Talking with signs
A simple method to detect nouns and numbers in a non-annotated signs language corpus

Eric Raphael Huiza Pereyra, César Augusto Olivares Poggi
Pontificia Universidad Católica del Perú
Av. Universitaria 1801, San Miguel, Lima 32, Perú
eric.huiza@pucp.edu.pe, cesar.olivares@pucp.edu.pe

Abstract

People with deafness or hearing disabilities who aim to use computer based systems rely on state-of-art video classification and human action recognition techniques that combine traditional movement pattern recognition and deep learning techniques. In this work we present a pipeline for semi-automatic video annotation applied to a non-annotated Peruvian Signs Language (PSL) corpus along with a novel method for a progressive detection of PSL elements (nSDm). We produced a set of video annotations indicating signs appearances for a small set of nouns and numbers along with a labeled PSL dataset (PSL dataset). A model obtained after ensemble a 2D CNN trained with movement patterns extracted from the PSL dataset using Lucas Kanade Opticalflow, and a RNN with LSTM cells trained with raw RGB frames extracted from the PSL dataset reporting state-of-art results over the PSL dataset on signs classification tasks in terms of AUC, Precision and Recall.

1. Introduction

The World Health Organization (WHO) stated that 466 million people world wide have disabling hearing loss, estimating that by 2050 over 900 million people will have disabling hearing loss that will represent a global cost of 750 million dollars annually [5].

The Peruvian Institute of Informatics and Statistics (INEI) conducted a national disabilities survey with the objective of segmenting and acquiring a better understanding about disabilities that affect the Peruvian population [3]. Results showed that 1.8% of the Peruvian population suffer at least partial when not permanent deafness or hearing limitations.

Peruvians with deafness or hearing limitations use the Peruvian Signs Language (PSL) as their main communication medium. PSL is of mandatory usage at universities and certain public institutions, henceforth the importance of designing systems that are capable to support PSL inputs and outputs. Furthermore, in the same way as spoken languages, signs languages also present local variations e.g. people who live in Lima metropolitan area are not expected to use the same set of signs as people in other parts of the territory. This work uses the PSL variation used in Lima due to the difficulty or inability to find datasets for other PSL variations.

The Grammar and Signs research group of the Pontifical Catholic University of Peru (PUCP) built the first PSL corpus [4] which is publicly available at the university digital archives. It is important to highlight that the corpus is neither labeled or annotated and cannot be used as it is for training or testing a model.

In this work we are approaching signs detection as a supervised learning task. Supervised learning requires labeled datasets to achieve satisfactory results during training and inference tasks. At the time of writing this work there were no labeled datasets available for PSL [2]. It configures a gap that could prevent or hinder research work on Human-Computer-Interaction at the Peruvian or Latin American space.

Current advances in Computer Vision (CV) and Natural Language Processing (NLP) make it possible to conceive systems that are capable of detecting and transcribing elements of sign languages thereby improving systems accessibility for people with physical limitations. This work reports results of a research conducted with the goal of producing a labeled PSL dataset for a set of signs limited to nouns and numbers as well as a novel method for detecting PSL signs by answering the following research questions:

- What are the currently available techniques for producing a labeled dataset for a set of signs limited to nouns and number from the non-annotated PSL corpus?
- What are most relevant and currently available techniques for training a model with the labeled dataset described in the question above for detecting PSL nouns and numbers?
- How precise and exhaustive is the model described in the above question on the detection of PSL nouns and numbers?

This work has the main objective of producing a simple method that can be used as a baseline for other researchers interested on studying signs language and their different applications on the Human-Computer-Interaction field.

The rest of the article is organized as follows. In section 2 we review the related work on video classification for human actions recognition using network architectures that combine CNNs, 3D CNNs and movement patterns for better features learning, we also review state-of-art pose estimation techniques. In section 3 we introduce nSDm describing its design and architecture. In section 3.1 we describe the video annotation and data pre-processing techniques applied to produce the labeled PSL dataset. In section 4 we evaluate nSDm precision and recall and answer research questions. In section 4.1 we describe the PSL dataset produced at PUCP and finally in sections 5 and 6 we present our conclusions and future work.

