International journal of basic and applied research

www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

HAND GESTURE USING CONVOLUTIONAL NEURAL NETWORKS

[1]Mr.K. Obulesh,[2]P. Shraddha Sree,[3]U. Anusha,[4]V. Rishitha
[1]Assistant Professor,[234]Students
Department Of CSE
Malla Reddy Engineering College for Women

ABSTRACT

The goal of Sign Language Recognition (SLR) is to enable deaf-mute individuals to communicate with the general public by translating sign language into text or voice. Despite the wide-ranging societal effects, the intricacy and wide-ranging hand motions make this work very difficult. Current state-of-the-art SLR approaches construct classification models using manually-crafted characteristics that characterize motion in sign language. Nevertheless, trustworthy features that can adjust to the wide variety of hand movements are challenging to build. In order to tackle this issue, we present a new convolutional neural network (CNN) that can automatically, and without human intervention, extract discriminative spatial-temporal characteristics from unprocessed video streams. Convolutional neural networks (CNNs) are trained to improve performance by feeding them multi-channel video feeds that include color information, depth clues, and the locations of the body's joints. By comparing it to more conventional methods that rely on manually created features, we show that the suggested model outperforms the former on a real-world dataset acquired using Microsoft Kinect.

1. INTRODUCTION:

Variations in hand-shapes, body language, and facial expressions constitute sign language, one of the most popular forms of communication for the deaf and hard of hearing. Sign language recognition remains a formidable challenge due to the difficulty of collectively using data derived from hand-shapes and body movement trajectories. In order to facilitate communication between the hearing impaired and non-disabled individuals using sign language, this research presents a recognition model that is successful in translating sign language into text or voice.

Creating descriptors that convey hand-shapes and motion trajectory is, from a technical standpoint, the primary obstacle to sign language recognition. To be more specific, hand-shape description needs solving challenges with gesture detection, monitoring hand areas in video streams, and segmenting hand-shape pictures from complicated backgrounds in each frame. Key point tracking and curve matching are also connected to motion trajectory. There has been a lot of effort on these two fronts, but getting good results with SLR is still challenging because of the variability and occlusion of hands and body joints. Furthermore, combining the hand-shape and trajectory features is not a simple task. We build convolutional neural networks (CNNs) that can organically include hand-shapes, action trajectory, and facial expression to overcome these challenges. We employ color, depth, and body skeleton pictures all at once as input from Microsoft Kinect, rather than the more conventional color images used by networks [1, 2].

Streams in both color and depth may be provided by the Kinect motion sensor. You may get the joint positions of the body in real-time using the public Windows SDK, as seen in Figure 1. Kinect was the obvious choice for recording the sign words dataset. To differentiate between the various sign actions, it is helpful to include information about the color change and the depth of the pixels. The trajectory of sign actions may be depicted by the variation of bodily joints in the time dimension. When fed data from a variety of visual sources, CNNs start to pay attention to changes in depth, direction, and hue. We may sidestep the challenges of hand tracking, background hand segmentation, and hand descriptor construction by using CNNs' inherent capacity to autonomously learn features from raw data in the absence of background information [3].

In recent years, CNNs have found use in video stream categorization. A possible issue with CNNs is the amount of time they take. Training a convolutional neural network (CNN) on a million-scale using a million videos takes weeks or months. As luck would have it, real-time efficiency is still within reach, thanks to CUDA and other parallel processing tools. Using convolutional neural networks (CNNs), we suggest enhancing sign language recognition (SLR) by extracting spatio-temporal information from live video streams. The current state of the art in SLR relies on manually created features to characterize the movements of sign language and construct a classification model from there. Contrarily, convolutional neural networks (CNNs) don't need features to automatically extract motion information from streaming video. We create CNNs that can process various kinds of data. By applying convolution and subsampling to neighboring video frames, this architecture incorporates depth, color, and trajectory data. Our experimental findings show that 3D CNNs perform far better than GMM-HMM baselines on a set of sign phrases that we recorded ourselves.

2. SYSTEM ANALYSIS

Existing System

Making a desktop tool that can record an individual's American Sign Language (ASL) motions using a computer's camera and instantly convert them into text and audio. We will obtain the translated sign language gesture in writing form, and then we will convert it to audio.

