Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/95/20241748

SignifAI: French sign language translation based on deep learning and large language models

Jiarui Dong
École Jeannine Manuel

jerry.dong246@gmail.com

Abstract. In this project, we propose SignifAI which is an AI-powered software that translates French Sign Language into spoken French. The goal of SignifAI is to remove the language barrier that separates the deaf and hearing communities. The translation process proves to be a major challenge for existing methods, as most of them rely on traditional Natural Language Processing (NLP) methods which solve the task in a similar way as spoken languages (e.g., paying more attention to the logics and reasoning). The drastic syntactical difference between sign and spoken languages, however, makes such approaches not as effective and efficient. In SignifAI, we propose to leverage a combination of deep learning models for sign gesture recognition and large language models (LLM) for sentence generation from recognized words. The former allows for precise detection of individual words, while the latter takes the recognized words as inputs and returns a fluently reformulated translated sentence. With extremely limited data available for French sign language, we created our own enhanced dataset through synthetically augmenting existing videos. Utilizing the SlowFast model for this task, we have achieved a maximum test accuracy of 98.9%, and a fluid translation process after implementing ChatGPT's API. The potential applications of SignifAI range from aiding communication between the deaf and the hearing, to helping the deaf place phone calls in numerical services. Ultimately, SignifAI strives to improve all aspects of a deaf person's life, and compensates for their speaking disabilities.

Keywords: French Sign Language (LSF), SlowFast, Words Recognization, Large Language Models (LLM), Sentence Generation

1. Introduction

1.1. Background

Three entire centuries after its invention in 1755 by the French cleric Charles-Michel de l'Epee, little has changed in the world of sign language [1]. But now, we have reached a major turning point in the face of a digitalized era. A stellar breakthrough is awaited that will fundamentally transform sign language and make it more accessible than ever, and it is only fitting to start this revolution from the ancestor of all sign languages, LSF (Langue des Signes Francais, or French sign language).

Today, 1.5 billion people, equivalent to 20% of the world population, possess some degree of hearing loss [2]. Of these, 430 million people have serious disabling hearing loss. In France alone, an estimated total of 5.4 million people experience such hearing loss, with over 360,000 categorized as profoundly deaf [3].

We must acknowledge that this is a massive faction of the population that we are addressing. The events of COVID only hastened this digital revolution, transferring utterly everything we do into digital

DOI: 10.54254/2755-2721/95/20241748

format. This might be beneficial for the majority of us, but for the deaf and hard of hearing, it deteriorates every aspect of their everyday lives. To get a doctor's appointment, for instance, they have to place a phone call, which can pose a major problem for someone who cannot speak. This population direly needs a dedicated tool with the goal of facilitating their communication, as a means to help them adapt to this numerical era.

1.2. Understanding Sign Languages

Sign languages are diverse: Instead of one universal sign language, over 300 dialects are widely in use across the globe [4], including ASL (American Sign Language), BSL (British Sign Language), CSL (Chinese Sign Language), and what matters most to us, LSF (French Sign Language).

Sign languages have their unique syntax and grammar: This means that traditional NLP approaches will not be effective in the translation process, as one cannot simply transliterate between sign and spoken languages. NLP methods [5] rely on identifying a logical correspondence between the two languages, but this does not always exist in our situation. Let's take a concrete example, with the sentence:

The girl in the school is drawing a cat.

In spoken English, we follow the general order of subject, verb, and object. However, in sign language, we reverse this order. We might sign the following:

In the school, cat, girl, draw

In sign language, the structure of a sentence often mirrors storytelling. Instead of following the typical Subject-Verb-Object (SVO) order, sign language tends to prioritize setting the scene first. The signer begins by establishing the time and place (answering the questions "when?" and "where?"), then the subject (answering "who?"), and finally evokes the action (answering "what?") . This approach creates a vivid mental image of the scene, where the setting is established before the action unfolds [6][7].

Sign language data is extremely sparse: The population for sign language users are not as large compared to spoken languages, making sign laguage data relatively sparse. This is especially true for LSF, which has decreased in popularity over the years.

1.3. Past contributions

Many contributions have been carried out around the world in the past decades in hopes of achieving sign language translation. Here we summarized them into several groups based on the methods they use.

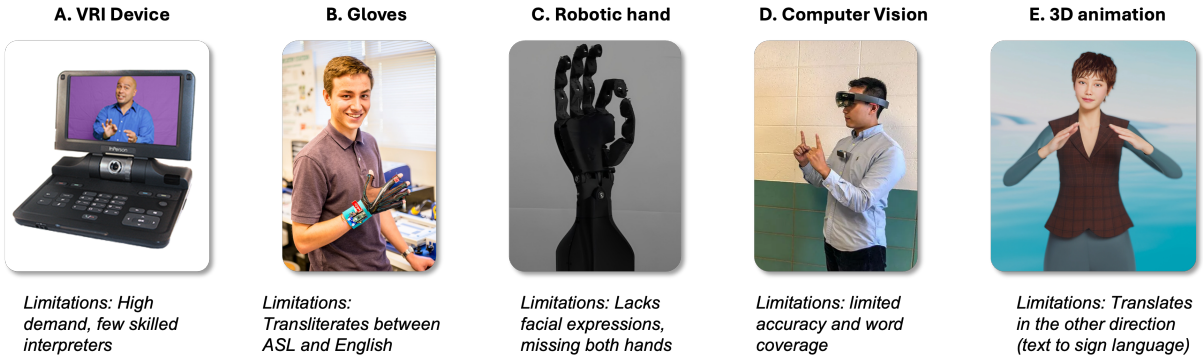

Figure 1: Illustrations and limitations of past contributions

Human interpreters: Over these past years, remote sign language interpreting services [8] (Video Remote Interpreting, VRI) and the presence of interpreters at public events have become significantly

DOI: 10.54254/2755-2721/95/20241748

more common (as shown in A.Fig.1). This facilitates the overall accessibility of interpreters, as there is no longer the need for them to be physically present in the room at all times. The technology is already widely in use in hospitals and public facilities. Nonetheless, however, there is still a lack of skilled sign language interpreters to meet the growing demand.

Gloves: In 2016, two Lemelson-MIT award winners, Navid Azodi and Thomas Pryor, designed a pair of gloves capable of translating ASL into English in real time (as shown in B.Fig.1). Their project, SignAloud, used precise sensors to detect the subtle movements of the human hand, and relayed all the gathered information via Bluetooth to an external computer for real time processing and translation [9]. Having won multiple awards for this engineering feat, the gloves saw worldwide recognition. The software backbone of these gloves relies on word to word translation, as the computer iterates over each individual gesture to find a match. This may pose a problem, as ASL and English differ drastically in their syntax and grammar. The BrightSign project [10][11], based in the United Arab Emirates, strives to achieve a similar result, but is still subject to the same underlying issues pointed above.