2. Related Work

2.1. Action Recognition

Human action recognition is an extensively studied field. Action recognition dataset like UCF101, HMDB51, THUMOS14 are available, researches tried to solve the human action recognition problem using different approaches including Optical Flow and 3D CNN [6].

Optical Flow, is defined as the pattern obtained from the motion of objects, surfaces and edges in a visual scene caused by the relative motion between the observer and a scene. It is computed by distributing movement velocities and brightness across frames. It is a key concept in action recognition from videos [9]. Optical flow estimation is treated as an image reconstruction problem. Given a frame set, the optical flow is generated and allows to reconstruct one frame from the others [10]. Formally, taking the optical flow displacement field as input and training a CNN with it, then the network should have learned useful representations of the underlying motions. Even though Optical Flow represents the movement between a set of frames, if camera motion is considered as an action motion, it may corrupt the action classification [8]. Various types of camera motion can be observed in realistic videos, e.g., zooming, tilting, rotation, etc.

Motion Boundary Histogram (MBH) is a simple an efficient way to achieve robustness during human action detection when camera movements are mixed within the recorded actions by computing derivatives separately for the horizontal and vertical components of the optical flow. Since MBH represents the gradient of optical flow, locally constant camera motion is removed and information about changes in the flow field is kept. MBH is more robust to camera motion than optical flow, thus more discriminative for action recognition.[8]. 3D CNN are not as effective as optical flow to detect human actions on its own, 3D CNN can be trained to learn optical flow so we can avoid costly computation and storage and obtain task-specific motion representation [10] and increase models performance, precision and recall on human action recognition.

2.2. Pose Estimation

Pose estimation is also an extensively studied field. Techniques based on key points have shown state-of-art results on human pose estimation. An approach on key points estimation [7] uses *Point of View Determination* and *Key Points Prediction* components. Point of View Determination is formulated by the prediction of three Euler angles (azimut, elevation and cyclotation) generating a global position estimate, then a local appearance is modeled by obtaining a heat map that corresponds to the spatial distribution likelihood for each key point, finally key points predictions are obtained by combining heat maps obtained in a previous stage with a conditioned likelihood at the point of view predicted in the previous stage.

Key points detection methods based CNNs have received an special attention in Human Pose Detection problems. CNNs methods are divided in bottom-up and top-down. Bottom-up methods process images from low resolution to high resolution, focusing first on detecting joints before associating them to human actions. Top-down methods focus first on detecting human subjects and then estimating the human pose to predict key points.

The datasets MPII and COCO have been used in state-of-art methods obtaining good results[1] and establishing a framework for future work in combination with classic approaches like optical flow for recognizing patterns movement between frames by increasing accuracy on key points detection.

2.3. Video Classification

Bag of Words (BoW) or **Bag of Visual Words (BoVW)** based on natural language processing techniques is one of the simplest and oldest local descriptor encoding strategies. In its simplest form, it consists of (i) clustering with k-means a collection of descriptor vectors from the training set to build so-called visual vocabulary, (ii) as signing each descriptor to its nearest cluster center from the visual dictionary, and (iii) aggregating the one-hot assignment vectors via average pooling [9], when applied to Computer Vision is a technique used to create images representations or features vectors used that can be learned by CNNs, resulting on improved images classification and video classification. Feature trajectory detection are much improved using statistical methods like **Fisher Vectors** obtaining better results over traditional BoW **Fussing parallel CNN.**. The Bag of Visual Words representation suffers from sparsity and high dimensionality, in the other hand representations obtained

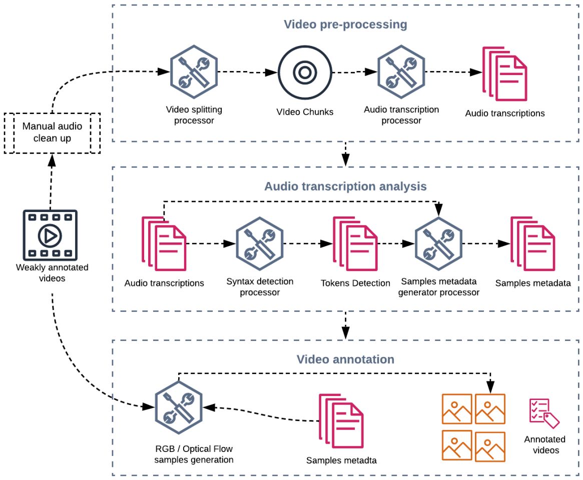

Figure 1. Video annotation process

using the Fisher Vectors kernel are more compact and dense which results on better results for image and video classification problems.