Disadvantage of existing system

1. Less efficiency.

Proposed system

A sign language translator that uses finger spelling is being implemented in this way. We are using a Convolutional neural network (CNN) to facilitate gesture detection. After enough training, a convolutional neural network (CNN) can efficiently tackle computer vision issues and accurately recognize the necessary characteristics.

Advantage of Proposed system

1. More efficiency.

3. MODULES DESCRIPTION:

User:

Launching the project is as simple as double-clicking the run.py file. We can train CNN using gesture images once the user uploads a dataset of hand gestures. Open the CV class, user. In this context, VideoCapture(0) refers to the main camera of the system, whereas VideoCapture(1) denotes the secondary camera. We may import the video file from disk even without the camera by using VideoCapture(Videfile path). Vgg16 and Vgg19 have been configured programmatically. The code allows the user to run the program in several ways and alter the selected model.

HSR System:

utilizing video Using one's Visual cues are the basis for sign recognition. The RGB or depth picture is used by the vision-based approach. The user is not obligated to wear any sensors or carry any equipment. So, this approach is becoming more popular currently, which makes the HSR framework easier to use and deploy in various contexts. The frames for each activity were first retrieved from the movies. In particular, we acquire trained machine learning classifiers and deep picture features using transfer learning.

Using one's Hand Motion Detection While computer processing power has quadrupled in the last decade, HCIs have remained mostly unchanged. The use of intermediate devices, such as keyboards and mouse, limits our freedom of movement while working with computers. Unfortunately, they have become a stumbling block in the field of human-computer interaction due to how annoying they are. Every day, we communicate verbally and using body language to indicate, highlight, and guide ourselves. For humans, these methods of interacting with computers are much more intuitive and pleasant. But getting computers to grasp this is no picnic. The field of computer science and language technology known as "gesture recognition" aims to understand human gestures by use of mathematical algorithms. Although they often begin with the hands or face, gestures may come from any part of the body. One interpretation of gesture recognition is that it paves the door for computers to decipher human body language, allowing for a more robust interaction between the two species. With the use of gesture recognition, people may have natural conversations with machines without the need for any kind of mechanical equipment. Combining methods from computer vision and image processing allows for gesture recognition.

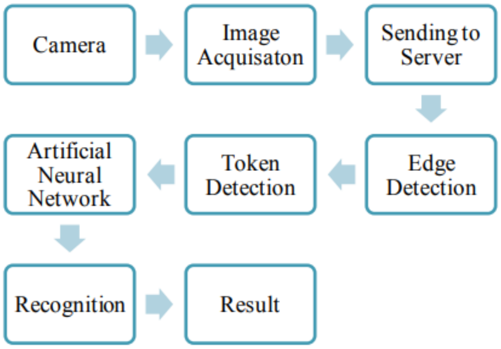


Figure 1: Block diagram of System

Hand and Fingers and Palm Segmentation

There is a demonstration of the original photographs used in the work for hand gesture recognition. A standard camera was used to take these pictures. The identical conditions were used to get these photos of hands. Each of these pictures has the same backdrop. Therefore, using the background subtraction approach, it is simple and effective to recognize the hand area from the original picture. But sometimes, the outcome of background reduction includes additional moving objects. The hand area may be distinguished from other moving objects using its skin tone. The HSV model is used to quantify skin color. The skin tone has the following HSV values: 315, 94, and 37 for hue, saturation, and value, respectively. To make the gesture identification invariant to picture scale, the identified hand's image is scaled.

The binary picture produced by the hand detection process has white pixels representing the hand area and black pixels representing the backdrop. Next, the fingers and palm of the binary hand picture are segmented using the following process.

Point Palm. The central point of the palm is known as the palm point. The distance transform technique is used to find it. An picture is represented by a distance transform, often known as a distance map. Every pixel in the distance transform picture keeps track of both its own distance and the distance to the closest border pixel.

Inside the Maximal Radius's Circle. Once the palm point has been located, it may create a circle with the palm point serving as the innermost point. Because the circle is a part of the palm, it is referred to as the inner circle. The circle's radius steadily grows until it approaches the palm's edge.

The palm mask and wrist points. A wider circle with a radius 1.2 times that of the maximum inner circle is created upon acquiring the radius of the maximal inner circle.