Robotics: In the realm of robotics, there have been multiple attempts at creating finger spelling robots. This includes the TATUM robot developed by Samantha Johnson at Northeastern University [12] (as shown in C.Fig.1), as well as ASLAN (Antwerp's Sign Language Actuating Node) [13]. For instance, TATUM consists of a single robotic hand which takes text as input and acts out the corresponding sign. The advantages of having a physical robotic hand is that it impacts the blind community as well, as many blind individuals rely on feeling the movement of the hand and fingers. However, with TATUM consisting of a single robotic hand in its early stages, it omits a large part of the essence behind sign language, such as an individual's facial expressions or gestures that involve the whole body.

Computer Vision and Deep Learning: With large amounts of ASL datasets easily accessible on Kaggle and everywhere else on the web [14], there are numerous attempts at creating an AI-powered translator that takes video frames as input, and processes them in real time. Efforts have been conducted to varying extents, with the majority of projects still in their early development phase. Examples include the SignAll initiative [15] that captures and translates ASL, as well DeepASL [16] from Michigan State University(as shown in D.Fig.1). Areas of improvement mainly include enhancing the accuracy of existing translators, as well as expanding the word dictionary to encompass more words.

3-dimensional animation: Lastly, 3D animation is an innovative and promising way of translating text into sign. Recent advancements include the Japanese virtual robot KIKI [17] (as shown in E.Fig.1), with a realistic humanoid appearance to closely replicate human signing, and the HandTalk app [18] which translates text into ASL via a cartoon animated character. However, this meta-human approach only deals with text to sign translation, which is arguably the easier direction of translation, and is unable to perform the translation vice versa.

1.4. The remaining challenges

With all of the current efforts in tackling this endeavor now being listed, it is time to address the remaining challenges. These include:

Improving sign language detection accuracy: Current methods may still detect certain gestures inaccurately, especially when the word dictionary expands. To improve detection accuracy, we will need large quantities of varied, high quality LSF data.

Improving sign language translation accuracy: Current methods rely on identifying a logical correspondence between sign and spoken languages, using NLP methods. This is effectively a weak approach to the translation process, as one cannot simply transliterate between the two languages.

Lack of LSF support: All of the existing solutions deal exclusively with ASL, hardware and software approaches alike. French Sign Language support, on the other hand, is extremely limited. This scarcity of resources can be seen at play when we scour for data online, as very few results appear. French Sign language really is a loophole in this otherwise packed technological era, with almost no previous attempts in addressing this demand.

DOI: 10.54254/2755-2721/95/20241748

1.5. Our contributions

To address these remaining challenges, the paper's main contributions are:

1. **Created a synthetically enhanced LSF dataset.** Given the current shortage of LSF data online, we created our own dataset from synthetically enhancing existing videos. By doing so, we were able to create our training and testing datasets.
2. **Leveraged the power of LLMs to improve the translation accuracy.** As opposed to NLP approaches which are based on logical reasoning, we strived to integrate ChatGPT's API into the translation process. It can help bridge the gap between the different grammatical systems, analyze the speaker's intent, and predict subsequent words that are likely to be said after.
3. **Developed a French Sign Language translation software using a finetuned SlowFast model.** Our system will take on the form of an app to be installed on a mobile device. Doing so ensures that our translator is versatile, and widely accessible to use. It will be able to access the device's front camera for live video processing, powered by the backbone of the Slowfast model. No extra hardware is involved in this translation process, ensuring absolute simplicity.

2. The proposed method

The overall workflow of our proposed method is shown in Figures 2-3. We split the entire process into two parts, the development phase (Fig.2) and the deployment phase (Fig.3). The former focuses more on the back-end development of the AI technology, whilst the latter focuses on integrating the model into a user-friendly app with an easy-to-use UI.

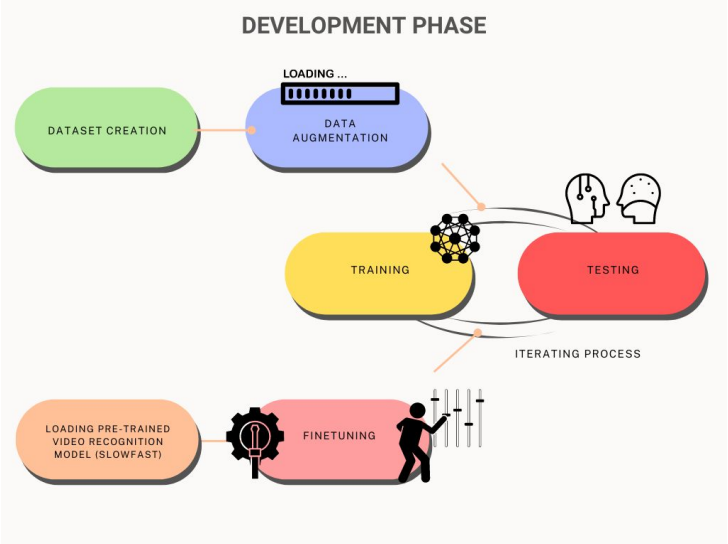

Figure 2: The Development phase

DOI: 10.54254/2755-2721/95/20241748

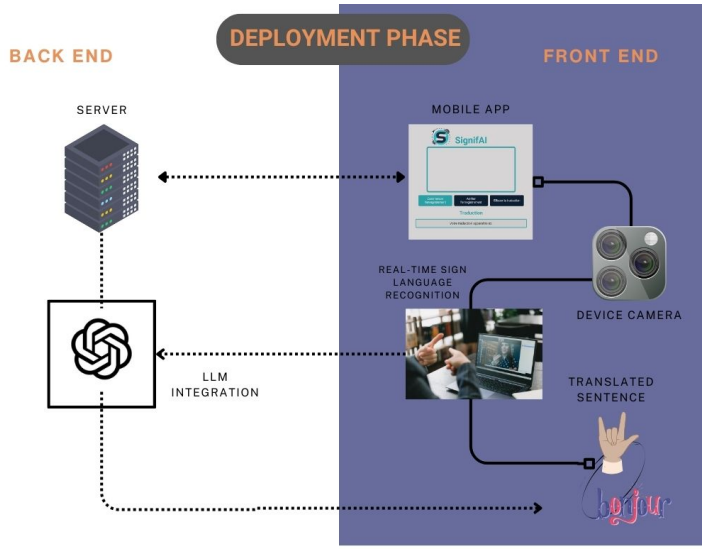

Figure 3: The Deployment phase

2.1. Development phase

In this phase (as shown in Fig.2), we created an initial dataset that we then synthetically augmented. We simultaneously loaded a pre-trained SlowFast model, that we then finetuned according to our use case. This is followed by the training-testing phase.