3. Method

3.1. Video Annotation

The PSL dataset is non-annotated because there is not a direct relation between the instant when a sign is emitted and when its translation to Spanish is delivered. We propose a semi-automatic video annotation pipeline described in Figure 1 for cleaning, pre-processing and analyzing PSL videos in order to produce an labeled PSL dataset that can be used for training nSDm using supervised learning. The pipeline is described in detail in sections 3.1.1, 3.1.2, 3.1.3 and 3.1.4

We used the PSL dataset to train and test a set of neural networks described in detail in sections 3.2, 3.3 and 3.4

Implementation details can be found at https://github.com/erichuizapucp/signs-recognition

3.1.1 Semi Automatic Video Clean Up

The PSL recordings described on 4.1 contain a considerable amount of noise introduced during recording sessions. It makes difficult to easily find video intervals that clearly show a relation between signs emitted by the informant and the translation delivered by the translator. Noise factors are the following:

- Multiple participants speaking during the session.
- Conversations between participants that are not relevant to emitted sings.
- High frequency of large silent periods.

Video	Start	End	Alignment
01-session-01-part-01.mp4	00:30	00:55	center
	01:15	01:29	center
	00:53	01:07	center
	08:12	09:01	center
02-session-01-part-01.mp4	00:15	00:21	center
	00:15	00:21	center
	00:53	01:07	center
	02:43	02:47	center
	17:33	18:01	left

Table 1. Noise free video segments extract

A manual video cleanup process is required to find noise free video intervals. This process requires watching all videos available at the PSL corpus for manually annotate the instant when an informant started emitting sings along with the instant when the translator delivered a translation. Table 1 shows a manual annotation example.

The recordings show the informant in two alignments (centered and left), the manual video clean up process also stores the informant alignment, we use the alignment annotation later in the process during the video frames extraction to create the labeled PSL dataset.

3.1.2 Video Pre-Processing

Non-annotated PSL videos require processing before any metadata can be extracted, we propose a sequence of preprocessing tasks that take advantage of the annotation generated on 3.1.1. A **video splitting processor** generates a set of video chunks using the *ffmpeg* multimedia framework and stores produced video chunks in Amazon S3 for later usage. Audio within video chunks is then transcribed by an **audio transcription processor**, using the Amazon Transcription service, we selected the Amazon Transcription service because it provides an accurate mapping between audio participants and transcribed words along with useful metadata that describes the start and end time when words are pronounced by the translator.

At the moment of writing this work Amazon Transcription service only supported Spain and US Spanish. This caused certain words that are specific for Peruvian Spanish not being fully recognized, in order to improve transcription accuracy we built a custom vocabulary containing Peruvian expressions which improved Peruvian words recognition, for the matters of this work Peruvian words that remained unrecognized were omitted and not processed.

3.1.3 Audio Transcription Analysis

Audio transcription requires additional processing in order to produce useful information that leads to a successful PSL signs detection. Bag of Embedding Words (BoEW) is a widely used technique on Natural Language Processing tasks providing a easy and flexible way to list the most relevant words based on frequency. This work is focused on detecting nouns and numbers (our method is designed to be progressively improved to handle a wider set of PSL elements) assuming that nouns (numbers are a subset of nouns) suffer less variations in spoken Spanish than verbs, pronouns, adverbs and adjectives, and provide more semantic value than conjunctions, prepositions and interjections.

We used Amazon comprehend for text analysis, specifically the syntax detection functionality which will provide a comprehensive list of detected language elements along with a score from 0.0 to 1.0 indicating the detection accuracy, we have selected the ones that have at least a 0.8 accuracy score and omitted the rest, this process was automated using a **transcription detection processor** which uses BoEW to provide a list of most relevant nouns and numbers based on appearance frequency.

Once a weighted list of nouns and numbers is generated a mapping showing when nouns and numbers appear in videos is required, moving forward called **Samples Metadata**. Table 2 shows mapping metadata extracted from PSL.