Rotation of the Hand. An arrow going from the palm point to the middle point of the wrist line at the bottom of the hand may be produced after the wrist and palm points have been determined. Subsequently, the arrow is reoriented towards the north. To make the hand motion invariant to the rotation, the hand picture spins simultaneously. In the meanwhile, the portions of the rotational picture below the wrist line are cropped to provide an accurate hand image that does not include the arm portion.

Convolutional Neural Network (CNN)

Sign language is a commonly used communication method for those with hearing impairments. It is conveyed using a variety of hand gestures, body postures, and facial expressions. Sign language recognition remains a very tough problem since it is hard to jointly leverage the information from hand forms and body movement trajectory. In order to assist the hearing challenged in communicating with the general public using sign language, this study suggests an efficient recognition model for translating sign language into text or voice.

Technically, creating descriptors that represent hand forms and motion trajectory is the primary issue in sign language recognition. Specifically, hand-shape description includes gesture detection issues, segmenting hand-shape pictures from complicated backgrounds in each frame, and monitoring hand areas in video streams. Curve matching and critical point tracking are also connected to motion trajectory. In order to tackle these challenges, we create CNNs that inherently include hand forms, motion trajectories, and face expressions. Rather of feeding standard color pictures into networks such as those seen in [1, 2], we employ

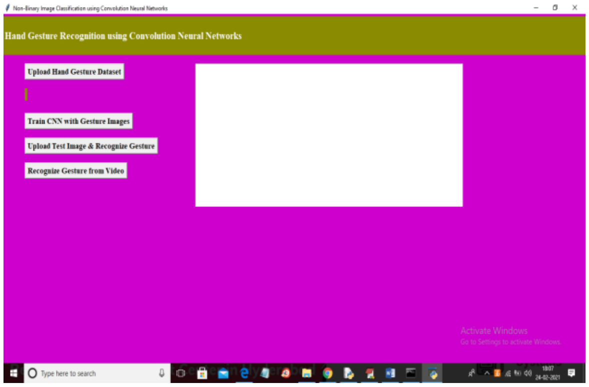
Microsoft Kinect to concurrently give color, depth, and body skeleton images as input.

Kinect is a motion sensor that offers depth and color streams. As shown in Fig.1, the body joint positions may be acquired in real-time using the public Windows SDK. As a result, we decided to record sign words dataset using Kinect as our capture equipment. Utilizing color shift and pixel level depth may help distinguish between various sign motions. Additionally, the trajectory of sign activities may be represented by the fluctuation of body joints in time dimension. When many kinds of visual inputs are used as input, CNNs learn to notice changes in depth, trajectory, and color as well as color changes. It is important to note that since CNNs can automatically learn features from raw data without any previous information, we may bypass the challenges of monitoring hands, segmenting hands from background, and generating hand descriptors [3].

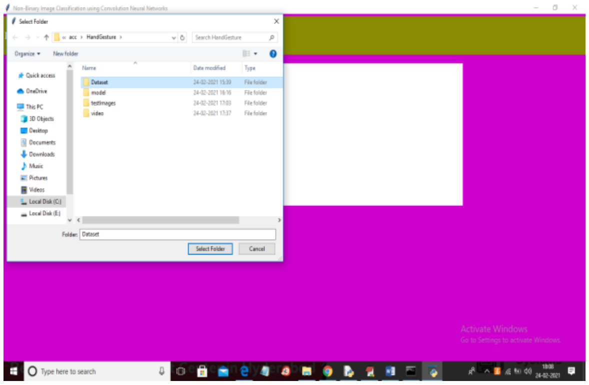
In recent years, CNNs have been used to classify video streams. CNNs may have issues with time consumption. The process of training a CNN with millions of scale in millions of videos takes weeks or months. Thankfully, real-time efficiency may still be attained with the use of CUDA for parallel processing. In order to extract spatial and temporal information from video streams for Sign Language Recognition (SLR), we suggest using CNNs. Current SLR techniques describe motion in sign language using hand-crafted characteristics, then utilize these features to create a classification model. Conversely, CNNs do not need to develop features since they can automatically extract motion information from unprocessed video data. We create CNNs by feeding them various kinds of data. This design uses convolution and subsampling on neighboring video frames to combine color, depth, and trajectory information. Based on our own recordings of certain sign phrases, experimental findings show that 3D CNNs can perform much better than Gaussian mixture models with Hidden Markov model (GMM-HMM) baselines.