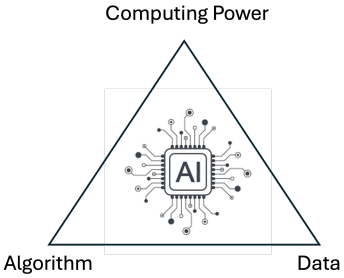

Figure 4: The three primary pillars of Artificial Intelligence

When it comes to the realm of AI, there are three main pillars that decide it all: computing power, algorithm, and data (as shown in Fig.4). With the computation limited by mobile device usage capacity and the algorithm limited by a strict requirement of GPU resources, data becomes very important for optimizing the model's performance. In this section, we will describe how we collect data for finetuning a Slowfast model capable of LSF word recognition.

2.1.1. Dataset creation As established previously, online resources related to LSF are extremely scarce. Nonetheless, we started with the LSF-data dataset [19] which included 1092 words (Fig.5). For each word, the dataset includes a video and its corresponding label.

We also remarked how our initial dataset lacked an abundance of common words in spoken French, including the likes of "parler" (to speak) or "savoir" (to know). As such, through the use of an online LSF dictionary [20], we manually downloaded an additional 153 videos and included their label to expand our total French coverage.

DOI: 10.54254/2755-2721/95/20241748

Figure 5: Example video from our LSF dataset - The signer is performing the word ”bonne journée” (good day) in LSF

Figure 6: Data augmentation flowchart

2.1.2. Data augmentation With only one copy of a video for each word, we were unable to perform a train test split on our existing dataset: doing so would result in missing class indices within the training and testing datasets.

In order to augment the data, we randomly applied a set of transformations onto an existing video and generated new ones. Our transformations included resizing (0.4x, 0.6x, 0.8x), rotating (90, 180, and 270 degrees), adjusting the brightness (0.5x, 1.5x, 2x), and grayscaling the video frames. The result of the transformations can been seen in Fig.7. The number and choice of transformations to apply were completely random. This innovative method ultimately allowed us to increase our dataset size by well over tenfold, going from 1092 to 14490 videos, with 1245 labels in the end. The entire data augmentation process is shown in Fig.6.

DOI: 10.54254/2755-2721/95/20241748

(a) Original video

(b) Synthetically generated video (0.8x resize, 1.5x brightness, grayscaling, 90 degree rotation)

Figure 7: Comparison of videos from the LSF dataset

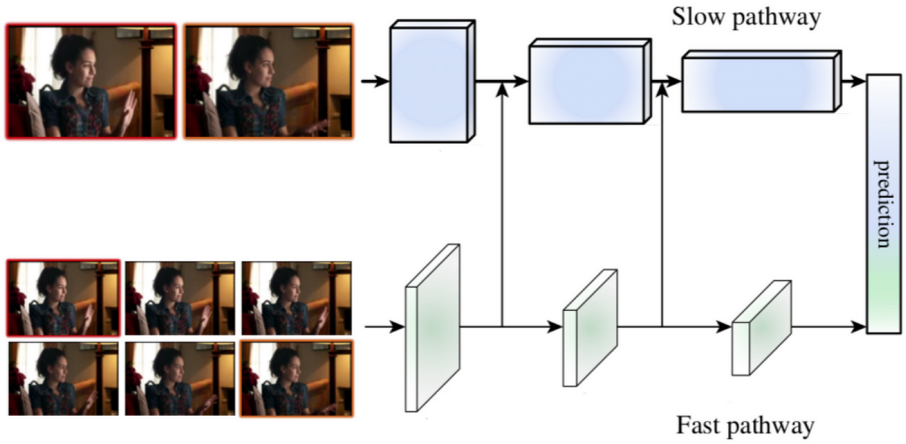

Figure 8: The SlowFast model architecture

2.1.3. The pre-trained model It is worth noting that sign language contains a temporal aspect, incorporating both static and dynamic gestures. In the realm of machine learning, this translates to spatial semantics and temporal resolution, respectively. The former permits detecting relatively static movements, while the latter allows for detecting gestures that last over a large time frame. As such, we needed to find a model capable of detecting gestures from both of these categories.

After further research, the SlowFast model came to mind and was the optimal choice to use in our project [21][22] (more details about the rational can be found in Section 4). With its two pathways operating in tandem, the model is able to capture spatial semantics and high temporal resolutions (Fig.8). Its two pathways perfectly correspond to the still and dynamic gestures found in LSF, thus making it suitable for sign language recognition.

2.1.4. Finetuning There are usually two primary approaches to finetune an existing model: the first option is to leave the model architecture intact, and simply change the output channel to adapt to the

DOI: 10.54254/2755-2721/95/20241748

new dataset that will be trained on. The second approach is to utilize the pre-trained model exclusively as a feature extractor, and to subsequently add new layers to adapt to a new task. In this scenario, we ultimately opted for the first approach, as we believe the SlowFast model architecture is already serving our purpose. Adding new layers onto the existing model would also increase the network size and lead to more computation, which is something we aim to avoid. As such, the fine tuning step consisted mainly of changing the number of output channels from 400 to 1245, to match the number of class indices in our word map and dataset. The SlowFast model was initially trained using the kinetics 400 dataset, and should thus be altered.

For our loss function, we chose to use the Cross Entropy loss function, which is given as follows:

$$L = -\sum_{i=1}^{C} y_i \log(p_i) \tag{1}$$

Where:

- L is the cross entropy loss
- C is the number of classes
- y_i is the true label for class i
- p_i is the predicted class distribution
- $log(p_i)$ is the natural logarithm of the predicted probability for class. The Softmax function is used in this case.

2.1.5. Training We trained the model using two Nvidia 4090 chips, for a total of 3 epochs. A version of the model is saved each time it reaches a new minimal loss. This effectively prevents the possibility of over-fitting the model, as if that occurs we would be able to utilize early stopping and load a previous version of the model with better performance. The training curve is visualized in Section 3, Fig.11. We used the standard Adam optimizer with a 0.0001 learning rate. This value was decided upon from empirical evidence, and is further discussed later on in the Discussion.

2.2. Deployment phase

This phase (Fig.3) is split into two parts: the back end, and the front end, which operate simultaneously. A server is utilized to power the functionalities of the app. The server communicates with the terminal, which is displayed to the user. The app is able to access the device's front camera for real time sign language recognition. The individual detected words are then sent to the LLM for sentence generation.

