Segmenting Continuous Gestures

Carmen Veenker

Layout: typeset by the author using LaTeX.
Cover illustration: Unknown artist

Segmenting Continuous Gestures

Recognising Isolated Signs from Continuous Gestures using a Tube Convolutional Neural Network

Carmen Veenker
11337400

Bachelor thesis
Credits: 18 EC

Bachelor *Beta Gamma Major Kunstmatige Intelligentie*

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor
Dr. dr. A. Visser

Informatics Institute
Faculty of Natural sciences and Mathematics
University of Amsterdam
Science Park 904
1090 GH Amsterdam

July 17th, 2020

Contents

Abstract

In the world, 5% of the population has disabling hearing loss. Many of them use sign language as their main way of communication. Therefore sign language plays an important role in the daily lives of many. But, as sign language is a visual and dynamic language, it is difficult to study. Capturing sign translations and alternating between signs has been proven difficult. Moreover, large sign language data-sets are scarce, because the transcription of sign language is still handled manually. Therefore progress has been slow in this discipline. To accelerate progress in sign language research, it is imperative to construct a method to automate this transcription process.

One possible solution is founded in recent sign language recognition developments. In this thesis a network called the Tube Convolutional Neural Network (T-CNN), a method of alternating between different signs in continuous sign language sentences, will be proposed. This T-CNN combines a 3D Convolutional Neural Network with a Faster R-CNN. As a consequence this network integrates spatial and temporal data.

1 Introduction

According to the World Health Organisation (WHO) approximately 5% of the world's population has disabling hearing loss, this is around 466 million people and includes 34 million children. This number is estimated to double by 2050 [1]. A wide range of individuals suffering from hearing disabilities have difficulty learning spoken language and might use sign language as a form of communication. For example, in Britain approximately 140,000 people suffering from disabling hearing loss communicate using British Sign Language (BSL) [18]. This makes sign language play an important part in the lives of many people with hearing disabilities. However, contrary to popular belief, there is no universal sign language. Rather, there are over a hundred different sign languages and even within a particular sign language there can possibly be distinct variations and dialects [12] [2].

Furthermore, because sign language is a visually dynamic language, it is difficult to use transcription. The transcription of sign language is the process of converting sign language into a written form. This is important for two distinct reasons.

First of all, the transcription of sign language can greatly improve the ease of learning sign language. This is because, learning sign language conventionally requires an in-person teacher. Consequently, this increases the variations and dialects within each sign language, since each teacher will deviate slightly from the norm. However, if there was a consistent measure of transcription for sign language, then these transcriptions could be used to rectify the slight variations. Moreover, these transcriptions can then also be used for studying sign language, greatly diminishing the need for an in-person teacher. These transcriptions would create a reliable and reproducible set of generic rules for sign language.

Second of all, the transcription of sign language makes the analysis of data easier, since there are reliable and reproducible rules. This also makes analysing through computers possible. There are already some transcription systems in existence, such as for example, KOMVA and ELAN (both computer based)[19]. However, this process of transcribing sign language into glosses (written language) is (at the moment) not an automated task, but done manually. Consequently, transcription takes up vast amounts of time and resources and this causes slow progress in the discipline of sign language. It would be optimal if there was a model to automatise this transcription of sign language.

Automatising the transcription of sign language is a very complex problem. This problem addresses both sign language recognition (SLR) and sign language translation (SLT). And these are two very complex problems in themselves. SLR alone can be divided into three steps: *(i)* face & hand detection from the image (or video), *(ii)* extracting the manual & non-manual features and as last step *(iii)* word segmentation. SLT in turn can be divided into two steps following SLR: *(i)* word isolation and glossing, and *(ii)* sentence recognition through subjecting the isolated words to a grammar model [11].

Furthermore, the annotations –otherwise referred to as glosses –for this problem constitute an extreme multi-label classification problem. Moreover it is also a multi-modal problem, as there is a wide range of signers. Because this is such a complex problem, this research will only be focusing on one step of the SLR part of the problem.

Presently, an effective real-world application for sign language recognition (SLR) is still unavailable [4]. However, many different approaches have been attempted for making such a system, as described in the overview paper [11].

The problem starts with detecting the face and hands of the signer from the image sequence. Some signs are performed with only one hand (the dominant hand) or two hands (the dominant

and non-dominant hand). Thus, each hand needs to be detected and tracked accurately. Then, the second phase, feature extraction begins. In this phase informative feature vectors are composed to distinguish between different signs. These feature vectors will include manual signs (such as: trajectory, velocity, moment, shape or the finger spelling of the hand) and non-manual signs (such as: facial expressions, head pose or lip shape).

The next, third, step is word segmentation also called temporal segmentation. In this phase, the analysis of the building blocks of a (signed) sentence is performed. In sign language there is not always a clear boundary between words. Therefore, the movements made between the end of the last sign and the beginning of the next sign, is referred to as the movement epenthesis (ME) boundary. Extracting this ME boundary is the core of the problem of word segmentation. This is also the focus of this research.

This problem can generally be solved with two different approaches. The first (*i)* is to simultaneously attack the segmentation and recognition problem. The second (*ii)* is to detect the boundaries between the signs directly, thus achieving segmentation. For the first approach (*i)* dynamic programming and viterbi decoding have generally been utilized [3] [16]. One type of dynamic programming that was typically applied, is dynamic time warping (DTW). This method was proposed for gesture recognition; it curves the time of an observed sequence of motion with weighted body joints to pre-learned (individual) gesture sequences [3]. Moreover, in sign language, time boundary information can also be extracted through statistical sub-unit construction and decoding [16] [14]. Furthermore, as deep learning started to grow, a multi-layer bi-directional Long Short Term Memory trained with an end-to-end deep Convolutional Neural Network (CNN), was introduced. This concatenated model was then inserted into a Hidden Markov Model (HMM) for iterative refinement and eventually continuous gesture recognition [13][14]. For the second approach (*ii)* , the division of the problems into separate processes is of main importance. The first process is temporal segmentation and the second process is gesture recognition: the segmented gestures will be put through a recognition model, which will output the results [14]. For this approach, a new model has recently become popular. This model is the Faster R-CNN, which will be explained in more detail in Chapter 2.3. The Faster R-CNN model proved to produce the best results on the Chalearn Continuous gesture dataset, which is a large continuous gesture dataset. This particular model has also proven to work well for action detection and object detection tasks [10] [17]. However, this model processes spatial and temporal data separately. Because it is believed that both types of data are especially important for gesture recognition, an new model that integrates these is explored. A new adaption to the Faster R-CNN called a Tube Convolutional Neural Network (T-CNN) has adapted Faster R-CNN to a 3D Convolutional Neural Network (3D CNN), hereby achieving the integration of spatial and temporal data. This model achieves capturing motion characteristics and has proven promising results for action recognition. Moreover, the results from the T-CNN show that the T-CNN model has a high capacity for generalizing to cross-dataset action detection [10]. Therefore the T-CNN model is used in this thesis.

The research question for this thesis can be summarized as: *To which extent is it possible to accurately segment isolated words (signs) from continuous (signed) sentences with a Tube Convolutional Neural Network (T-CNN)?* The expectation is that it is possible to accurately detect the gestures, since the dataset that will be used is quite large in comparison to other datasets (see section 2.4 and 3.1 for more information). And the model has been provide accurate results above the state of the art for action detection [10].

2 Theoretic Framework

In this section the theories which are the foundation of this thesis are described.

2.1 R-CNN

The Region Convolutional Neural Network (R-CNN) is a method that combines region proposals with CNN and has been wildly popular because it introduced the fundamental concept for all modern object detection networks. The R-CNN network was first proposed to solve the problem of locating objects in an image. The R-CNN employs a region proposal algorithm to determine regions in the image that have a high probability to contain the objects that should be detected. These regions are called region proposals. For our research, the interesting regions are the regions in the image where the hand and the head of the signer can be found. Region proposal algorithms determine prospective objects in an image with image segmentation. In this form of segmentation (which is different from the temporal segmentation discussed in the introduction) images are grouped together according to their similar adjacent regions, such as and texture [6].