3.1.4 Samples Generation

Our method requires PSL elements to be represented as a set of RGB frames and a calculated Optical Flow using the Lucas-Kanade method, both representations are inputs of two different models as presented on 3.4.

Translation Delay Factor: The difference in time between the instant when a sign is emitted and when a translation for that given sign is delivered is uncertain, we are calling that uncertainty *the translation delay factor*, we are trying to approximate it using a constant value, we chose a three seconds translation delay factor assuming that most of the translations will occur between three seconds after a sign is emitted.

A **RGB Samples generation processor** uses samples metadata in combination with the translation delay factor to determine frames that represent a given PSL element. We use *OpenCV* to extract frames and store them following a hierarchical folder structure that nSDm data loaders will use to feed data into the RGB branch in the nSDm model architecture 3.4.1 during training and testing.

An **Optical Flow Samples generation processor** uses video frames and the hierarchical folder structure generated by the RGB samples generation processor to calculate an Optical Flow representation for PSL elements and store them in a hierarchical folder structure that will also be used by the nSDm data loaders to feed the optical flow branch on the nSDm model architecture 3.4.1 during training and testing. We selected optical flow as a samples generation strategy due to its ability to represent movement traces from previous frames. It is particular useful for representing body movement patterns executed by informant while emitting a PSL sign. A PSL sign is made up of different body movements including: elbow, arms, neck, eyes, shoulders and hands, which are performed quickly, a way to detect movement traces between frames allows to generate a single image representation of all movement involved on a sign. See figure 6 for details.

Token	Video	Start	End
cine	02-session-01-part-01-00.mp4	4.19	4.75
cine	02-session-01-part-01-01.mp4	1.19	1.75
terror	02-session-01-part-01-01.mp4	3.82	4.4
parque	02-session-01-part-01-03.mp4	8.97	9.3
casa	02-session-01-part-01-03.mp4	10.12	10.57
pareja	02-session-01-part-01-04.mp4	3.91	4.36
noche	02-session-01-part-01-04.mp4	4.49	4.92
noche	02-session-01-part-01-04.mp4	7.91	8.2

Table 2. Shows metadata extacted from the PSL dataset: (1)*Token* could be a noun or a number (2)*Video Path* shows the video where the token was detected (3)*Start Time* time when the token reproduction starts (4)*End Time* time when the token reproduction ends.

3.2. Opticalflow Model

The model uses a 2D CNN architecture to learn features from Opticalflow samples calculated from RGB frames using the Lucas Kanade method for features tracking. Opticalflow samples hold features tracked from an entire frames set sequentially that way all the features found across frame sets are condensed in a single image.

3.2.1 Model Architecture

The Opticalflow model architecture described in Figure 2 uses a Resnet152 backbone pre-trained with ImageNet. We used a fine tuning transfer learning approach, the backbone produces a 7x7x2048 output that then is passed to a Global Average Pooling layer for obtaining a flattened output of 1x1x2048 which is then passed to a dense layer for logits computation and finally to a softmax activation function for classes probability computation.

3.3. RGB Recurrent Model

The model uses a RNN architecture to learn features in a sequential way from RGB frames set generated by the video annotation pipeline see Figure 1. RGB frame sets hold a sequence of images representing a PSL element. We selected

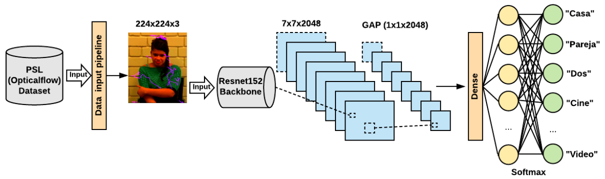

Figure 2. Opticalflow model architecture

a RNN architecture based on Natural Language Processing text based techniques that already shown good results.

3.3.1 Model Architecture

The RGB recurrent model architecture described in Figure 3 receives a sequence of decoded video frames bidirectionally where each frame set represents a PSL sample, frames were resized to 128x128 for GPU memory optimization during training decreasing considerably the number of training parameters. Frame set samples length varies on each sample requiring a layer to mask entries ensuring same length samples. We decided on using a bidirectional approach because we found benefits on learning features from left to right and right to left in the same way as text based NLP. It uses a *many-to-one* architecture with LSTM cells that hold state of 64 units length, the output produced by the recurrent layers is then passed to a dense layer for logits computation and subsequent softmax activation function for classes probability computation.