2.2.1. App interface App interface: After compiling the model into a format that is easy to integrate, we conceived an app with an intuitive UI (Fig.9) and embedded the model within that. Once the device's front camera is placed in front of the signer, the user will press a button to begin recording, and press again to end the recording. The resulting video is converted into an mp4 format and sent to a server to process.

DOI: 10.54254/2755-2721/95/20241748

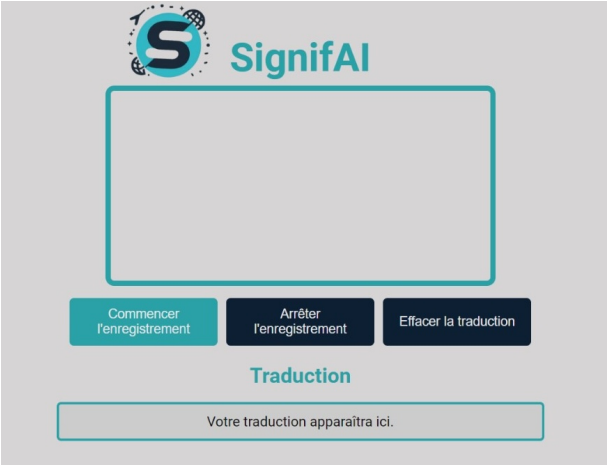

Figure 9: Prototype app interface

2.2.2. LLM integration The LLM integration comes at the very end of our translation process, and is also one of our primary innovations. After all the individual gestures have been translated into their corresponding french words, the LLM takes in all of these discrete inputs and, through its contextual understanding, reformulates everything into a fluent sentence. We chose ChatGPT [23] as our LLM over other alternatives, thanks to its easy to use API and its mature performance. Later on, when we bring SignifAI to a larger scale, we will consider switching to an open source LLM, which will allow us to integrate it into the system, avoiding the use of an API . Below is an example that illustrates ChatGPT 3.5's capabilities to understanding meaning and relate words together (Fig.10).

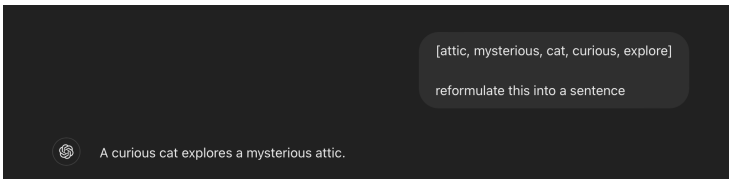

Figure 10: ChatGPT reformulating a sentence

The use of an LLM in our translation system also compensates for some of the model's prediction inaccuracies: if one word stands out as an outlier and fails to align with the overall context, the LLM can intelligently replace these mismatched words with alternatives that better fit the intended meaning.

3. Experimental results

3.1. Test performance at the word level

Once the training loop finished running, we entered the testing phase. The entire training-testing phase is an iterative cycle, with the primary objective of improving the model accuracy. When this has not been attained to a high enough degree, the training-testing process is carried out again.

The testing dataset we utilized consists of a total of 2563 videos (as shown in Fig.6), with at least one copy for each word contained within the word map. Back when we synthetically augmented our data, we generated 10 new videos for the training dataset, and 1 or 2 videos for the testing dataset. As such, the videos contained within the testing dataset are remotely similar to the training data that the model has previously seen, but the color shifts and size changes effectively make these testing videos new.

To calculate our validation loss, we utilized once again the Cross Entropy Loss [24] to maintain coherence. The validation accuracy is then subsequently calculated as in (2):

DOI: 10.54254/2755-2721/95/20241748

$$\text{Accuracy} = \left(\frac{\text{Number of Correct Predictions}}{N} \right) \times 100\%, \tag{2}$$

where N is the batch size. This process is iterated through all the batches, and the total accuracy is obtained when adding up the individual results from each batch.