One such region proposal algorithm is the Sliding Window Algorithm (SLA). This is a brute force method to compute region proposals. This algorithm divides an image into several window slots, then the window consistently slides, selects and then classifies each window slot in the image with the object recognition model. Because of this the search space for the object(s) spans the entire image, where the recognition model has to search all possible locations, as well as searching using different scales. Thus, this algorithm is an exhaustive search algorithm and very computationally expensive, especially when taking into account different angles and different aspect ratios [21].
However, most other region proposal algorithms work by grouping pixel segments. This significantly reduces the amount of image slots that need to be classified. Furthermore, these generated region proposals are all of different sizes and scales.
One of the most import properties of a region proposal algorithm is to have a high recall, because the recall is high when the ground truth regions (the regions that contain the sought after objects or in the case of this thesis hand positions) and within the list of region proposals that the algorithms puts out. As a consequence region proposals algorithms also output a lot of regions that do not possess objects, also called false positives. However, even though this might negatively impact the time consumption and accuracy, this is not an issue as long as it also discovers all the true positives. This is because, most false positives are rejected in the later stage by the recognition algorithm, whereas missing true positives affects the detection rate .

One of the most commonly used region proposals is the selective search algorithm [21]. This algorithm first segments over the image based on the intensity of the pixels with a graph-based segmentation method. Second, the selective search algorithm adds bounding boxes over the segmented regions. These bounding boxes are then added to a list of region proposals. Third, adjacent segments are grouped together according to their similarity. The similarity of these segments is calculated based on colour, texture, size and shape.
Then this process is iterated. With each iteration larger segments are added to the list of proposals. This process is show in Figure 1.

Figure 1: Example selective search algorithm. Courtesy [21]

After extracting the region proposals with the selective search algorithm, the region proposals are fed into a Convolutional Neural Network (CNN). This CNN maps the feature vector for the image to a smaller vector. Because the last layer of the CNN is a fully connected layer, this network will output a feature vector of 4096 dimensions.
Finally, a SVM is used to classify the object classes form this 4096 dimensional feature vector. One SVM is used for each object class, resulting in a confidence score for the confidence that a certain feature vector represents this particular class.
Lastly, a greedy non-maximum suppression is used. This entails that all the regions are combined, except when there is an overlap between regions. Then the proposal with the higher confidence score is combined with the rest of the regions. This is referred to as the Intersection over Union (IoU). This process is repeated independently for each object class. Then all the regions that have a score over 0.5 are kept as regions. This model works fairly good for object detection, but disadvantages are still present. One of the disadvantages of this approach is that the CNN and the Region Proposal algorithm need to be trained separately. Thus, making the training not parallelizable. Also, each region proposal needs to be resized, since all proposals have different shapes whereas the state of the art CNNs only take the same size input. [6]

2.2 Fast R-CNN

The Fast Region Convolutional Neural Network (Fast R-CNN) is and adaptation of the R-CNN, that solved some of the disadvantages from the R-CNN. This resulted in a much faster application for object detection, hence the name Fast R-CNN.
First of all, the Fast R-CNN takes an 'image-centric' approach as the network takes an image and a set of object proposals as its input, instead of region proposals. Then a feature map is obtained through convolutional and pooling layers. Subsequently, a Region of Interest (RoI) pooling layer is used to acquire the region of interest (RoI) for each object proposal. This RoI comes in the form of a feature vector with a fixed length which is extracted from the feature map. Through the use of max pooling the features in a valid region of interest are mapped to a small feature map with a fixed spatial extent of $H \times W$ (H and W are layer hyper parameters and independent of all RoI). Each RoI consists of a four tuple (r, c, h, w) containing the top-left coordinate (r,c) and the height and width (h,w), these coordinates define the RoI window. During RoI max pooling, this

RoI window is divided into sub-windows, by dividing the width and height of the window into an $H \times W$ grid of sub-windows. These sub-windows should all be approximately the same size. As a next step, the values in each sub-windows are independently pooled to their respective grid cells. [5]

The Fast R-CNN network also uses backpropagation, whereas the R-CNN is not suitable for back-propagation because each RoI or proposal comes from a different image. Because of this, receptive fields of an RoI can span over a fairly large surface of the image or even an entire image. Meanwhile each forward pass is required to process this entire receptive field, thus making backpropagation highly inefficient.
Therefore, Fast R-CNN hierarchically samples mini batches for Stochastic Gradient Descent (SGD) by first sampling N images and subsequently sampling R/N RoIs for each individual image. One of the advantages of Fast R-CNN is that RoIs belonging to the same image share computation and memory in the forward and backward passes. Because of this, Fast R-CNN is opportune to a fine-tuning stage for simultaneously optimizing bound box regression and softmax classification. This is in contrast to R-CNN where this optimisation is done in separate stages.

Fast R-CNN also possesses two sibling output layers that compute the probability p with a softmax over the $K+1$ (the K object classes and a background class) of the fully connected layers for each RoI, and offsets for bounding box regression for each of the K object classes (Equation 1).

$$t^k = (t^k_x, t^k_y, t^k_w, t^k_h) \tag{1}$$

where:

k = index of each object class in K
t^k = scale-invariant translation and log-space height/width shift w.r.t the object proposals

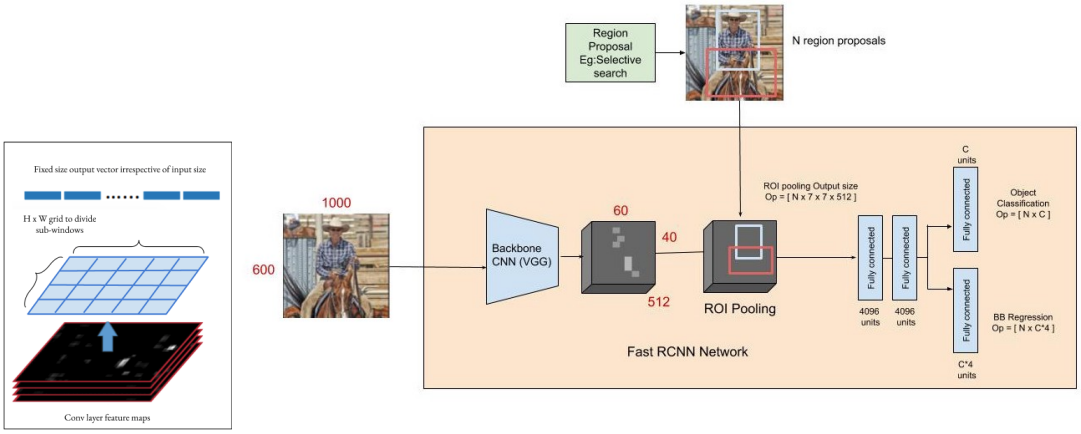
For the loss, a multi-task loss function is used, since it can simultaneously train for classification and bounding box regression for each labeled RoI. The loss function is given in equation 2. In this equation $L_{cls}(p, u) - \log(p_u)$ is the log loss for a true class u and $L_{loc}(t^u, v)$ defines the overlap between the coordinates from the ground-truth bounding box ($v = v_x, v_y, v_w, v_h$) and the bounding box predicted by bounding box regression ($t^u = t^u_x, t^u_y, t^u_w, t^u_h$).
Moreover, when the function $u \geq 1$ does not evaluate to 1, it will evaluate to zero, meaning that the current class is the background class. For this class there are no ground truth bounding boxes available. In this case L_{loc} is negated.
Moreover, for the computation of the bounding box regression the following loss function L_{loc} will be used (Equation 3):

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v) \tag{2}$$

where:

u = ground truth class from RoI
v = ground truth bounding box regression target

$$L_{loc}(t^u, v) = \sum_{i \in \{x,y,w,h\}} smooth_{L1}(t^u_i - v_i), \text{in which } smooth_{L1}(x) = \begin{cases} 0.5x^2 \text{ if } |x| \leq 1 \\ |x| - 0.5 \text{ otherwise} \end{cases} \tag{3}$$



(a) Overview of RoI Pooling layer.[8]

(b) Architecture Fast R-CNN. [5]

Figure 2: Overview Fast R-CNN

This loss function is most optimal for Fast R-CNN, because it is less sensitive to outliers. However, equation 3 is sensitive for exploding gradient, therefore equation $smooth_{L1}(x)$ is defined to solve this problem [5].