4. SCREEN SHOTS

Click "run" twice to launch the project.bat file to get the screen below


Non-Binary Image Classification using Convolution Neural Networks
Hand Gesture Recognition using Convolution Neural Networks
Upload Hand Gesture Dataset
Train CNN with Gesture Images
Upload Test Image & Recognize Gesture
Recognize Gesture from Video
Type here to search

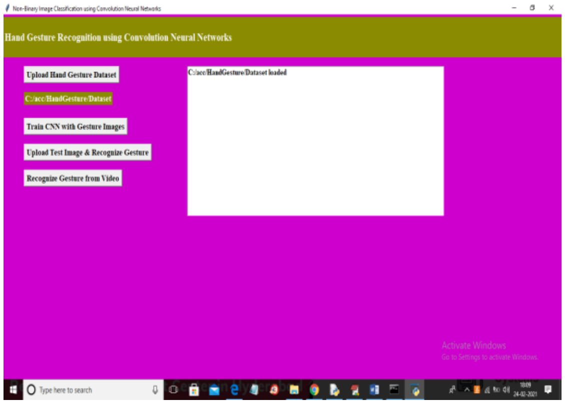
Click the "Upload Hand Gesture Dataset" button on the top screen to upload the dataset and see the screen below.



Name	Date modified	Type
Dataset	24-02-2021 15:39	File folder
model	24-02-2021 16:16	File folder
testImages	24-02-2021 17:03	File folder
video	24-02-2021 17:37	File folder

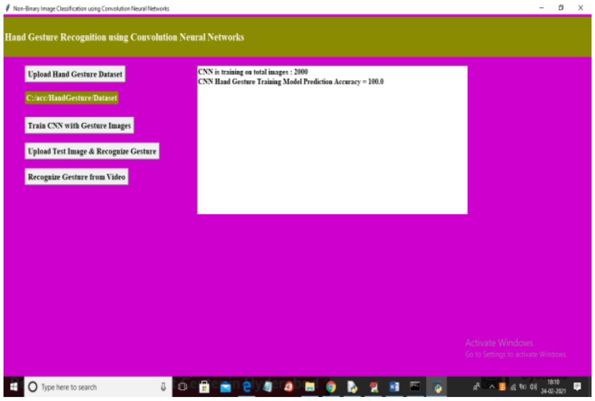
Folder: Dataset

To load the dataset and see the screen below, pick and upload the "Dataset" folder in the above screen. Then, click the "Select Folder" button.


Non-Binary Image Classification using Convolution Neural Networks
Hand Gesture Recognition using Convolution Neural Networks
Upload Hand Gesture Dataset
C:/acc/HandGesture/Dataset
Train CNN with Gesture Images
Upload Test Image & Recognize Gesture
Recognize Gesture from Video
Activate Windows
Go to Settings to activate Windows.
Type here to search

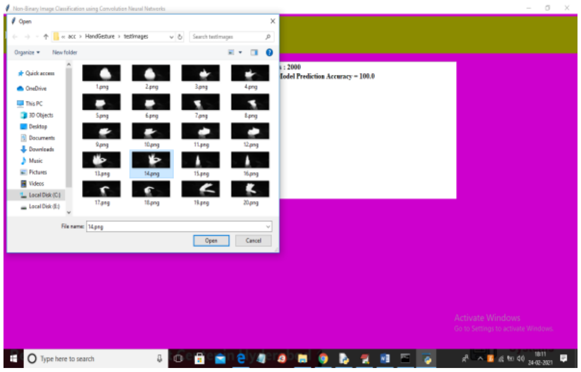
C:/acc/HandGesture/Dataset loaded

The dataset is loaded on the above screen; click the "Train CNN with Gesture Images" button to train the CNN model and see the screen below.