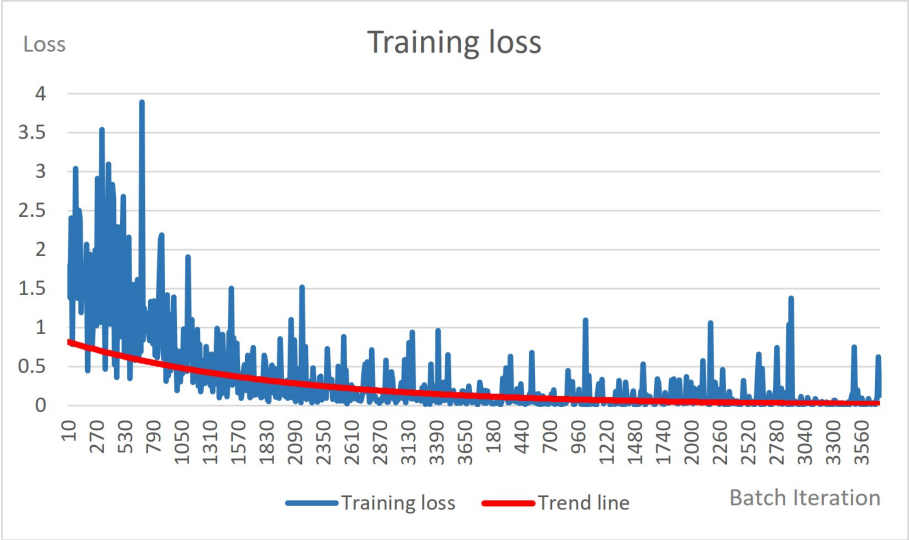

Figure 11: The training loss curve

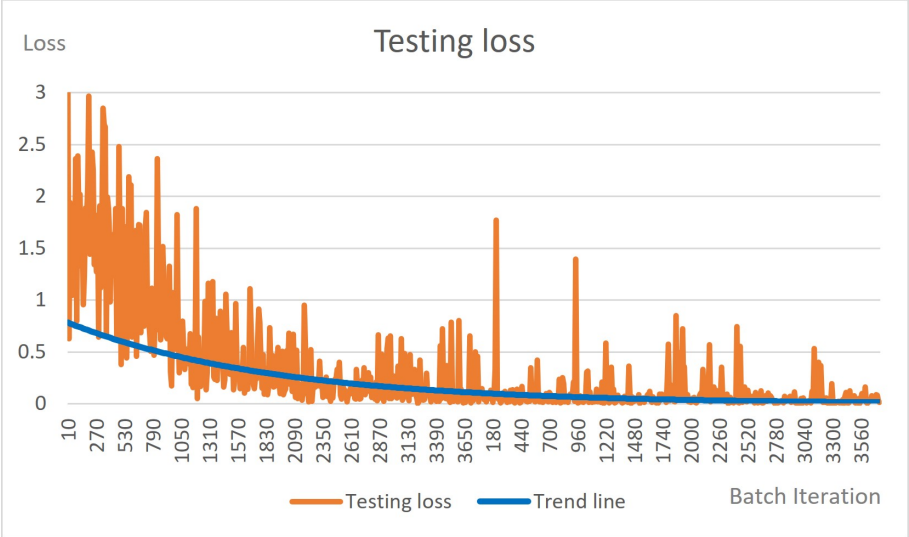

Figure 12: The testing loss curve

DOI: 10.54254/2755-2721/95/20241748

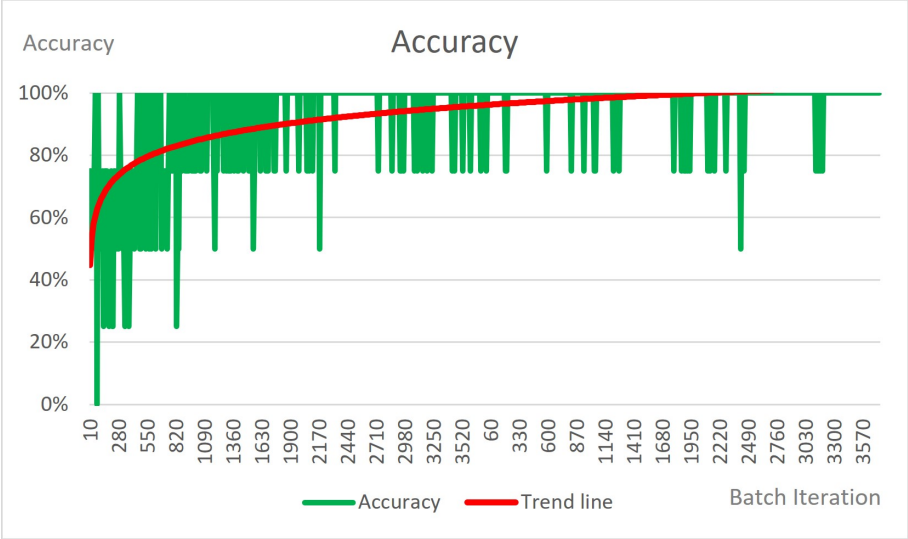

Figure 13: The accuracy improves over iterations

The training and testing loss curves, as well as the accuracy score are shown in Figure 11, Figure 12, and 13, respectively. We trained the model for three epochs, each consisting of 3670 batches, with batch size = 4. In the training loss curve we can observe a general decreasing trend over the batch iterations, going from a value of 1.78 in batch 10, to a value of 0.0114 in the final batch (Fig.11). The testing loss follows a similar decreasing trend, going from 2.99 in batch 10, to 0.0179 in the final batch (Fig.12). Lastly, the accuracy curve undeniably follows an increasing curve and improves over the batch iterations, going from 40% in batch 10 to 98.9% in the final batch (Fig.13).

3.2. Experimental setup for end-to-end testing

We had a camera setup facing a white background, with sources of light coming from each of the four corners pointing towards the speaker (as shown in Fig.15. Doing so ensures that the lighting is uniform, and also helps with fully exposing the facial features of the speaker, a key element in sign language. Furthermore, we instructed the speakers to wear dark colored, monotone pieces of clothing, as a means of minimizing noise and distractions. The signers involved performed in varying proficiency: this simulates real life situations in which not all users are perfectly fluent in sign language, perhaps having minor differences in their hand gestures. When this occurs, the model must be powerful and capable enough of taking these alterations into account, and still give a correct prediction. The age and gender of signers also varied (as shown in Fig.14), to further diversify the data and encourage the model to generalize the patterns that it observes.

DOI: 10.54254/2755-2721/95/20241748

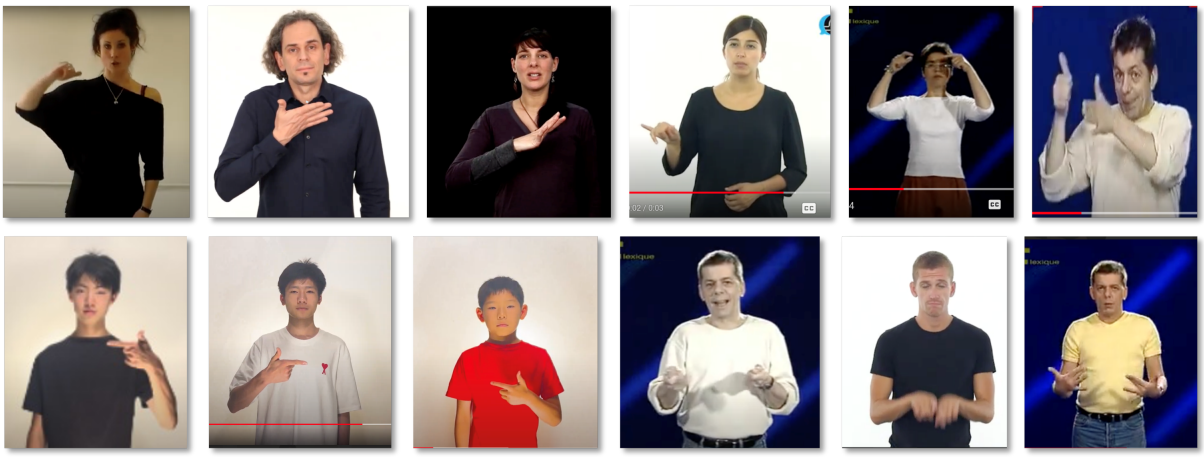

Figure 14: The vast collection of signers in our dataset

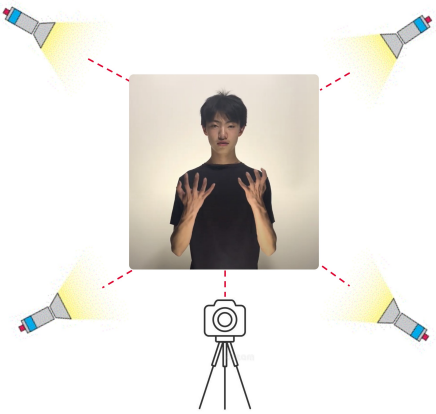

Figure 15: Example of our experimental setup

This collection of recorded videos follow the same standard process: each video will be sent to the trained network for word-level recognition, and the recognized sequence of words will be sent to ChatGPT to generate the full sentence.