During the finetuning stage, 25% of the object proposal RoIs are that have an Intersection over Union (IoU) over 0.5 overlap with a ground truth bounding box, meaning they are one of the object from the object classes, are used. The rest of the RoIs are sampled with a [0.1,0.5] IoU overlap with the ground truth bounding boxes, these are the background objects.
After this backpropagation is used for computing the derivatives and routing them through the RoI pooling layer. The function computes partial derivatives of the loss function for each input value x_i with the help of argmax switches (equation 4)

$$\frac{\delta L}{\delta x_i} = \sum_r \sum_j [i - i * (r, j)] \frac{\delta L}{\delta y_{rj}} \tag{4}$$

2.3 Faster R-CNN

The Faster Region Convolutional Neural Network (Faster R-CNN) is an adaptation of the Fast R-CNN that that uses an object detection algorithm that does not use the selective search algorithm, but instead uses a convolutional neural network to learn the region proposals. A Faster R-CNN consists of a detection pipeline that uses a region proposal network (RPN) as its algorithm and a detector or classification network. The RPN takes in an image and retrieves prospective region proposals paired with objectness scores. This object score measures the possibility, that a region represents an object or the background.
These region proposals are found using a sliding window approach. A small network is slid over the convolutional feature map that is output by the last convolutional layer of the network. Then in

sliding window fashion a spatial window is taken from the convolutional feature map and dimension reduced, before being fed into two sibling fully connected layers for classification and bounding box regression. This sliding window fashion makes it possible for the fully-connected layers to be shared across all spatial locations. At the location of each sliding window, multiple region proposals were predicted. These region proposals are then mapped to anchors.
Anchors are reference boxes that consist of a scale and an aspect ratio. Faster R-CNN uses 3 scales and 3 aspect ratios. This means that in total each sliding window position produces 9 anchors. This method of using nine predefined anchors ensures that the anchors and functions that compute the proposals relative to these anchors are translation-invariant. This means that when an object in an image is translated, this should output the same proposal as when the image is used to generate the proposal. This property reduces the model size.
Furthermore, because the anchors are based on a pyramid of anchors method, they are more computationally efficient. Bounding boxes are classified and regressed with anchor boxes with different scales and ratios as references. The model uses references of a single scale and sliding window filter with ratios of a single size. This also makes it possible to compute convolutional features on a single scale image. Since, the detection network uses single scale images, this multi-scale anchors method is crucial for sharing features between the different layer without adding extra costs due to scaling.

Each anchor is assigned a binary class label, either being an object or not. Then the anchors with the highest Intersection-over-Union (IoU) overlap with the ground truth box and the anchors that have an IoU overlap higher than 0.7 with any ground-truth box are assigned a positive label. The anchors that have IoU overlap lower than 0.3 corresponding to all ground-truth boxes are assigned a negative label. The loss function is:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls)}} \sum_i L_{cls}(p_i, p_i*) + \lambda \frac{1}{N_{reg}} \sum_i p_i * L_{reg}(t_i, t_i*) \tag{5}$$

And the functions for parameterization are:

$$\begin{aligned} t_x &= (x-_a)/w_a, & t_y &= (y-y_a)/h_a, \\ t_w &= log(w/w_a), & t_h &= log(h/h_a), \\ t_x* &= (x*-x_a)/w_a, & t_y* &= (y*-y_a)/h_a, \\ t_w* &= log(w*/w_a), & t_h* &= log(h*/h_a), \end{aligned}$$

The clear contrast between Fast R-CNN and Faster R-CNN is that Fast R-CNN pools features from randomly sized RoIs for bounding box regression, shares the weights of these regressions over all region sizes, whereas the Faster R-CNN uses same sized features for regression.
Furthermore, bounding box regressors are learned to represent the for the different anchor sizes. As a consequence, boxes differentiating in size can be detected whilst the features remain of fixed scale.
The 'image-centric' sampling strategy that is used by Fast R-CNN is also used by this model. At the same time, the Faster R-CNN is an end-to-end network with the help of back propagation and stochastic descent. Each mini-batch includes an image with an even amount of 256 randomly sampled positive and negative anchors. The mini-batch is padded with negative anchors when there are less than 128 positive samples in an image. For the detection a Fast R-CNN is used that

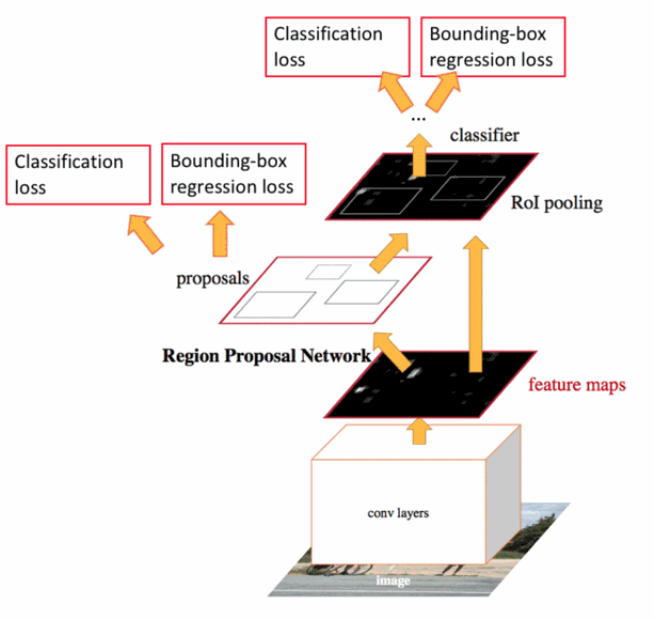


Figure 3: Architecture Faster R-CNN. [15]

uses the proposals found by the RPN as its input. Since the RPN and Fast R-CNN modify their convolutional layers differently when trained separately, two different methods for sharing these features were applied. The first devised method is called *Alternating training*. For this method the RPN is first trained and then the Fast R-CNN is trained on the output proposals from the RPN. Then the tuned weights of the Fast R-CNN are used to initialize the RPN a second time. After this the proposals obtained after the new fine-tuning are used for the Fast R-CNN. The second method is *Approximate joint training*. The two networks are merged into one network. And each forward pass generates region proposals and treats these as precomputed proposal samples through the network. Whereas during back propagation the backward propagated weights are the combination of the RPN loss and Fast R-CNN loss. Although this is an easily implementable method, it is also approximate [17].

2.4 Datasets

There are two main approaches that could be taken to creating datasets for SLR. The device based approach and the vision based approach.
The device based approach uses devices, such as data gloves, power gloves, cyber gloves and dexterous master gloves. The vision based approach uses coloured gloves or bare hands (normal gloves with different colour for each finger and the wrist).
Device based approaches are overall very accurate, but what they win in terms of accuracy they lose in applicability. Because using this approach for SLR in real life applications, would entail the signer to bear the device with them at all times. Furthermore, most of these devices are very expensive, so not every signer would be able to have such expensive gear. On the other hand, the

bare hand techniques are low cost, realistic and more mobile (no connection to any device). However, these techniques are computationally very expensive. Consequently, the response time will increase causing this technology to not be usable in real-time either. In summary, each approach has its pros and cons.
Most datasets nowadays use the vision based approach for conjuring up the dataset. However, the biggest problem with most SLR datasets is that their size is fairly small and fairly homogeneous, especially for continuous SLR. Therefore, this thesis will be focusing on a Continuous gesture dataset called Chalearn ConGD . This dataset is one of the largest multi-modal video datasets for gesture recognition (GR). The Gesture dataset includes many different gestures, including sign language gestures, but also gestures used in human communication outside the sign language community. However, because they use such a wide variety, they are more easily generalisable. Table 1 shows some of the mainly used Dataset for GR and SLR, and their respective quantities.

Name	Samples	Subjects	Language
RWTH-PHOENIX weather	1980	9	GSL
RWTH-BOSTON	200	2	ASL
NGT corpus	2000	-	NGT
Chalearn ConGD	22535	21	-

Table 1: Overview of different datasets

2.5 Related Work

Wang et al. used a Convolutional Neural Network (CNN) with depth maps to tackle this problem. They acquired segmentation by using Quantity of Movement (QoM) and then estimated compact representations for depth sequences by the name of Improved Depth Motion Map(IDMM). This representation converts each depth sequence into an image and then achieves the recognition task with the assistance of CNNs [23]. This achieved results above the baseline [24]. The best results to date, however, are results accomplished by Liu et al.[14].