Non-Binary Image Classification using Convolution Neural Networks
Hand Gesture Recognition using Convolution Neural Networks
Upload Hand Gesture Dataset
C:/acc/HandGesture/Dataset
Train CNN with Gesture Images
Upload Test Image & Recognize Gesture
Recognize Gesture from Video
Activate Windows
Go to Settings to activate Windows.
Type here to search

CNN is training on total images : 2000
CNN Hand Gesture Training Model Prediction Accuracy = 100.0

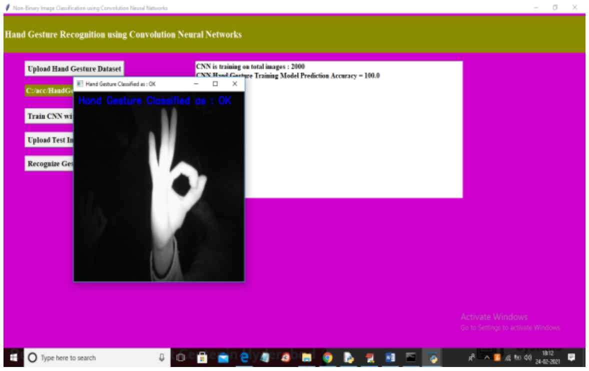
In the screen above, the CNN model was trained on 2000 photos, and its prediction accuracy was 100%. The model is now ready; to submit an image and enable gesture recognition, click the "Upload Test Image & Recognize Gesture" button.



File name: 14.png

: 2000
Model Prediction Accuracy = 100.0

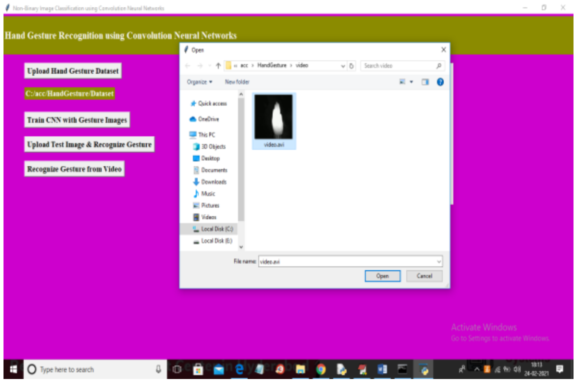
Selecting and uploading the "14.png" file in the above screen, then clicking the Open button to see the outcome below


Non-Binary Image Classification using Convolution Neural Networks
Hand Gesture Recognition using Convolution Neural Networks
Upload Hand Gesture Dataset
C:/acc/HandGe
Train CNN wi
Upload Test In
Recognize Ges
Hand Gesture Classified as : OK
Activate Windows
Go to Settings to activate Windows.
Type here to search

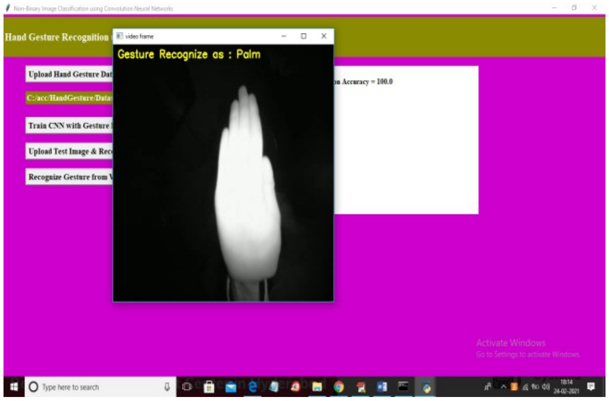
CNN is training on total images : 2000
CNN Hand Gesture Training Model Prediction Accuracy = 100.0

Hand Gesture Classified as : OK

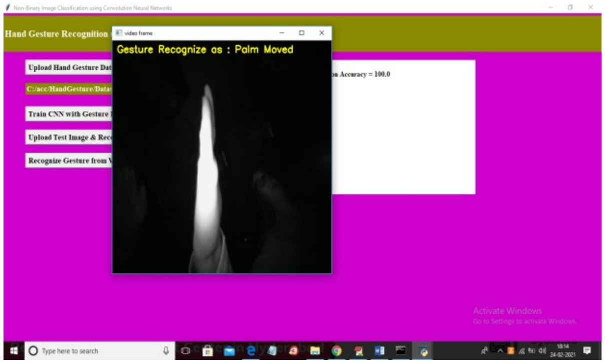
The gesture on the following screen is recognized as OK, and you may submit any picture to achieve the same result. Alternatively, you can upload a video by clicking the "Recognize Gesture from Video" button.