3.3. Case study

An illustrative example is provided in Fig.16 , demonstrating a real-world use case of the application: grocery shopping. For instance, when a user is grocery shopping, they might want to communicate a sentence like ”I would like to buy 2 kilograms of apples.” The user would sign this sentence in LSF using the SignifAI app, where each gesture is processed individually. The predicted signs are then compiled into a list, maintaining their original order from the LSF sentence (as sign language syntax differs from that of spoken languages). This list is sent to ChatGPT with a tailored prompt, resulting in a fluently formulated French sentence being returned to the user. The LLM will also attempt to capture the speaker’s intent and sentiment concealed behind the translated words, and offer advice on how to communicate back to the signer accordingly. Through this hypothetical use case, we can see how SignifAI effectively assists the deaf in overcoming everyday communication challenges.

DOI: 10.54254/2755-2721/95/20241748

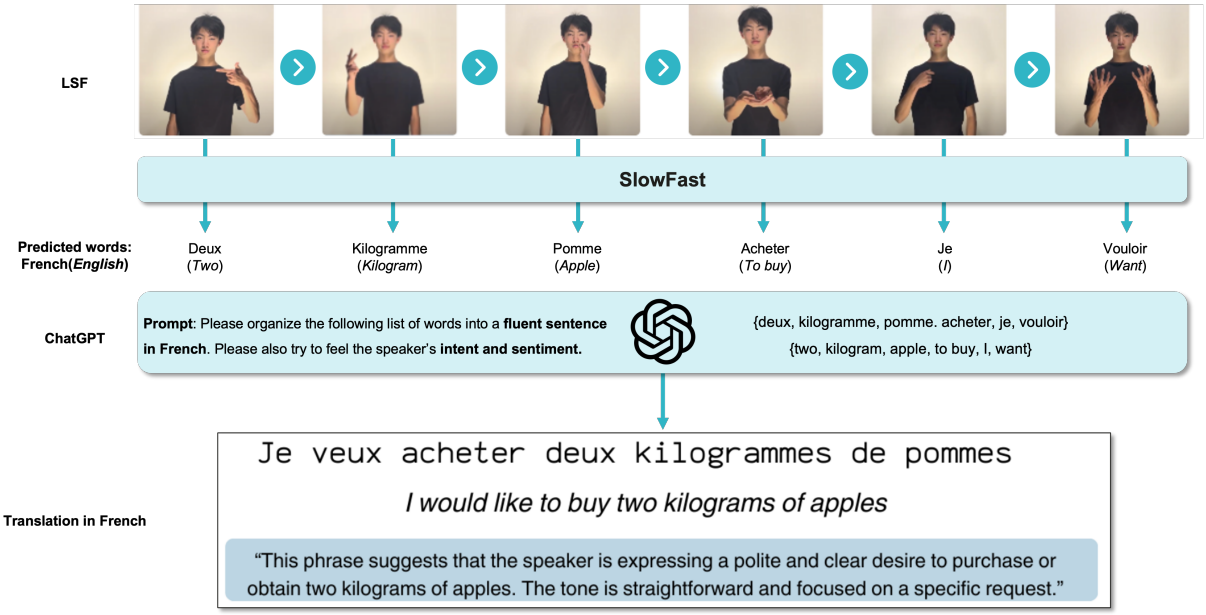

Figure 16: Illustrative example of the model detection and translation process

4. Discussion

4.1. Model Selection

Before we decide the final choice to use SlowFast, we also have explored the usage of other models, such as VideoMAE [25], and LSTM, but they all could not meet our specific demands in this situation. The VideoMAE model, for starters, had a high computation cost, as it required 8 A100 GPU cards which were sadly not available. The LSTM network, on the other hand, did not have the necessary complexity and scale to deal with such a large translation task. The choice of models is further documented in the radar chart (as seen in Fig. 17). The SlowFast model is more all-rounded compared to the other two choices.

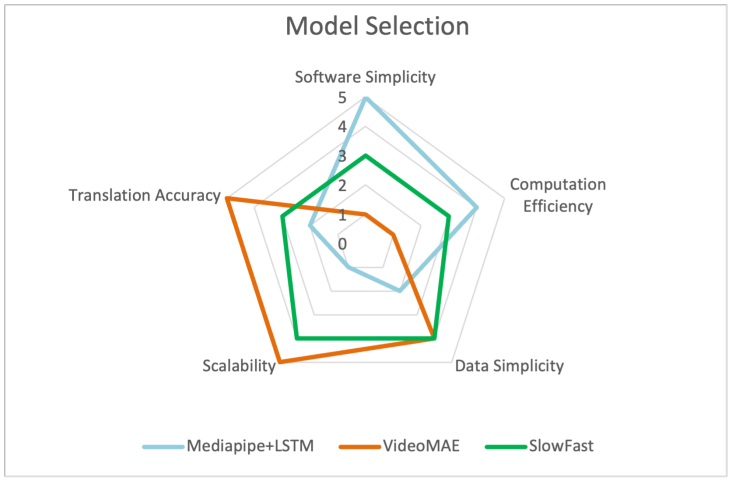

Figure 17: Radar chart comparing the different models

We started out this project using the LSTM network, in tandem with the Mediapipe Holistic package [26] for feature detection. This LSTM + Mediapipe combination proved to be a great entry point into the project, and helped immensely in the prototyping phase.

DOI: 10.54254/2755-2721/95/20241748

We trained the prototype model to detect three signs in ASL: ”hello”, ”I love you”, and ”thanks”. We then fed the model 90 videos that we recorded ourselves, 30 for each sign. This formed the training dataset.

After training, we were ultimately able to obtain a detection accuracy of 98% on the LSTM model. To test the model, we accessed the device camera via OpenCV [27] and signed facing the camera. The model would accurately detect the three signs in real time (as shown in Fig.18).

Although the LSTM + Mediapipe combination initially served as a great prototype, the model architecture was not complex enough to handle a full scale translation task. That is why we ultimately switched to the more complex SlowFast model in the subsequent iteration.