The research from Liu et al follows the second approach *(ii)* to temporal segmentation. In their research the Continuous Gesture Dataset (ConGD) is used (this dataset will be further explained in section 3.1.1). The data from this dataset contains RGB-D videos and Depth videos. Liu et al, first performed temporal segmentation and data fusion on this dataset by using precise hand detection. For this hand detection they first align the depth pixels of the images to the colour coordinate space. This is done by comparing the mapping relationship between the colour and the depth coordinate spaces. A technique called camera calibration is hereby utilised. Then the hand detection is performed by a two-stream Faster R-CNN (further information on this network in section 2.2.2). First a region proposal network (RPN) is used to generate significant regions of interest (ROI). Then bounding box regression will be done for each ROI with a Fast-RCNN. After this a non-maximum suppression is performed independently for each class.

For the temporal segmentation a fixed height threshold is used on the hand-positions to establish the boundaries of a gesture. When the hand-positions are above this height, a new gesture has begun and when they are beneath this threshold the end of a sign has been reached. This method only works under the assumption that a signer raises his hand when starting a sign and putting

it back down when ending a sign. Finally the specific spatio-temporal feature maps of both the RGB and aligned depth videos are the gesture representation for each isolated gesture are extracted with 3D convolutional neural network (3D CNN). This 3D CNN only accounts for the regions and face location in each frame that were detected by the Faster R-CNN. This is done to decrease the influence of meaningless distractor regions, such as the background, clothing or body.
The concatenation of these feature maps is considered the final feature representation.
Then the fused features from the RGB and depth channels were provided to a SVM classifier to perform gesture classification.This achieved a Mean Jaccard Index of $0,6103$ (on the test set).

3 Method and Approach

In this section the methods and approach taken in this thesis are described.

3.1 Datasets

A handful of datasets for continuous sign language recognition is available. However, most of these datasets are relatively small and relatively homogeneous. Sign language datasets that include multiple signers are also scarce. Moreover, most of these datasets do not include temporal segmentation labels, which is a crucial part of this research. Therefore, this paper does not use a sign language datasets, but rather gesture datasets. These datasets are the Chalearn LAP RGB-D isolated Gesture Dataset (IsoGD) and Continuous Gesture Dataset (ConGD). THey are two large multi-modal video dataset designed for gesture recognition. These datasets are the largest RGB-D gesture dataset regarding gesture numbers and classes in comparison to other state-of-the-art datasets [24]. The data is captured with the Kinect camera, providing both RGB and depth images. This data is more consistent to real life applications, because it does not need any expensive equipment, such as data gloves or 3D trackers. It also uses the bare hands approach, which is more realistic.

3.1.1 Chalearn ConGD Dataset

The Chalearn ConGD dataset is derived from the Chalearn Gesture Dataset (CGD), which contains over 54,000 gestures which are split into sub-tasks. A subset of these gestures was temporally segmented manually to capture the start and end frames for each gesture. In total there are 249 different gesture labels and each video can represent one or more of these labels. The dataset consists of 47933 RGB-D gestures in 22535 RGB-D gesture videos presented by 21 different individuals. The focus for this dataset was on "large-scale" and "user independent" learning.For each class there are over 200 RGB and depth videos and training samples from the same individual. They are situated in either the training, validation or test samples and not mixed among all three [24]. An overview of this dataset can be found in table 2.

3.1.2 Chalearn IsoGD Dataset

This dataset is created in the same way as the Chalearn ConGD Dataset, the only difference is that this dataset only contains the isolated gestures and no continuous gestures. In total, this dataset contains 47933 RGB-D gesture videos. One for each of the individual gestures of the ConGD dataset. This dataset is also divided into a training, validation and test set, and they have the same prerequisites as the ConGD dataset [22].

Sets	Labels	Gestures	RGB videos & Depth videos	Signers
Training	249	30442	14314	17
Validation	249	8889	4179	2
Test	249	8602	4042	2

Table 2: Overview of Dataset splits for ConGD

3.1.3 Preprocessing

The T-CNN model requires a fixed input size for the images. Therefore, but also due to memory constraints each Isolated gesture video is split into eight frames image files using interpolation. Then each frame is filtered with a Gaussian filter. During this process, a couple of videos were removed, as they were only one frame long.

Furthermore, the model also requires a set of bounding boxes. These were not present in either the IsoGD dataset or the ConGD dataset. However, the research from Wan et al., had already performed bounding box regression with a Faster R-CNN [24]. These bounding boxes mainly covered face and hand regions. Therefore, these bounding boxes are used as the ground truth bounding boxes in the model and they are also used to determine the anchor boxes for the model. The bounding box frames were matched with the eight interpolated frames, to get the accurate bounding boxes for each image. However, not all frames possess a corresponding bounding box. This is because, the regions are not found, or the frame does not contain any regions (this happens for example when the hands are resting to the side of the body). When there is no bounding box available for a frame, instead a zero box is used as ground truth bounding box.

However, if the frame does contain a region, but this region is not found, and therefore the ground truth bounding box for the frame is padded with zeros, a significant part of the data is lost. This might cause interference. Therefore, especially when the original video is long, it is important to consider bounding boxes from an earlier and later frame.

Thus, before padding a bounding box with zeros, it is first checked whether one earlier frame one and later frame also miss their bounding boxes. If they both miss the bounding boxes, the ground truth box is padded with zeros. But if either one does contain a bounding box, then this box is used as the ground truth box for the frame.

For the ConGD the videos were also split into eight frames and filtered with a Gaussian filter. The videos that were only one frame long were also removed. However, there were no bounding boxes available for this dataset, neither has any bounding-box regression ever been performed on this dataset.

Since the ConGD dataset is in theory created from the IsoGD, the bounding boxes from the IsoGD were used to create the bounding boxes for the ConGD. This was done by matching each label for a video from the ConGD dataset to videos in the IsoGD dataset with the corresponding label. Then the temporal labels of the ConGD video are used to determine which bounding boxes needed to be selected from the bounding boxes of the corresponding IsoGD video. The temporal labels define the interval for the frames of the bounding boxes that need to be selected. This is then done for each of the labels/intervals of the (ConGD) video. These bounding boxes are then concatenated to form a new bounding box. After this, the bounding boxes are interpolated using the same method as used for the IsoGD bounding boxes.

The new bounding boxes will not be perfectly aligned with the images, but this should not pose a significant problem, since bounding box regression is also performed by the model.

3.2 Model

The Tube Convolutional Neural Network (T-CNN) will be used as the model for this problem. An overview of the T-CNN architecture is found in figure 4. This model proved to be a superior model for action detection and segmentation [10]. Moreover, this model is an adaptation of Faster R-CNN, described in Section 2.3. It was adapted towards a 3D Neural Network as to preserve both the temporal and spatial data, and because 3D CNN is able to capture motion characteristics and has had promising results for action recognition problems. Therefore, it combines the idea of Faster R-CNN with the descriptive power of a 3D Convolutional Neural Network .