Hand Gesture Recognition using Convolution Neural Networks
Upload Hand Gesture Dataset
C:/acc/HandGesture/Dataset
Train CNN with Gesture Images
Upload Test Image & Recognize Gesture
Recognize Gesture from Video
Open
video.avi
File name: video.avi
Open
Cancel

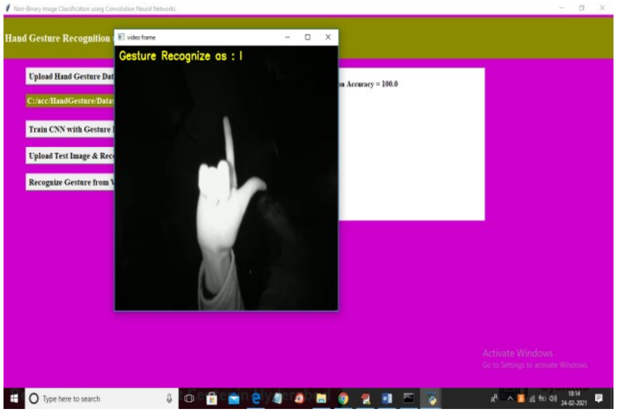
To get the following result, pick and upload the "video.avi" file on the above screen, then click "Open."



Gesture Recognize as : Palm



Gesture Recognize as : Palm Moved



Gesture Recognize as : I

When the video on the top screen plays, a recognition result will appear.

5. CONCLUSION

For the purpose of sign language recognition, we used a convolutional neural network (CNN) model. Through the use of 3D convolutions, our model is able to learn and extract information related to both space and time. In order to do convolution and subsampling independently, the created deep architecture first pulls various kinds of information from neighboring input frames. Every channel's data is included into the final feature representation. In order to categorize these feature representations, we use a multilayer perceptron classifier. We test CNN and GMM-HMM on the same dataset so you can see how they compare. The experimental findings prove that the suggested approach works.

REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in

Advances in neural information processing systems, 2012, pp. 1097–1105.

[2] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei, "Large-scale video classification with convolutional neural networks," in CVPR, 2014. [3] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick ´ Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] Hueihan Jhuang, Thomas Serre, Lior Wolf, and Tomaso Poggio, "A biologically inspired system for action recognition," in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. Ieee, 2007, pp. 1–8. [5] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu, "3D convolutional neural networks for human action recognition," IEEE TPAMI, vol. 35, no. 1, pp. 221–231, 2013.

[6] Kirsti Grobel and Marcell Assan, "Isolated sign language recognition using hidden markov models," in Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on. IEEE, 1997, vol. 1, pp. 162–167.

[7] Thad Starner, Joshua Weaver, and Alex Pentland, "Realtime american sign language recognition using desk and wearable computer based video," IEEE TPAMI, vol. 20, no. 12, pp. 1371–1375, 1998.

[8] Christian Vogler and Dimitris Metaxas, "Parallel hidden markov models for american sign language recognition," in Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on. IEEE, 1999, vol. 1, pp. 116–122.

[9] Kouichi Murakami and Hitomi Taguchi, "Gesture recognition using recurrent neural networks," in Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 1991, pp. 237–242.

[10] Chung-Lin Huang and Wen-Yi Huang, "Sign language recognition using model-based tracking and a 3D hopfield neural network," Machine vision and applications, vol. 10, no. 5-6, pp. 292–307, 1998.

[11] Jong-Sung Kim, Won Jang, and Zeungnam Bien, "A dynamic gesture recognition system for the korean sign language (ksl)," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 26, no. 2, pp. 354–359, 1996.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," arXiv preprint arXiv:1311.2524, 2013.

[13] Ronan Collobert and Jason Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in ICML. ACM, 2008, pp. 160–167.

[14] Clement Farabet, Camille Couprie, Laurent Najman, ´ and Yann LeCun, "Learning hierarchical features for scene labeling," IEEE TPAMI, vol. 35, no. 8, pp. 1915– 1929, 2013.

[15] Srinivas C Turaga, Joseph F Murray, Viren Jain, Fabian Roth, Moritz Helmstaedter, Kevin Briggman, Winfried Denk, and H Sebastian

Seung, “Convolutional networks can learn to generate affinity graphs for image segmentation,” Neural Computation, vol. 22, no. 2, pp. 511– 538, 2010.