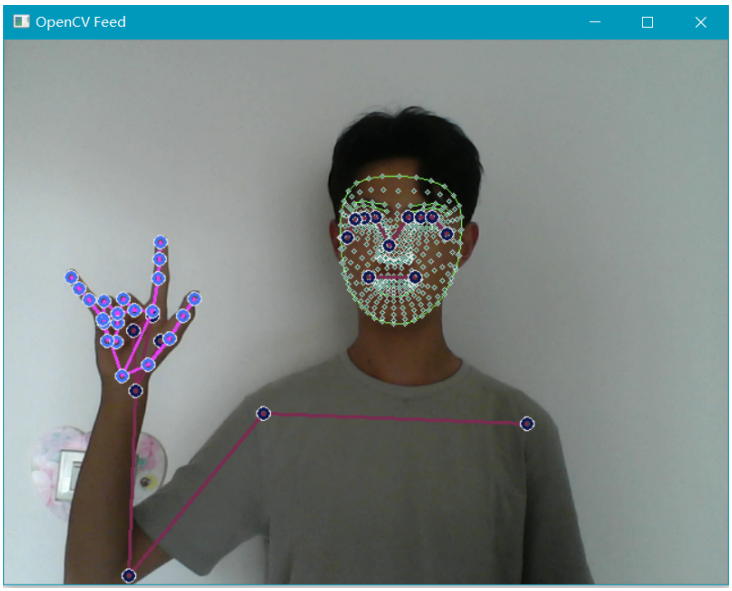

Figure 18: Mediapipe detecting ”I love you” signed by the user

4.2. Hyperparameter tuning - learning rate

During our initial training phase, we tested a diverse set of different learning rates in attempt to minimize the loss. We first started with a learning rate of 10^{-3}, often considered to be the industry standard. However, after training for 1000 batches, we noticed that the training loss still remained abnormally high, stagnating at a value of 7.0. We subsequently experimented with an increased learning rate of 10^{-2}, which only resulted in a higher overall loss of 7.2 with no decreasing trend. Finally, we tested a learning rate of 10^{-4}, which turned out to be the optimal value in this scenario. After 2 epochs, the training loss drastically decreased, going from 7.2 down to 0.044. The table below (Table 1) effectively summarizes our empirical observations.

Table 1: The change in loss depending on the specified learning rate

Learning rate	Initial Training Loss	Final Training Loss	Change in loss
0.01	7.2	7.1	0.1
0.001	7.3	6.8	0.5
0.0001	7.2	0.04	7.16

DOI: 10.54254/2755-2721/95/20241748

4.3. Prompt Temperature

There are various adjustable parameters when specifying the prompt, one of them being the prompt temperature. The temperature value of the prompt affects the creativity and randomness in the final output. It is a value ranging from 0 to 1, with 0 gearing towards focused, common responses, and 1 leaning towards more creative ones. We tested various temperature values and compared their final outputs, as shown below.

Original sentence: The bird sings in the blue sky during the morning sunrise.
Prompt: Imagine yourself as a helper assitant for French deaf people. Organize these words into a fluent sentence: [morning, sunrise, sky, blue, bird, sing]

Table 2: Prompt temperature results

Prompt temperature	Output
1	In the morning, the blue sky welcomes the sunrise as birds sing.
0.8	In the morning, as the sun rises, the sky turns a beautiful shade of blue, and birds begin to sing.
0.6-0.0	In the morning, as the sun rises, the blue sky is filled with birds singing.

It can be seen that all outputs generally capture the main idea of the sentence, with slight variations in the formulation. As we go under a temperature value of 0.6, the outputs become identical each time, maintaining consistency. We ultimately opted for a temperature value of 0.7, as it returns a blend of creativity and consistency in the output.

5. Conclusion

5.1. Synthesis

In this project, we proposed SignifAI as a French Sign Language translation software that leverages the power of Deep Learning and Large Language Models.

The obtained result is a finetuned version of the SlowFast model, trained on 14,490 synthetically enhanced sign language videos. Ultimately, we were able to obtain a maximum test accuracy of 98.9%. We successfully demonstrated the potential impact of SignifAI on the deaf community through a case study: the model is able to accurately detect the sentence "I would like to buy two kilograms of apples", a saying that is useful in grocery shopping, which is an integral part to every person's everyday life.

5.2. Future Work

However, what we currently have is only the first iteration of many to come. Being still in the prototype phase, SignifAI does have certain limitations for the time being.

Real time translation: At its current stage, the model can only take in pre-recorded mp4 videos as input, and process those instead. Our next step in this direction would be to implement a streaming feature using OpenCV, that samples over a live video recording upon fixed time intervals, in order to truly achieve live translation. In an upcoming version of SignifAI, the user will sign in real time towards a device camera, and take short three second pauses in between each sentence. During this time interval, the model will process the words and return the fully translated sentence.

Improving the model accuracy: As discussed previously, the currently obtained 98.9% is still unstable. This is essentially caused by our current dataset not being diverse enough. Containing mostly synthetically augmented videos, each sign is often performed by the same person, under different angles and lighting conditions perhaps. As such, the best solution to improve the accuracy is to expand our existing dataset. We have already commenced work in this direction, having acquired the Matignon-LSF dataset [28] containing live interpretations of LSF during public speeches. We also plan to obtain access to the Mediapi-RGB dataset [29], containing 86 hours of LSF videos.

DOI: 10.54254/2755-2721/95/20241748

Emotion detection: Emotion, and facial expressions in general, are fundamentally core to sign language. If the same sign were produced with with a slightly different facial expression, the entire meaning can change. For instance, take the example of ”interesting”: if the signer had a smile on their face, this could translate to ”It’s interesting!”, as the speaker expresses immense enthusiasm. However, if the signer squinched their eyebrows, the word can translate to ”It’s interesting...”, indicating a certain sense of skepticism and criticality. It is vital for SignifAI to be able to take into account the signer’s emotions and facial expressions, and eventually transmit these information to the LLM during the translation process. In some situations, the user’s face may be obstructed (e.g., during the events of COVID when the user is wearing a mask). When this happens, the LLM can utilize its contextual understanding to predict the most apropos emotion in the given situation.

Voice-over feature: Once paired with a voice-over add on, the model can also help the deaf place phone calls in their daily lives - maybe for a doctor’s appointment, as illustrated below (Fig.19). The translated text from sign language is transmitted into a voice-over software that pronounces the words out loud. The audio response from the other end is then converted into sign language performed by a meta-human, returned back to the deaf user. This entire process is a cycle, and lasts for the duration of the call.

Figure 19: Illustrative example of the SignifAI translation system working with a voice-over feature

Multilingual support: Once LSF support is successfully implemented, we could expand SignifAI’s sign language translation features to incorporate even more dialects, such as the commonly used American Sign Language, British Sign Language, and Chinese Sign language. On the other end of the line, we could also offer more text-based languages to choose from, beyond the sole option of French.

Translation between sign language dialects: In addition to this, we will also work on translation between the existing sign languages, to attempt breaking the language barriers between the dialects themselves, and in doing so, uniting the deaf community around the world.