3.4. Novel Signs Detection Model (nSDm)

We propose a novel model for signs detection that ensemble the two neural networks architectures described in sections 3.2.1 and 3.3.1 with the objective to learn visual features like edges, corners and ridges (CNN) and at the same time patterns learned from a time based series of inputs (RNN) to boost the performance on detecting PSL elements. CNN network receives optical flow inputs and the RNN branch receives RGB frames extracted from the labeled PSL dataset described in 3.1.

We designed two neural network architectures for nSDm, both architectures use pre trained Opticalflow and RGB models as base models and applies different model ensemble techniques on top of them. This architectures are described in detail in section 3.4.1.

For this work we selected the Tensorflow/Keras functional API for its ability to define combined models along with a versatile data extraction and transformation layer.

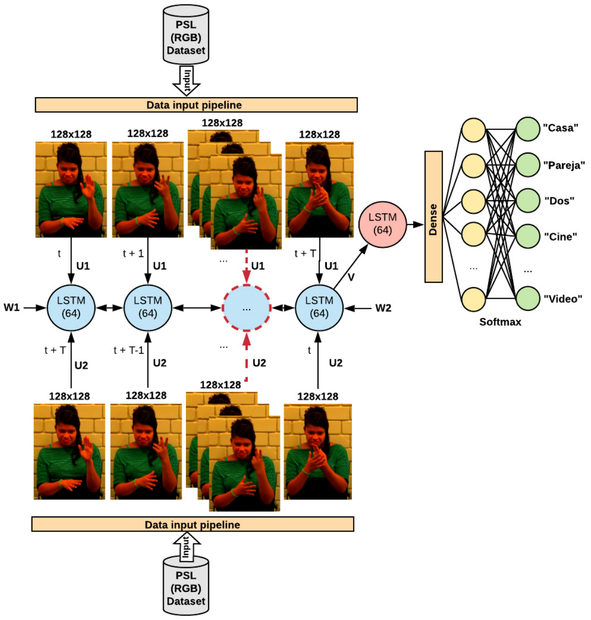

Figure 3. Recurrent RGB model architecture

3.4.1 nSDm V2 Model Architecture

Pre-trained Opticalflow and RGB recurrent models are ensemble using transfer learning with all layers freeze along with a flexible data input pipeline for data feeding, transformation and normalization.

The **Input pipeline** accesses the labeled PSL samples and applies transformations preparing the data for upper layers, transformations were applied for both Opticalflow and RGB frames, PSL Opticalflow samples were resized to be compliant with ImageNet pre-trained models using a 224 by 224 shape and three channels for color images in the other hand PSL RGB samples were resized to a 128x128 shape for GPU memory optimization, data augmentation transformations were not applied due to the nature of the experiment where samples were captured using similar light conditions and camera orientations, PSL samples were transformed to tensors and normalized to floats in the [0, 1] interval. We removed the last dense layers (classifiers) from both base models with the objective to add a single classifier in an outer layer. We concatenated the outputs and finally added a Dense layer with a softmax activation function to convert logits into probabilities used for a correct sign classification. nSDmV2 architecture is described in Figure 4.

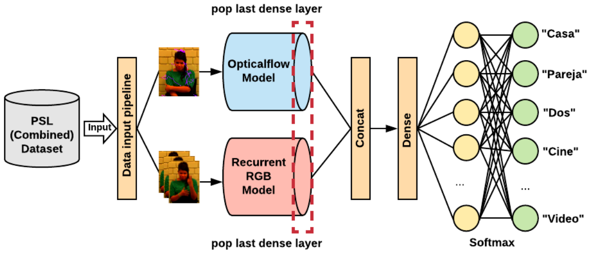

Figure 4. nSDmV2 model architecture

4. Experimentation

4.1. Dataset Description

The PSL dataset was developed by the PUCP Grammar and Signs research group in 2014 and consists in a set of videos recorded during the interviews of 24 individuals, 12 male and 12 female informants, all of them are Lima Peru residents and reported to be born with a permanent deafness condition or acquired the condition before the acquisition of Spanish.

The dataset consists in 718 video clips recorded with a ADR-CX220 SONY HD camera which included an embedded microphone. The camera focused only the informant but also recorded questions, instructions and translations.