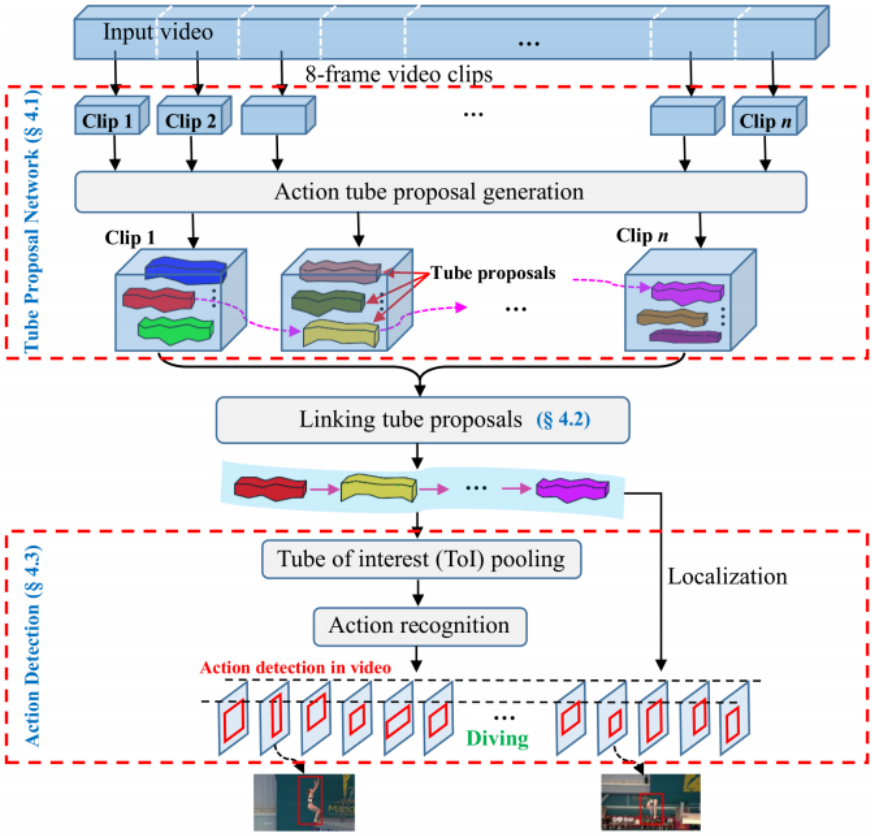


Figure 4: Overview of the T-CNN model. [10]

Other than adaptations such as the two-stream networks- widely applied networks in which spatial and temporal data are processed separately, a 3D Convolutional Neural Network (on which their model is based) integrates the temporal and spatial data. Because of this the T-CNN was able to procure better results than the popular two-stream network pipeline for action detection and video object segmentation. Therefore it is believed that this model will also procure better

results than the state of the art and two stream networks for gesture segmentation.

The T-CNN uses a Region Proposal Network in combination with a detection network, just like the Faster R-CNN. However, the T-CNN generalises this model to a 3D approach, so it is able to classify videos instead of images (An overview of the T-CNN architecture is found in figure 4).

Videos differ widely in their temporal dimension, therefore each video is split into a fixed range of eight frames. This makes it possible for the videos to be processed by a fixed size 3D CNN architecture.

The T-CNN treats action detection as a binary (e.g. foreground action and background) video segmentation task. T-CNN utilises bounding-box regression to refine a bounding box around a person. This is presumed to trump models for our problem, since similarly to action detection, the silhouette and movement of the silhouette is most important for our task.[9]

3.2.1 ToI Pooling Layer

To produce a fixed length feature vector of the tube proposals used for the prediction of the labels and localisation, a new pooling layer is introduced. This is the Tube of Interest (ToI) pooling layer, illustrated in Fig. 5.

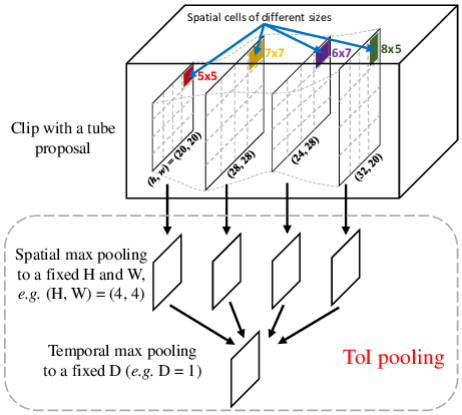


Figure 5: Overview ToI Pooling. [10]

This layer is a 3D generalisation of the RoI pooling layer of the R-CNN. However, unlike RoI pooling, for ToI pooling the output shape is fixed first before determining the kernel size and stride in reference to the output shape. ToI pooling uses feature cubes with a size of $d \times h \times w$ (where d, w and h are the depth, width and height respectively) and 3D tubes, that contain a $d \times 4$ matrix with d proposal boxes. These proposal boxes consist of a four tuple $(x_1^i, y_1^i, x_2^i, y_2^i)$, in which i is the index for the feature map and the x's and y's are coordinates of the top left and bottom right corners of the proposal box. It is possible for the proposal boxes to vary in size, aspect ratio and position. Therefore, spatial and temporal pooling are then performed separately.

Similarly to RoI pooling in the Fast R-CNN, feature maps are first divided into $H \times W$ sub-windows, by dividing the width and height of the window into an $H \times W$ grid of sub-windows. These sub-windows should all be approximately the same size. Then the max values in each sub-windows are independently pooled to their respective grid cells through max-pooling. Second, the spatially pooled depth feature maps are divided into D sub-windows, also all approximately

the same size (d/D). Subsequently, as with the spatial pooling, the adjacent feature maps are grouped together for temporal max pooling. This results in a Tube of Interest (ToI) with the size $D \times H \times W$.

The network also backpropagates gradients through the ToI pooling layer. This routes the derivatives from the output to the input. The computation for this is shown in equation 6, in which x_i is the index for the activation of the ToI pooling layer and $f(j)$ represents the argmax selection function from the ToI.

$$\frac{\delta L}{\delta x_i} = \sum_j [i = f(j)] \frac{\delta L}{\delta y_j} \tag{6}$$

3.2.2 Tube Proposal Network

The Tube Proposal Network (TPN) produces tube proposals for each clip. These tube proposals are then linked together and used for temporal segmentation of continuous gestures. This TPN architecture (see Fig. 6) was inspired by the work of Hou *et al.*

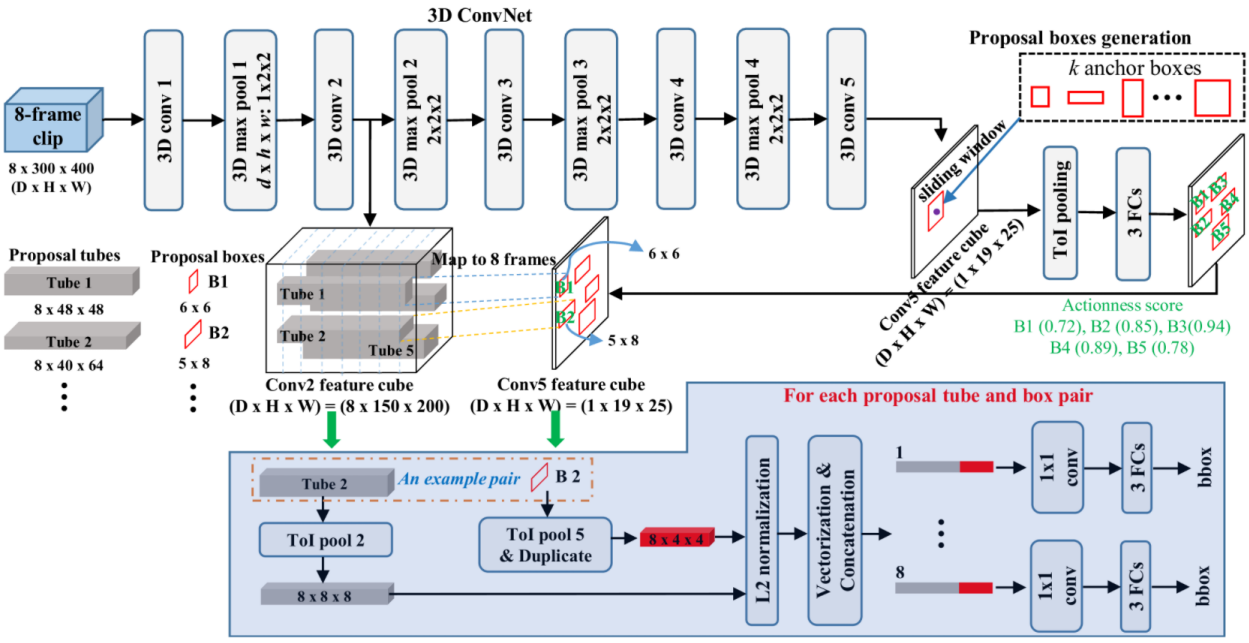


Figure 6: Overview of the TPN network from Hou *et al.* [10]

The C3D model is used to pretrain the model.