Acknowledgement

I would first like to thank Dr Liting Sun and Mr. Yming Zhu for their academic insight in the SignifAI project. I would also like to thank Mr. Jonas Sanson and M. Romain Primet at INRIA (National Institute for Research in Digital Science and Technology in France) for their professional input, and Mr. Yanis

DOI: 10.54254/2755-2721/95/20241748

Oukarim for his aid in acquiring the LSF datasets. I would also like to give special thanks to my peers, Noa Aguilar, Hien Anh Dang, Romain Ferrandon, Siena Kamalodine, and Lilia Srun, who assisted me in carrying out this initiative as a community service project. Lastly, I would like to express my sincere regards towards my family, for their relentless support over the past year.

References

[1] Study.com. *The History of Sign Language*. https://study.com/academy/lesson/the-history-of-sign-language.html. Accessed: 2024-08-21. 2024.

[2] World Health Organization. "Deafness and Hearing Loss". In: *Who.int* (Sept. 2019).

[3] Direction de la Recherche, des Études, de l'Évaluation et des Statistiques (DREES). *Report on Social and Health Solidarity in France*. https://drees.solidarites-sante.gouv.fr/sites/default/files/2020-08/dss52.pdf. Accessed: 2024-08-21. 2020.

[4] Project EASIER. *There is No Universal Sign Language*. https://www.project-easier.eu/news/2022/09/22/there-is-no-universal-sign-language/. Accessed: 2024-08-21. Sept. 2022.

[5] IBM. *Natural Language Processing (NLP)*. https://www.ibm.com/topics/natural-language-processing. Accessed: 2024-08-21. 2024.

[6] Lifeprint. *American Sign Language Grammar*. https://www.lifeprint.com/asl101/pages-layout/grammar.htm. Accessed: 2024-08-22.

[7] Aymeline LSF. - *YouTube*. https://youtu.be/y5JyG4s5rBo?si=ANxwt1xA0zwArYhJ. 2024.

[8] National Deaf Center on Postsecondary Outcomes. *Video Remote Interpreting*. https://nationaldeafcenter.org/resources/access-accommodations/coordinating-services/interpreting/video-remote-interpreting/. Accessed: 2024-08-21. 2024.

[9] University of Washington News. *UW Undergraduate Team Wins $10,000 Lemelson-MIT Student Prize for Gloves that Translate Sign Language*. https://www.washington.edu/news/2016/04/12/uw-undergraduate-team-wins-10000-lemelson-mit-student-prize-for-gloves-that-translate-sign-language/. Accessed: 2024-08-21. 2016.

[10] Keith Kirkpatrick. "Technology for the deaf". In: *Commun. ACM* 61.12 (Nov. 2018), pp. 16–18. ISSN: 0001-0782. DOI: 10.1145/3283224. URL: https://doi.org/10.1145/3283224.

[11] BrightSign Glove. *BrightSign Glove: A Smart Glove that Translates Sign Language*. https://www.brightsignglove.com/. Accessed: 2024-08-21. 2024.

[12] Samantha Johnson et al. "An Adaptive, Affordable, Open-Source Robotic Hand for Deaf and Deaf-Blind Communication Using Tactile American Sign Language". In: *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*. 2021, pp. 4732–4737. DOI: 10.1109/EMBC46164.2021.9629994.

[13] University of Antwerp. *Sign Language Robot Project ASLAN*. https://www.designboom.com/technology/sign-language-robot-project-aslan-08-23-2017. Accessed: 2024-08-21. 2017.

[14] Kaggle. *Sign Language MNIST Dataset*. https://www.kaggle.com/datasets/datamunge/sign-language-mnist. Accessed: 2024-08-21. 2024.

[15] Google Developers. *SignAll SDK: Sign Language Interface Using MediaPipe is Now Available for Developers*. https://developers.googleblog.com/en/signall-sdk-sign-language-interface-using-mediapipe-is-now-available-for-developers/. Accessed: 2024-08-21. 2024.

[16] Mi Zahng, Biyi Fang, Jillian Co. *DeepASL: A Wearable System for Unobtrusive End-to-End American Sign Language Translation*. https://msut.technologypublisher.com/technology/26552. Accessed: 2024-08-21. 2024.

[17] NHK Enterprises. *KIKI: She specializes in sign language*. https://www.nhk-ep.co.jp/signlanguage/en/. Accessed: 2024-08-21. 2024.

[18] HandTalk. *HandTalk App: Bridging Communication through Sign Language*. https://www.handtalk.me/en/app/. Accessed: 2024-08-21. 2024.

[19] Parlr. *Parlr/lsf-data: A repository for LSF data*. https://github.com/parlr/lsf-data. Accessed: 2024-08-20. 2023.

[20] Elix Dictionary. *Dictionnaire Elix - Langue des Signes Française (LSF)*. https://dico.elix-lsf.fr. Accessed: 2024-08-21. 2024.

[21] Christoph Feichtenhofer. *SlowFast*. https://github.com/facebookresearch/SlowFast. Accessed: 2024-08-20. 2019.

[22] Christoph Feichtenhofer et al. "SlowFast Networks for Video Recognition". In: *CoRR* abs/1812.03982 (2018). arXiv: 1812.03982. URL: http://arxiv.org/abs/1812.03982.

[23] OpenAI. *OpenAI API Reference: Introduction*. https://platform.openai.com/docs/api-reference/introduction. Accessed: 2024-08-21. 2024.

[24] PyTorch. *torch.nn.CrossEntropyLoss*. Accessed: 2024-08-22. 2024.

[25] Zhan Tong et al. *VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training*. 2022. arXiv: 2203.12602 [cs.CV]. URL: https://arxiv.org/abs/2203.12602.

[26] Google AI. *MediaPipe Holistic*. https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/holistic.md. Accessed: 2024-08-22.

[27] OpenCV. *OpenCV: Open Source Computer Vision Library*. https://github.com/opencv/opencv. Accessed: 2024-08-22.

[28] Julie Halbout et al. "Matignon-LSF: a Large Corpus of Interpreted French Sign Language". In: *LREC-COLING 2024 11th Workshop on the Representation and Processing of Sign Languages: Evaluation of Sign Language Resources*. Ed. by ELRA Language Resources Association (ELRA) and the International Committee on Computational Linguistics (ICCL). Turin, Italy, May 2024, pp. 202–208. URL: https://hal.science/hal-04593865.

[29] Yanis Ouakrim et al. "Mediapi-RGB: Enabling Technological Breakthroughs in French Sign Language (LSF) Research through an Extensive Video-Text Corpus". In: *VISAPP 2024 - 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. Vol. 2. Rome, Italy, Feb. 2024. DOI: 10.5220/0012372600003660. URL: https://hal.science/hal-04494094.