The video clips were recorded in three sessions with the following participants: A coordinator, a PSL [2] translator and a informant.

Recording Session 1: A 45-60 minutes semi structured interview that included: Biographic information as well as habits, anecdotes, opinion about cultural subjects and elicitation of names, states and actions.

Recording Session 2: The informant was presented with a set of 55 cards describing actions and were asked to choose a set of them in order to build a coherent story that was subsequently told by the informant.

Recording Session 3: A PSL [2] conversation facilitated by the coordinator happening between the informant and the translator.

During all the sessions a PSL [2] translator performs a translation after a word or phrase is completed.

4.2. Video Annotation Results

The video annotation pipeline described on 3.1 produced an annotated PSL dataset suitable for using it in a supervised learning experiment. The annotated dataset is divided in two main parts (RGB and Optical Flow samples).

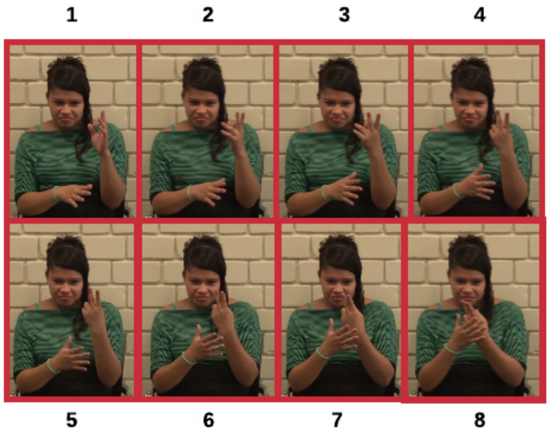

Figure 5. PSL number ”Two” RGB representation

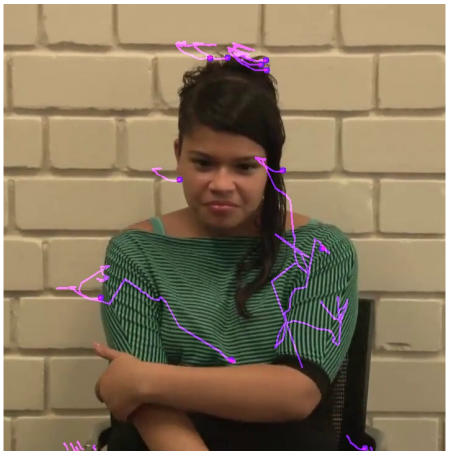

Figure 6. PSL number ”Two” OpticalFlow representation

4.2.1 RGB Annotation Results

It is a hierarchical folder structure where each detected sample is hold in a folder named with the detected noun or number containing the video frames Figure 5 shows how video frames are stored.

4.2.2 Opticalflow Annotation Results

It is a hierarchical folder structure based on the RGB samples folder structure, the Opticalflow nature of tracing movement between frames allow to produce a single image for each detected PSL combining all video frames into a single image representing the movement occurred during the sign execution, Figure 6 shows an example of an Opticalflow generated sample.

Hyper parameter	Value
Learning Rate	0.001
No Epochs	10
Batch Size	64
Shuffle Buffer Size	5000

Table 3. training hyper parameters

Name	Value
Loss Function	Categorical Cross Entropy
Optimizer	Adam

Table 4. Loss and optimization functions

Epoch	Loss	Precision	Recall	AUC
Epoch 1	3.2775	0.0000e+00	0.0000e+00	0.0900
Epoch 2	2.5090	0.0000e+00	0.0000e+00	0.1327
Epoch 3	2.0908	1.0000	0.0455	0.2616
Epoch 4	1.8167	1.0000	0.0455	0.4361
Epoch 5	1.5668	1.0000	0.1364	0.5651
Epoch 6	1.3303	1.0000	0.1364	0.7557
Epoch 7	1.1146	1.0000	0.3182	0.8657
Epoch 8	0.9199	1.0000	0.4545	0.9205
Epoch 9	0.7464	1.0000	0.6364	0.9657
Epoch 10	0.5981	1.0000	0.6818	0.9877

Table 5. Shows results of training the Opticalflow model with the 5% of the labeled PSL dataset: (1)*Epoch* identifies the epoch in in the training process (2)*Loss* obtained loss (3)*Precision* obtained precision (4)*Recall* obtained recall (5)*AUC* area under the precision-recall curve.