3.2.3 Anchor Boxes

Unlike with the Faster R-CNN where the bounding box dimensions are manually modelled, the T-CNN uses a k-means clustering algorithm to determine the "best" 12 anchor boxes within the training set. Then each bounding box is assigned an 'actionness' score, much like the 'objectness

score' for the Faster R-CNN. This score entails the probability that the bounding box represents a valid action. According to this score the bounding boxes will be given a binary label, either being an action or not. Moreover, depending on the height of this score (a certain threshold) the bounding box will be discarded or kept. During the training phase, the bounding boxes are also divided into positive bounding boxes and negative bounding boxes. As is done in an R-CNN, when the IoU overlap with any ground truth bounding box is higher than 0.7 or the bounding box has the highest IoU overlap with a ground truth bounding box, it is defined as a positive bounding box proposal.

3.2.4 Temporal Skip Pooling

In this model temporal max-pooling is used to assist the discovery of bounding box proposals. However, this also reduces the temporal dimension from eight frames to only one frame, meaning that the temporal order of the frames is lost. To counter this, skip pooling is used to interject the temporal order for frame-level detection.
Temporal skip pooling is performed by mapping each positive bounding box collected through ToI pooling to a feature cube from the second convolutional layer. This feature cube possesses eight feature slots, corresponding to the original eight input frames.

A set of tube proposals is obtained for each video clip after the TPN. For example, if there are three positive bounding box proposals found in the feature cube from the fifth convolutional layer, each of these bounding box proposals is mapped to each of corresponding slots of the second convolutional layer and at the right location within that slot. This results in three tube proposals, which are then paired with corresponding box proposals to facilitate frame-level gesture detection. Then ToI pooling is applied to the tube proposal and the bounding box proposals. However, the tube proposals spanning eight frames is reduced to one frame with the ToI pooling, so the resulting bounding box is duplicated eight times to form a new tube proposal. An L2 normalization is applied and the tubes are pained, vectorised and concatenated. Finally the results of the normalisation are fed through three fully connected layers, which provide a matrix containing the height, width and coordinates of the center for all the bounding boxes on each frame.

The regression loss of a tube proposal is the sum of the difference in height,width and the center coordinates between all the bounding box proposals and the ground truth bounding boxes.

3.2.5 Linking tube proposals

The tube proposals that are obtained must be linked together before spatio-temporal gesture localization can be performed on the entirety of the video. So the tubes need to be linked together. However, not all tube combinations paint an accurate picture, as some tube proposals may catch an action whereas other tube proposals might only encounter the background. Ideally, the selected tube proposals all describe a gesture and intuitively the consecutive tube proposals possess a significant temporal overlap between them. Therefore, the tubes are linked with an equation that takes both these properties into account, see Equation 7. The the possibility of an action taking place within the tube proposal is measured with an *actioness score*.

$$S = \frac{1}{m}\sum_{i=1}^{m} Actioness_i + \frac{1}{m-1}\sum_{j=1}^{m-1} Overlap_{i,j+1} \tag{7}$$

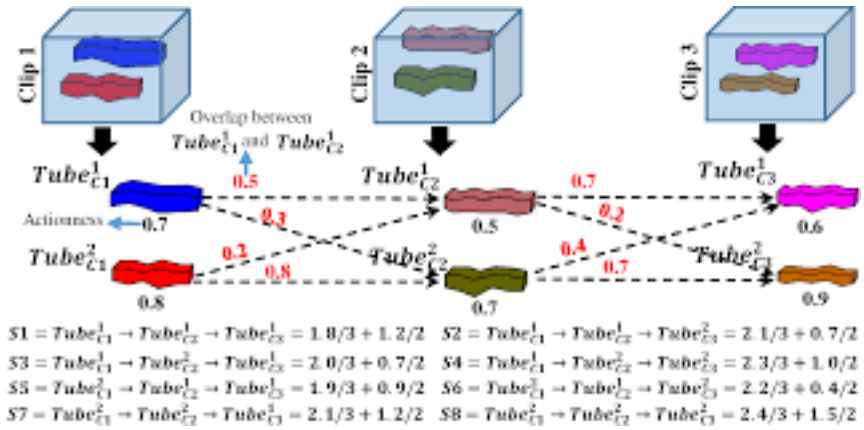


Figure 7: Example of linking tube proposals. [10]

The overlap between two consecutive tube proposals is measured with an IoU, which is also used in Fast R-CNN. Multiple linked proposal tubes in a video exhibiting the highest scores are then chosen for gesture detection. An example of linking tubes and calculating their scores can be found in Figure 7.

3.2.6 Gesture Detection

After linking the tubes proposals together, the gestures need to be classified. However, the linked tube proposals, may differ in size. Therefore, ToI pooling is used to extract a fixed length feature vector from each linked proposals. After this, the new fixed length feature vector is fed to the detection model. For the IsoGD the action detection model from Hou et al. was used. However, since the ConGD is a multi-label dataset, the model was adapted to this by changing the ending layer to sigmoid and cross-entropy layer, to calculate the loss for this dataset. Furthermore, the labels were inserted as a one-hot vector. The dimension of the last fully connected layer turned out to be N + 1 (for the N classes and one background class). See table 3 for an overview of the new architecture Table 3.

3.3 Experiment

For the experiment, first of all transfer learning will be utilised because research has shown that transfer learning can significantly increase the accuracy of gesture classification even if the model was pre-trained on a different task (e.g. action recognition) [7]. Therefore, the model is first initialised with the 3D Convolutional Neural Network from Tran *et al.* [20]. This network has been trained on an action recognition task. Then the network was pre-trained on the IsoGD, the hypothesis is that this familiarizes the network with the individual sign, making it easier to detect the ME. Afterwards, the network is trained on the ConGD and the Data is classified with a multi-label classifier.

Then the results are be evaluated with the Mean Jaccard Index (MJI). For training the TPN and detection network alternating training is used, this is described in section 2.3. First the pre-training is done by training the TPN network with the IsoGD, then the detection network is initialised with

name	kernel dims	output dims
conv1	$3 \times 3 \times 3$	$64 \times 8 \times 300 \times 400$
max-pool1	$1 \times 2 \times 2$	$64 \times 8 \times 150 \times 200$
conv2	$3 \times 3 \times 3$	$128 \times 8 \times 150 \times 200$
max-pool2	$2 \times 2 \times 2$	$128 \times 4 \times 75 \times 100$
conv3a	$3 \times 3 \times 3$	$256 \times 4 \times 75 \times 100$
conv3b	$3 \times 3 \times 3$	$256 \times 4 \times 75 \times 100$
max-pool3	$2 \times 2 \times 2$	$256 \times 4 \times 75 \times 100$
conv4a	$3 \times 3 \times 3$	$512 \times 2 \times 38 \times 50$
conv4b	$3 \times 3 \times 3$	$512 \times 2 \times 38 \times 50$
max-pool4	$2 \times 2 \times 2$	$512 \times 1 \times 19 \times 25$
conv5a	$3 \times 3 \times 3$	$512 \times 1 \times 19 \times 25$
conv5b	$3 \times 3 \times 3$	$512 \times 1 \times 19 \times 25$
toi-pool	-	$128 \times 8 \times 8 \times 8$
fc6	-	$512 \times 1 \times 4 \times 4$
fc7	-	4069
sigmoid	-	4069
fc8	-	250
loss	-	1

Table 3: Detection network architecture of this research

the resulting weights from the TPN training. Subsequently, the detection network is trained. The weights achieved by the this network are then used to initialise the TPN with ConGD dataset. After this, the the new weights from the TPN and ConGD are used to initialise the final detection model. This is called alternating training (for more information see section 2.3).

Then the model in fine-tuned. Figure 8 shows the learning curve from the model, the x axis represents the number of epochs the model has trained for and the y axis represents the loss function.

