL: https://www.science.org/doi/abs/10.1126/science.2006424 (cit. on p. 2).

[4] Robert J. Ruben. «Sign language: Its history and contribution to the understanding of the biological nature of language». In: *Acta Oto-Laryngologica*. Vol. 125. 2005, pp. 464–467. DOI: 10.1080/00016480510026287 (cit. on p. 2).

[5] Marie Alaghband, Hamid Reza Maghroor, and Ivan Garibay. «A survey on sign language literature». In: *Machine Learning with Applications* 14 (Dec. 2023), p. 100504. ISSN: 26668270. DOI: 10.1016/j.mlwa.2023.100504 (cit. on pp. 2, 3).

[6] National Deaf Children's Society. *What is sign language?* URL: https://www.ndcs.org.uk/information-and-support/language-and-communication/sign-language/what-is-sign-language// (visited on 2024-08-28) (cit. on p. 2).

[7] AI-Media. *Sign Language Alphabets From Around The World*. URL: https://www.ai-media.tv/knowledge-hub/insights/sign-language-alphabets/ (visited on 2024-08-27) (cit. on p. 2).

[8] National Institute on Deafness and Other Communication Disorders (NIDCD). *American Sign Language*. URL: https://www.nidcd.nih.gov/health/american-sign-language (visited on 2024-08-28) (cit. on p. 2).

[9] Wpclipart. *American Sign Language Alphabet*. URL: https://www.wpclipart.com/sign_language/American_Sign_Language_Alphabet.png.html (visited on 2024-08-28) (cit. on p. 3).

[10] Terptree. *British Sign Language Alphabet*. URL: https://terptree.co.uk/british-sign-language-alphabet/ (visited on 2024-08-28) (cit. on p. 3).

[11] Sylvie C.W. Ong and Surendra Ranganath. «Automatic sign language analysis: A survey and the future beyond lexical meaning». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (6 June 2005), pp. 873–891. ISSN: 01628828. DOI: 10.1109/TPAMI.2005.112 (cit. on pp. 3, 4).

[12] Hee-Deok Yang and Seong-Whan Lee2. *Combination of Manual and Non-Manual Features for Sign Language Recognition Based on Conditional Random Field and Active Appearance Model*. 2011 (cit. on pp. 3, 4).

[13] Philip Vickerman and Milly Blundell. «Hearing the voices of disabled students in higher education». In: *Disability and Society* 25 (1 Jan. 2010), pp. 21–32. ISSN: 09687599. DOI: 10.1080/09687590903363290 (cit. on p. 4).

[14] Andrea MacLeod, Julie Allan, Ann Lewis, and Christopher Robertson. «'Here I come again': the cost of success for higher education students diagnosed with autism». In: *International Journal of Inclusive Education* 22 (6 June 2018), pp. 683–697. ISSN: 14645173. DOI: 10.1080/13603116.2017.1396502 (cit. on p. 4).

[15] Gillian Hendry, Alison Hendry, Henri Ige, and Natalie McGrath. «"I was isolated and this was difficult": Investigating the communication barriers to inclusive further/higher education for deaf Scottish students». In: *Deafness and Education International* 23 (4 2021), pp. 295–312. ISSN: 1557069X. DOI: 10.1080/14643154.2020.1818044 (cit. on p. 4).

[16] Marion Grimes Mary Brennan and Ernst D Thoutenhoofd. *Deaf students in Scottish higher education*. Scottish Funding Council, 2006 (cit. on p. 4).

[17] Patrick Stefan Kermit and Sidsel Holiman. «Inclusion in norwegian higher education: Deaf students' experiences with lecturers». In: *Social Inclusion* 6 (4Students with Disabilities in Higher Education Dec. 2018), pp. 158–167. ISSN: 21832803. DOI: 10.17645/si.v6i4.1656 (cit. on p. 4).

[18] S Foster, G Long, and K Snell. «Inclusive instruction and learning for deaf students in postsecondary education.» In: *The Journal of Deaf Studies and Deaf Education* 4.3 (July 1999), pp. 225–235. ISSN: 1081-4159. DOI: 10.1093/deafed/4.3.225. URL: https://doi.org/10.1093/deafed/4.3.225 (cit. on p. 4).

[19] V. Tartter and K. Knowlton. «Perception of sign language from an array of 27 moving spots». In: *Nature* 289 (1981), pp. 676–678. DOI: https://doi.org/10.1038/289676a0 (cit. on p. 6).