4.3. Sign detection results

We trained models described on sections 3.2, 3.3 and 3.4 with the 5% of the PSL dataset and validated it with the 5% of the validation PSL dataset, models were trained during ten epochs obtaining the results in Tables 5, 6 and 7.

We used the same hyper parameters while training all models. These are listed on Table 3

We used the same loss function and optimizer for all models. These are listed on Table 4

Even though models were trained with a small number of samples and are subject to over fitting, train results show patterns that indicates that performance will increase as we add more samples where metrics will become stronger as we add more samples to the input data pipeline, we are planning on processing more PSL samples as well as including PSL samples from external sources as described on section 5.

Epoch	Loss	Precision	Recall	AUC
Epoch 1	2.7568	0.0000e+00	0.0000e+00	0.0911
Epoch 2	2.6647	0.0000e+00	0.0000e+00	0.1008
Epoch 3	2.6455	0.0000e+00	0.0000e+00	0.1017
Epoch 4	2.6296	0.0000e+00	0.0000e+00	0.1193
Epoch 5	2.6121	0.0000e+00	0.0000e+00	0.1222
Epoch 6	2.6032	0.0000e+00	0.0000e+00	0.1222
Epoch 7	2.5943	0.0000e+00	0.0000e+00	0.1229
Epoch 8	2.5875	0.0000e+00	0.0000e+00	0.1229
Epoch 9	2.5825	0.0000e+00	0.0000e+00	0.1229
Epoch 10	2.5790	0.0000e+00	0.0000e+00	0.1229

Table 6. Shows results of training the RGB Recurrent model with the 5% of the labeled PSL dataset: (1)*Epoch* identifies the epoch in in the training process (2)*Loss* obtained loss (3)*Precision* obtained precision (4)*Recall* obtained recall (5)*AUC* area under the precision-recall curve.

Epoch	Loss	Precision	Recall	AUC
Epoch 1	2.9711	1.0000	0.0000e+00	0.2959
Epoch 2	2.2159	1.0000	0.0455	0.6564
Epoch 3	1.7181	1.0000	0.0909	0.8154
Epoch 4	1.3140	1.0000	0.1818	0.9340
Epoch 5	0.9585	1.0000	0.3636	0.9691
Epoch 6	0.6747	1.0000	0.5909	1.0000
Epoch 7	0.4739	1.0000	0.8636	1.0000
Epoch 8	0.3261	1.0000	0.8636	1.0000
Epoch 9	0.2263	1.0000	0.9545	1.0000
Epoch 10	0.1651	1.0000	1.0000	1.0000

Table 7. Shows results of training nSDmV2 with the 5% of the labeled PSL dataset: (1)*Epoch* identifies the epoch in in the training process (2)*Loss* obtained loss (3)*Precision* obtained precision (4)*Recall* obtained recall (5)*AUC* area under the precision-recall curve.

Train results shows the RGB recurrent model having the lowest performance with a loss equals to 2.5790 and a AUC equals to 0.1229 and Precision and Recall equals to 0.0000 which indicates recurrent models are not learning enough features. In the other hand the Opticalflow model performs better with a loss equals to 0.5981 a AUC equals to 0.9877 a Precision equals to 1.000 and a Recall equals to 0.6818 which indicates features available in Lucas Kanade Opticalflow representations are learned more effectively with a 2D CNN architecture. 2D CNN architectures show better performance than RNN architectures for detecting PSL elements.

nSDmV2 shows the highest performance reporting a loss equals to 0.1651 a Precision equals to 1.0000 and Recall equals to 1.000 and a AUC equals to 1.0000.

The results indicate ensemble models perform better than single models justifying the effort to design models that combine 2D CNN and RNN architectures. nSDmV2 shows the highest performance presumably related to the classifiers removal action applied to Opticalflow and RGB models and the subsequent concatenation which is then sent to a new classifier layer (dense layer with softmax activation) as described in section 3.4.1.