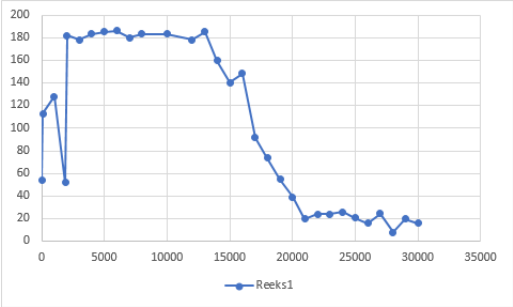


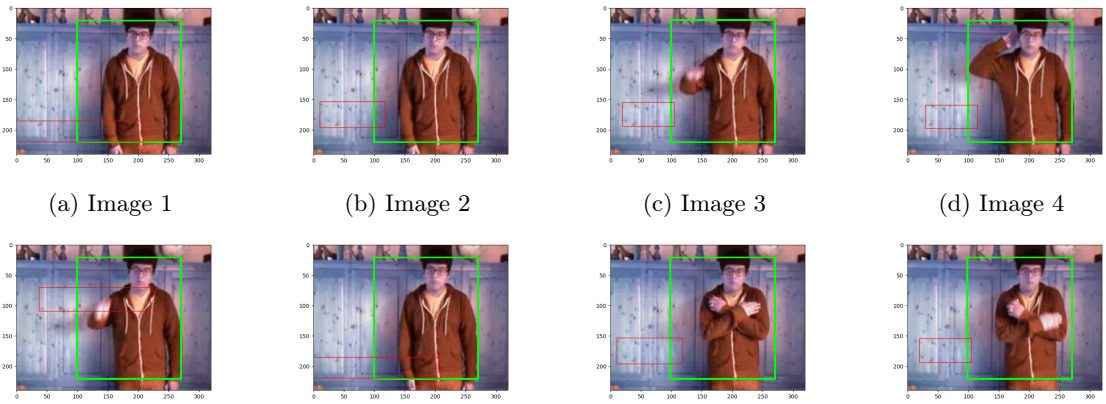
Figure 8: The learning curve from training

3.4 Evaluation

The evaluation will be done with the Mean Jaccard Index (MJI), as this is also the evaluation used for the benchmarks. Consequently, the result will be evaluated against the benchmarks.The segmentation error will also be measured. The Mean Jaccard Index is provided as the performance measurement for different methods. The Mean Jaccard Index measures the average relative overlap between predicted and ground-truth labels for all given continuous videos.

4 Results

In Figure 9 and 10, the bounding box regression is shown on two videos (each eight frames long. Figure 9 shows a video that was labeled right and figure 10 shows a video that was labeled wrong. The red box in each frame represents the ground truth bounding box, whereas the green box represents the new bounding box that was found by regression. From these images it is clear that

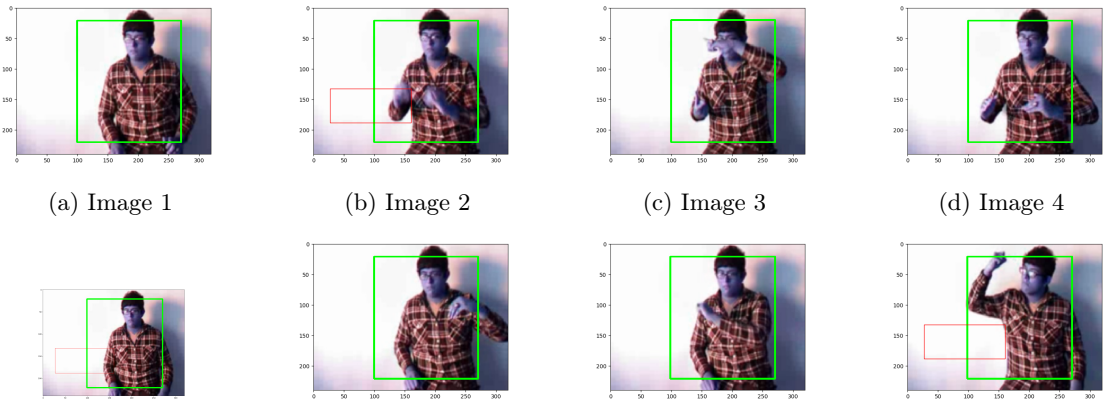


(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4

(e) Image 5 (f) Image 6 (g) Image 7 (h) Image 8

Figure 9: Bounding box regression performed on the rightly interpreted video

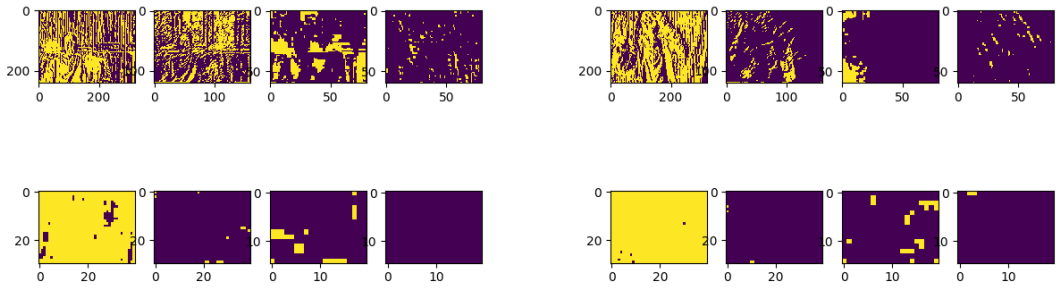
the truth bounding boxes, as expected, were not completely accurate. This is because the dataset itself did not provide bounding boxes and neither had bounding box regression ever been performed on this dataset. Therefore, it was decided to create these bounding boxes from existing bounding boxes that were created for the ISO GD, which contains the same videos, but isolated. Although



(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4

(e) Image 5 (f) Image 6 (g) Image 7 (h) Image 8

Figure 10: Bounding box regression performed on the misinterpreted video



(a) Image 1.

(b) Image 3.

Figure 11: Convolutions of misinterpreted video

the ground truth bounding boxes were not completely accurate, the new bounding boxes are fairly accurate, as they envelop the body of each signer. However, they are not very exact and target a big area on each image. This is likely due to transfer learning. As the anchors and bounding regression done by the action dataset created boxes that are similar to the boxes seen here. The bounding boxes for action detection needed to envelop the whole person, to accurately classify the action, but for gesture recognition a more accurate bounding box is more beneficial (f.e. a bounding box that only includes the hands). The lacking of the ground truth bounding boxes can also be attributed to the loss in precision. After the bounding boxes were found, the classification was done. The results

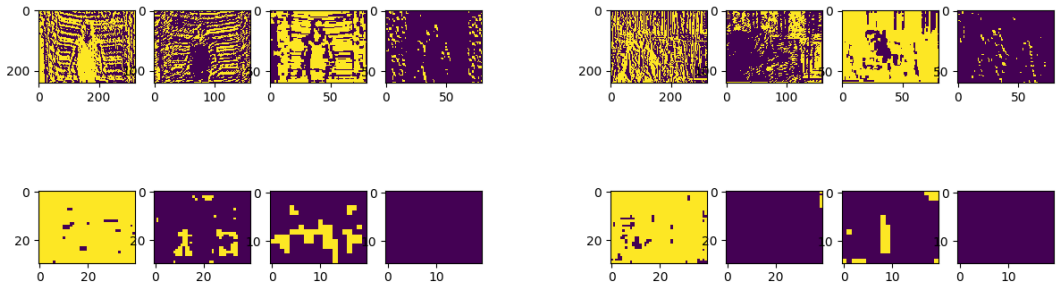
Name	Validation set
ICT_NHCI	0.516
AMRL	0.5957
PaFIFA	0.3846
Deepgesture	0.319
This study	0.00554

Table 4: Caption: Overview of results from CON GD classification in comparison with benchmarks

of the classification for the CON GD are shown in Table 4. And the results for classification for the ISO GD are shown in Table 6. The results for both these dataset were very poor, even more so in comparison with the benchmarks. One of the reasons for this development becomes clear when looking at the convolutional layers.

The convolutional layers are shown in figure 12 and 11. In figure 12 the layers of a video that is classified right is shown, whereas in figure 11 the convolutional layer of a misinterpreted video are shown. The network has eight convolutional layers, therefore eight pictures are shown. From left to right the first picture is from the first convolutional layer and the last picture is from the last convolutional layer.