[20] T. Starner and A. Pentland. «Real-time American Sign Language recognition from video using hidden Markov models». In: *Proceedings of International Symposium on Computer Vision - ISCV*. 1995, pp. 265–270. DOI: 10.1109/ISCV.1995.477012 (cit. on pp. 6, 7).

[21] Rung-Huei Liang and Ming Ouhyoung. «A sign language recognition system using hidden markov model and context sensitive search». In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. VRST '96. Hong Kong: Association for Computing Machinery, 1996, pp. 59–66. ISBN: 0897918258. DOI: 10.1145/3304181.3304194. URL: https://doi.org/10.1145/3304181.3304194 (cit. on p. 6).

[22] K. Grobel and M. Assan. «Isolated sign language recognition using hidden Markov models». In: *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*. Vol. 1. 1997, 162–167 vol.1. DOI: 10.1109/ICSMC.1997.625742 (cit. on p. 6).

[23] Marcell Assan and Kirsti Grobel. «Video-based sign language recognition using Hidden Markov Models». In: *Gesture and Sign Language in Human-Computer Interaction*. Ed. by Ipke Wachsmuth and Martin Fröhlich. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 97–109. ISBN: 978-3-540-69782-4 (cit. on p. 6).

[24] Rung-Huei Liang and Ming Ouhyoung. «A real-time continuous gesture recognition system for sign language». In: *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*. 1998, pp. 558–567. DOI: 10.1109/AFGR.1998.671007 (cit. on p. 6).

[25] T. Starner, J. Weaver, and A. Pentland. «Real-time American sign language recognition using desk and wearable computer based video». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.12 (1998), pp. 1371–1375. DOI: 10.1109/34.735811 (cit. on pp. 6, 7).

[26] Jiyong Ma, Wen Gao, Jiangqin Wu, and Chunli Wang. «A continuous Chinese sign language recognition system». In: *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*. 2000, pp. 428–433. DOI: 10.1109/AFGR.2000.840670 (cit. on p. 6).

[27] B. Bauer, H. Hienz, and K.-F. Kraiss. «Video-based continuous sign language recognition using statistical methods». In: *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*. Vol. 2. 2000, 463–466 vol.2. DOI: 10.1109/ICPR.2000.906112 (cit. on pp. 6, 7).

[28] Lihong Zheng, Bin Liang, and Ailian Jiang. «Recent Advances of Deep Learning for Sign Language Recognition». In: *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. 2017, pp. 1–7. DOI: 10.1109/DICTA.2017.8227483 (cit. on p. 7).

[29] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. *Sign Language Recognition: A Deep Survey*. Feb. 2021. DOI: 10.1016/j.eswa.2020.113794 (cit. on p. 7).

[30] A.M. Martinez, R.B. Wilbur, R. Shay, and A.C. Kak. «Purdue RVL-SLLL ASL database for automatic recognition of American Sign Language». In: *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces*. 2002, pp. 167–172. DOI: 10.1109/ICMI.2002.1166987 (cit. on pp. 8, 9).

[31] Vassilis Athitsos, Carol Neidle, Stan Sclaroff, Joan Nash, Alexandra Stefan, Quan Yuan, and Ashwin Thangali. «The American Sign Language Lexicon Video Dataset». In: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 2008, pp. 1–8. DOI: 10.1109/CVPRW.2008.4563181 (cit. on pp. 8, 9).

[32] Philippe Dreuw, David Rybach, Thomas Deselaers, Morteza Zahedi, and Hermann Ney. «Speech Recognition Techniques for a Sign Language Recognition System». In: *Interspeech*. ISCA best student paper award Interspeech 2007. Antwerp, Belgium, Aug. 2007, pp. 2513–2516. URL: http://www.isca-speech.org/awards.html#student (cit. on pp. 8, 9).

[33] Philippe Dreuw, Carol Neidle, Vassilis Athitsos, Stan Sclaroff, and Hermann Ney. «Benchmark Databases for Video-Based Automatic Sign Language Recognition». In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Ed. by Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias. Marrakech, Morocco: European Language Resources Association (ELRA), May 2008. URL: http://www.lrec-conf.org/proceedings/lrec2008/pdf/287_paper.pdf (cit. on pp. 8, 9).

[34] Andre Barczak, Napoleon Reyes, M Abastillas, A Piccio, and Teo Susnjak. «A New 2D Static Hand Gesture Colour Image Dataset for ASL Gestures». In: *Res Lett Inf Math Sci* 15 (Jan. 2011) (cit. on pp. 8, 9).