5. Discussion and Future Work

We processed the five percent of the PSL dataset with the proposed video annotation pipeline producing PSL samples for nouns and numbers using the Lucas Kanade Opticalflow representation and sequential RGB frames respectively. We trained four models described on sections 3.2, 3.3 and 3.4 obtaining results presented on section 4. Results shown over fitting due to number of samples used to train the models. As a continuation of this work we will continue processing the rest of the PSL dataset and train models to improve their robustness.

A successful supervised learning task requires a labeled dataset where samples are carefully produced and annotated. The video annotation pipeline described on section 3.1 requires a significant amount of human intervention to find video segments where signs are followed by a translation delivered after a delay factor that varies between translations. In this work we have estimated a delay factor of 3 seconds to ensure extracted frames contain the target sign but a the same time it introduces additional frames requiring human intervention to remove frames that are not relevant to the target sign. Applying self supervised learning techniques to avoid or minimize the need for human intervention while labeling the PSL dataset and other external PSL datasets available like the "Aprendo en Casa" dataset (Gisella Bejarano et al.) is the natural next step for this work where pre-trained nSDm models enriched with an auto-encoder architecture can be used to remove the need to human intervention on the proposed video annotation pipeline.

State of art on pose estimation and body expressions detection are based on key points, joints and heat maps regression. The method described in this work is a supervised learning task for signs classification, converting a classification problem into a regression one seems to be a good option that could be beneficial. Movement across frames is captured with Opticalflow showing the body parts a PSL consultant moved to emit a sign. We are looking for a method for calculating key points and joint coordinates from Opticalflow samples, caluculated key points and joint coordinates which are inputs for a 2D CNN (dowsampling) and 2D Transposed CNN (upsampling) for heat maps regression that will finally be used to detect PSL elements.

6. Conclusion

Human intervention was required for cleaning and preprocessing input videos before they can passed to the proposed video annotation pipeline. It positively affected produced samples quality because video segments containing noise and non relevant frames can be easily removed in advance. A *Delay Factor* between signs emitting and signs translation introduces noise because it varies on each produced sample requiring additional human intervention to post-process produced samples to remove non relevant frames.

Lucas Kanade Opticalflow feature tracking method successfully represented movement that occurred during signs emitting, it is important to note that when a sign is emitted many body parts are moved including arms, hands, head, neck and eyes, Opticalflow is capable to capture movement patterns for the entire body configuring an excellent tool for visual features representation in PSL elements. It is a very CPU inexpensive algorithm that can be applied as a data augmentation/transformation in data input pipelines for both training and test allowing to expand its utilization to a wide range of datasets.

Opticalflow model shown better performance than the RGB recurrent model in terms of AUC, Precision and Recall, the Opticalflow model uses a pre-trained RestNet152 base model with transfer learning (freezing) indicating that using a pre-trained base model positively affect the model performance. RGB recurrent model performance is subject to improve as we train with more PSL samples.

Ensemble models shown better performance than Opticalflow and RGB recurrent models where nSDmV2 shown the highest performance. The nSDmV2 novel architecture where pre-trained base models were popped and then concatenated allowing adding additional layers for learning features after based model were concatenated and subsequent classifier.

The area under the precision-recall curve allow measuring how well is nSDm detecting because it summarizes the trade-off between the true positive signs rate and the predicted signs.

References

[1] Xiao B., Wu H., and Wei Y. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 466–481, 2018.

[2] Dirección General de Educación Básica Especial. Guía para el aprendizaje de la lengua de señas peruana. pages 1–66, 2015.

[3] Instituto Nacional de Estadística e Informática. Primera encuesta nacional especializada sobre discapacidad. pages 41–68, 2012.

[4] Grupo de investigación y Señas (PUCP). Corpus de la lengua de señas peruana, 2014.

[5] World Health Organization. Deafness and hearing loss, Mar 2019.

[6] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks.

In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017.

[7] Tulsiani S. and & Malik J. Viewpoints and keypoints. pages 1510–1519, 2015.

[8] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013.

[9] Lei Wang, Piotr Koniusz, and Du Q Huynh. Hallucinating bag-of-words and fisher vector idt terms for cnn-based action recognition. *arXiv preprint arXiv:1906.05910*, 2019.

[10] Yi Zhu, Zhenzhong Lan, Shawn Newsam, and Alexander Hauptmann. Hidden two-stream convolutional networks for action recognition. In *Asian Conference on Computer Vision*, pages 363–378. Springer, 2018.