The images show that the first, second and third convolutional layer mainly look at edges. The



(a) Image 1.[8]

(b) Image 3. [5]

Figure 12: Convolutions of rightly interpreted video

first looks for inside edges and the second looks for the inverted edges. The third convolutional layer looks for more broader edges. The biggest difference between these two sets of pictures is in the later layers, where for the rightly interpreted video there is a more clustered and definitive area in the place where the hand is 1211, the misinterpreted video shows more scattered, detached and smaller areas which could resemble the hand. Although all the earlier layers show some sign of results, the last layer remains completely dark in almost all of the cases. This, combined with the attribute of the labels, explains why the Mean Jacquard Index is significantly low. Within the convolutional layers, the last layer is in almost all the cases completely dark. This signifies the fact that the model chooses the negative labels as they are less costly. The reason for this is that in

True Positive	True Negative	False Positive	False Negative
0.2867400670177118	211.36285303973193	36.71325993298229	1.6371469602680708

Table 5: Average True Positives, True Negatives, False Positives, False Negatives

total there are 250 labels for each video. However, the maximum amount of truth labels that a video in this dataset has is five. This means an overwhelming bias towards negative labels, since the data is represented as a one-hot-vector. Therefore, an overpowering amount of labels will always be interpreted as a negative label, because it costs less for the model to always choose to make labels negative, instead of entertaining the idea of a positive label.

In 5 the average amount of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) are presented. As expected the amount of TNs that gets classified far outweighs the other categories. However, on average there is also a significant amount of False Positives that is mis-classified. This can also be attritbuted to bias. However, this seems to be a bias within the dataset itself. Figure 13 shows the amount of times the labels are present in the dataset. As shown in the graph, there is a small amount of labels that is significantly more present within the dataset. This is probably the cause for the significant amount of FPs that are seen in Table 5.

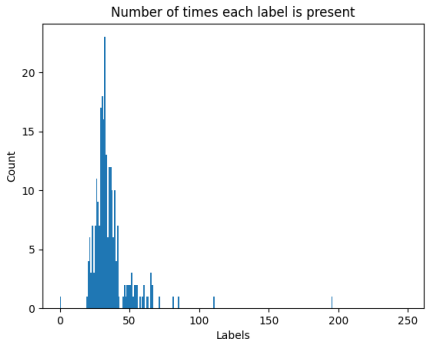


Figure 13: Number of times each label is present.

5 Conclusion

In conclusion, although the TPN presented us with fairly accurate bounding boxes from regression, the results are sufficiently sub-par. There is a bias in the positive versus the negative labels, because of the label representation for this multi-label problem. And there is also a bias in the data itself.

Name	Validation
ASU	64.4
SYSU_ISEE	59.7
Lostoy	62.02
AMRL	60.81
XDETVP	58.0
This study	0.00456

Table 6: Overview of results from ISO GD in comparison with benchmarks

6 Discussion

Although the results were very poor, the bounding boxes were fairly accurate. This is the first time that bounding box regression has been performed on the full ConGD, therefore the results could be beneficial for further research. Furthermore, the bounding box regression performed by earlier research proved to be slightly inaccurate. The ground truth bounding box that was used, was originally acquired for the IsoGD. However, even for the IsoGD the bounding boxes were not fairly accurate. Therefore, even though the bounding boxes span quite a large space, they always contain the signer, whereas the original bounding boxes created by Wang et al. sometimes only contained the background [24]. Therefore, the new bounding boxes may be beneficial for future research.

Furthermore, most of the benchmarks did not procure the segmentation of the data through models, but rather they segmented the data themselves with the temporal labels provided and then started training with this segmented data. Because this research did temporally segment the data through a model, instead of using the labels to segment the data, this may provide insight into the continuous gesture segmentation.

Another strength of this research are the convolutional layers. The convolutional layers show that the regions of interest are being learned and segmented. However, because of the bias in our data it was not possible to create significant results.

Furthermore, due to memory and time constraints, it was not possible to train the model for a very long time. To create better results, the time spent training the model should be increased. Moreover, due to memory constraints the videos had to be divided into eight frames. However, the original videos were all around forty frames in size. Thus, significantly reducing the time dimension of the data may have also created a loss in the data.

The solution for the problem of bias would be to sufficiently augment the data, to create a more balanced dataset. Either through position augmentation, color augmentation or deleting some of the data. Another improvement could be to either downsize the amount of labels available or create a different manner of data representation from the one hot vector that was used in this research. This would considerably improve the research.

References

[1] S. K. Aremu, "Deafness and Its Burdens-Prevention and Control", *International Journal of Innovative Research in Medical Science*, volume 4(04):pp. 268–to, 2019.

[2] R. Bayley, A. C. Schembri and C. Lucas, "Variation and change in sign languages", *Sociolinguistics and Deaf Communities*, volume 61:p. 94, 2015.

[3] S. Celebi, A. S. Aydin, T. T. Temiz and T. Arici, "Gesture recognition using skeleton data with weighted dynamic time warping.", in "VISAPP (1)", pp. 620–625, 2013.

[4] M. Ebrahim Al-Ahdal and M. T. Nooritawati, "Review in Sign Language Recognition Systems", in "2012 IEEE Symposium on Computers Informatics (ISCI)", pp. 52–57, 2012.

[5] R. Girshick, "Fast R-CNN object detection with Caffe", *Microsoft Research*, 2015.

[6] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 580–587, 2014.

[7] P. M. X. Y. S. Gupta and K. K. S. T. J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks", in "CVPR", 2016.

[8] K. He, X. Zhang, S. Ren and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition", *IEEE transactions on pattern analysis and machine intelligence*, volume 37(9):pp. 1904–1916, 2015.

[9] R. Hou, C. Chen and M. Shah, "An end-to-end 3d convolutional neural network for action detection and segmentation in videos", *arXiv preprint arXiv:1712.01111*, 2017.

[10] R. Hou, C. Chen and M. Shah, "Tube convolutional neural network (T-CNN) for action detection in videos", in "Proceedings of the IEEE international conference on computer vision", pp. 5822–5831, 2017.

[11] N. B. Ibrahim, H. H. Zayed and M. M. Selim, "Advances, Challenges and Opportunities in Continuous Sign Language Recognition", *Journal of Engineering and Applied Sciences*, volume 15(5):pp. 1205–1227, 2020.

[12] L. Jing, E. Vahdani, M. Huenerfauth and Y. Tian, "Recognizing American Sign Language Manual Signs from RGB-D Videos", *arXiv preprint arXiv:1906.02851*, 2019.

[13] O. Koller, S. Zargaran and H. Ney, "Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 4297–4305, 2017.

[14] Z. Liu, X. Chai, Z. Liu and X. Chen, "Continuous gesture recognition with hand-oriented spatiotemporal feature", in "Proceedings of the IEEE International Conference on Computer Vision Workshops", pp. 3056–3064, 2017.

[15] C. Nguyen, G. S. Tran, T. Nghiem, N. Doan, D. Gratadour, J.-C. Burie and C. Luong, "Towards Real-Time Smile Detection Based on Faster Region Convolutional Neural Network", in "1st International Conference on Multimedia Analysis and Pattern Recognition (MAPR)", pp. 1–6, 04 2018, doi:10.1109/MAPR.2018.8337524.

[16] V. Pitsikalis, S. Theodorakis, C. Vogler and P. Maragos, "Advances in phonetics-based sub-unit modeling for transcription alignment and sign language recognition", in "CVPR 2011 WORKSHOPS", pp. 1–6, IEEE, 2011.

[17] S. Ren, K. He, R. B. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 39:pp. 1137–1149, 2015.

[18] Sign Community UK, "Introduction to Sign Language", 2013.

[19] B. Sumer, "Lecture notes in Transcription of Sign Language", Faculty of Linguistics, University of Amsterdam, January 2019, lecture 1, Intro to Komva.

[20] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks", in "Proceedings of the IEEE international conference on computer vision", pp. 4489–4497, 2015.

[21] J. R. Uijlings, K. E. Van De Sande, T. Gevers and A. W. Smeulders, "Selective search for object recognition", *International journal of computer vision*, volume 104(2):pp. 154–171, 2013.

[22] J. Wan, Y. Zhao, S. Zhou, I. Guyon, S. Escalera and S. Z. Li, "Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops", pp. 56–64, 2016.

[23] H. Wang, P. Wang, Z. Song and W. Li, "Large-Scale Multimodal Gesture Recognition Using Heterogeneous Networks", in "2017 IEEE International Conference on Computer Vision Workshops (ICCVW)", pp. 3129–3137, 2017.

[24] H. Wang, P. Wang, Z. Song and W. Li, "Large-Scale Multimodal Gesture Recognition Using Heterogeneous Networks", in "The IEEE International Conference on Computer Vision (ICCV)", Oct 2017.