[35] Samaa M. Shohieb, Hamdy K. Elminir, and A.M. Riad. «SignsWorld Atlas; a benchmark Arabic Sign Language database». In: *Journal of King Saud University - Computer and Information Sciences* 27.1 (2015), pp. 68–76. ISSN: 1319-1578. DOI: https://doi.org/10.1016/j.jksuci.2014.03.011. URL: https://www.sciencedirect.com/science/article/pii/S1319157814000548 (cit. on pp. 8, 9).

[36] Ghazanfar Latif, Nazeeruddin Mohammad, Jaafar Alghazo, Roaa AlKhalaf, and Rawan AlKhalaf. «ArASL: Arabic Alphabets Sign Language Dataset». In: *Data in Brief* 23 (2019), p. 103777. ISSN: 2352-3409. DOI: https://doi.org/10.1016/j.dib.2019.103777. URL: https://www.sciencedirect.com/science/article/pii/S2352340919301283 (cit. on pp. 8, 10).

[37] Andres Jessé Porfirio, Kelly Laís Wiggers, Luiz E.S. Oliveira, and Daniel Weingaertner. «LIBRAS Sign Language Hand Configuration Recognition Based on 3D Meshes». In: *2013 IEEE International Conference on Systems, Man, and Cybernetics*. 2013, pp. 1588–1593. DOI: 10.1109/SMC.2013.274 (cit. on pp. 8, 10).

[38] Adam Schembri, Jordan Fenlon, Ramas Rentelis, Sally Reynolds, and Kearsy Cormier. «Building the British Sign Language Corpus». In: 7 (2013), pp. 136–154. ISSN: 1934-5275. URL: http://nflrc.hawaii.edu/ldchttp://hdl.handle.net/10125/4592 (cit. on pp. 8, 10).

[39] Ulrich Von Agris and Karl-Friedrich Kraiss. «Towards a video corpus for signer-independent continuous sign language recognition». In: *Gesture in Human-Computer Interaction and Simulation, Lisbon, Portugal* 11 (2007) (cit. on pp. 8, 10).

[40] Jens Forster, Christoph Schmidt, Oscar Koller, Martin Bellgardt, and Hermann Ney. «Extensions of the Sign Language Recognition and Translation Corpus RWTH-PHOENIX-Weather». In: vol. 1. May 2014 (cit. on pp. 8, 10).

[41] Marco Fagiani, Stefano Squartini, Emanuele Principi, and Francesco Piazza. «A New Italian Sign Language Database». In: July 2012. ISBN: 978-3-642-31560-2. DOI: 10.1007/978-3-642-31561-9_18 (cit. on pp. 9, 10, 20).

[42] Richard E. Woods Rafael C. Gonzalez. *Digital Image Processing*. New York: Pearson Education, 2018 (cit. on p. 11).

[43] Nikhil Verma and Maitreyee Dutta. «Contrast Enhancement Techniques: A Brief and Concise Review». In: *International Research Journal of Engineering and Technology* (2017). ISSN: 2395-0072. URL: www.irjet.net (cit. on p. 11).

[44] Nungsanginla Longkumer A, Mukesh Kumar A, and Rohini Saxena A. *Contrast Enhancement Techniques using Histogram Equalization: A Survey*. Tech. rep. 2014. URL: http://inpressco.com/category/ijcet (cit. on p. 11).

[45] I A Adeyanju, O O Bello, and M A Adegboye. «Machine learning methods for sign language recognition: A critical review and analysis». In: *Intelligent Systems with Applications* 12 (2021), p. 56. DOI: 10.1016/j.iswa.2021.20. URL: https://doi.org/10.1016/j.iswa.2021.20 (cit. on pp. 11, 12).

[46] Debasish Biswas, Amitava Nag, Soumadip Ghosh, Arindrajit Pal, Sushanta Biswas, Snehasish Banerjee, and Anjan Pal. *NOVEL GRAY SCALE CONVERSION TECHNIQUES BASED ON PIXEL DEPTH*. Tech. rep. 2011. URL: www.jgrcs.info (cit. on p. 11).

[47] Stanley J. Reeves. «Image Restoration: Fundamentals of Image Restoration». In: 2014, pp. 165–192. DOI: 10.1016/b978-0-12-396501-1.00006-6 (cit. on p. 12).

[48] Marlon Oliveira, Alistair Sutherland, and Mohamed Farouk. «Two-stage PCA with interpolated data for hand shape recognition in sign language». In: *2016 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. 2016, pp. 1–4. DOI: 10.1109/AIPR.2016.8010587 (cit. on p. 12).

[49] Rajesh Kaluri and C. H. Pradeep Reddy. «An enhanced framework for sign gesture recognition using hidden markov model and adaptive histogram technique». In: *International Journal of Intelligent Engineering and Systems* 10 (3 June 2017), pp. 11–19. ISSN: 21853118. DOI: 10.22266/ijies2017.0630.02 (cit. on p. 12).

[50] M. Egmont-Petersen, D. de Ridder, and H. Handels. «Image processing with neural networks—a review». In: *Pattern Recognition* 35.10 (2002), pp. 2279–2301. ISSN: 0031-3203. DOI: https://doi.org/10.1016/S0031-3203(01)00178-9. URL: https://www.sciencedirect.com/science/article/pii/S0031320301001789 (cit. on p. 12).

[51] G. Ananth Rao and P. V. V. Kishore. «Selfie Video-Based Continuous Indian Sign Language Recognition System». In: *Ain Shams Engineering Journal* 9.4 (2018), pp. 1929–1939. ISSN: 2090-4479. DOI: 10.1016/j.asej.2016.10.013 (cit. on p. 12).

[52] Abhishek Dudhal, Heramb Mathkar, Abhishek Jain, Omkar Kadam, and Mahesh Shirole. «Hybrid SIFT feature extraction approach for Indian sign language recognition system based on CNN». In: vol. 30. Springer Netherlands, 2019, pp. 727–738. DOI: 10.1007/978-3-030-00665-5_72 (cit. on p. 12).

[53] Khamar Basha Shaik, P. Ganesan, V. Kalist, B.S. Sathish, and J. Merlin Mary Jenitha. «Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space». In: *Procedia Computer Science* 57 (2015). 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015), pp. 41–48. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.07.362. URL: https://www.sciencedirect.com/science/article/pii/S1877050915018918 (cit. on p. 13).

[54] Sudeep D. Thepade, Gandhali Kulkarni, Arati Narkhede, Priti Kelvekar, and Seema Tathe. «Sign language recognition using color means of gradient slope magnitude edge images». In: *2013 International Conference on Intelligent Systems and Signal Processing (ISSP)*. 2013, pp. 216–220. DOI: 10.1109/ISSP.2013.6526905 (cit. on p. 13).

[55] Yuancheng Ye, Yingli Tian, Matt Huenerfauth, and Jingya Liu. «Recognizing American Sign Language Gestures from Within Continuous Videos». In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2018, pp. 2145–214509. DOI: 10.1109/CVPRW.2018.00280 (cit. on p. 14).

[56] G. Anantha Rao, K. Syamala, P. V.V. Kishore, and A. S.C.S. Sastry. «Deep convolutional neural networks for sign language recognition». In: *2018 Conference on Signal Processing And Communication Engineering Systems, SPACES 2018*. Vol. 2018-January. Institute of Electrical and Electronics Engineers Inc., Mar. 2018, pp. 194–197. ISBN: 9781538623695. DOI: 10.1109/SPACES.2018.8316344 (cit. on p. 14).

[57] Kian Ming Lim, Alan Wee Chiat Tan, Chin Poo Lee, and Shing Chiang Tan. «Isolated sign language recognition using Convolutional Neural Network hand modelling and Hand Energy Image». In: *Multimedia Tools and Applications* 78 (14 July 2019), pp. 19917–19944. ISSN: 15737721. DOI: 10.1007/s11042-019-7263-7 (cit. on pp. 15, 27, 28).

[58] Pedro M. Ferreira, Jaime S. Cardoso, and Ana Rebelo. «On the role of multimodal learning in the recognition of sign language». In: *Multimedia Tools and Applications* 78 (8 Apr. 2019), pp. 10035–10056. ISSN: 15737721. DOI: 10.1007/s11042-018-6565-5 (cit. on p. 15).

[59] A. Wadhawan and P. Kumar. «Deep learning-based sign language recognition system for static signs». In: *Neural Computing and Applications* 32.20 (2020), pp. 7957–7968. DOI: 10.1007/s00521-019-04691-y. URL: https://doi.org/10.1007/s00521-019-04691-y (cit. on p. 16).

[60] Zifan Jiang, Gerard Sant Muniesa, Amit Moryossef, Mathias Müller, Rico Sennrich, and Sarah Ebling. «SignCLIP: Connecting Text and Sign Language by Contrastive Learning». In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 9171–9193. URL: https://doi.org/10.5167/uzh-264814 (cit. on p. 17).

[61] Camillo Lugaresi et al. «MediaPipe: A Framework for Building Perception Pipelines». In: (June 2019). URL: http://arxiv.org/abs/1906.08172 (cit. on p. 19).

[62] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. 4th. Pearson Prentice Hall, 2008 (cit. on p. 24).