Gesture Similarity Learning and Retrieval in Large Scale Real-World Video Collections

Inauguraldissertation

zur
Erlangung der Würde eines Doktors der Philosophie
vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel
und der
Faculté Polytechnique
der Université de Mons

von

Mahnaz Parian-Scherb

Basel, 2021

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät
auf Antrag von

Prof. Dr. Heiko Schuldt, Erstbetreuer
Prof. Dr. Stéphane Dupont, zusätzlicher Erstbetreuer
Prof. Dr. Volker Roth, Zweitbetreuer
Prof. Dr. Mark Turner, Externer Experte

Basel, den 25.05.2021

Prof. Dr. Marcel Mayor, Dekan

*"To my angel on earth, Mom
and Dad, my guardian in heaven."*

Abstract

Analyzing and understanding gestures plays a key role in our comprehension of communication. Investigating the co-occurrence of gestures and speech is currently a labor-intensive task in linguistics. Although, with advances in natural language processing methods, there have been various contributions in this field, computer vision tools and methods are not prominently used to aid the researchers in analyzing hand and body gestures.

In this thesis, we present different contributions tailored to tackle the challenges in real-world gesture retrieval which is an under-explored field in computer vision. The methods aim to systematically answer the questions of 'when' a gesture was performed and 'who' performed it in a video. Along the way, we develop different components to address various challenges in these videos, such as the presence of multiple persons in the scene, heavily occluded hand gestures and abrupt gesture cuts due to the change of camera angle.

In contrast to the majority of the existing methods developed for gesture recognition, our proposed methods do not rely on the depth modality or sensor signals, which is available in some datasets to aid the identification of gestures. Our vision-based methods are built upon the best practices in learning the representations of complicated actions using Deep Neural Networks. We have conducted a comprehensive analysis to choose the architectures and configurations to extract discriminative spatio-temporal features. These features enable the retrieval pipeline to find the 'similar' hand gestures. We have additionally explored the notion of similarity in the context of hand gestures through field studies and experiments.

Finally, we conduct exhaustive experiments on different benchmarks and to the best of the author's knowledge, run the largest gesture retrieval evaluations using the real-world news footage, the Newscape dataset, which is a collection of more than 400 000 videos with numerous challenging scenes for a retrieval method. The assessed results by experts from the linguistics domain suggest high potential of our proposed method in inter-disciplinary research and studies.

Jury members

Prof. Dr. Saïd Mahmoudi (UMONS)	President
Prof. Dr. Stéphane Dupont (UMONS)	Supervisor
Prof. Dr. Heiko Schuldt (University of Basel)	Supervisor
Prof. Dr. Thierry Dutoit (UMONS)	Jury member
Prof. Dr. Volker Roth (University of Basel)	Jury member
Prof. Dr. Mark Turner (Case Western Reserve University)	External Expert

Acknowledgements

First and foremost I am deeply grateful to my supervisors for their invaluable advice, continuous support, and patience during my PhD study. I would like to offer my special thanks to Prof. Dr. Stéphane Dupont, my supervisor from University of Mons, for all the support and valuable feedback on various topics throughout my PhD during both the time I spent in Mons and also the time I was in Basel. Thank you for so many brilliant ideas through our inspiring discussions. I would like to express my sincere gratitude to my supervisor at University of Basel, Prof. Dr. Heiko Schuldt for his encouragement and support during the entire period of my PhD. I am especially thankful for the countless opportunities you introduced me to and for being a personal mentor in my daily life in addition to a *Doktorvater* in my academic research.

I am very thankful to my committee members Prof. Dr. Thierry Dutoit, Prof. Dr. Saïd Mahmoudi and Prof. Dr. Volker Roth for their invaluable feedback during my annual committee meetings. Additionally, I would like to express my gratitude to Prof. Dr. Mark Turner for reviewing my thesis as an external expert.

I would especially like to thank Redhenlab and Prof. Dr. Mark Turner and Prof. Dr. Francis Steen for providing the NewsScape dataset and resources for evaluations of this thesis. I am especially thankful for being a part of this organization in multiple instances of Google Summer of Codes and gaining very unique mentoring experiences through this event. I am especially grateful for our discussions on the linguistic perspectives of gestures that inspired some of the ideas in this thesis. I would like to extend my gratitude to Dr. Peter Uhrig for our fruitful discussions and giving me the opportunity to explore this field further at FAU as a researcher.

A significant part of my experience in multi-media retrieval comes from my collaborations with the vitrivr team. I am especially thankful to my former colleague and my friend, Dr. Luca Rossetto for our fruitful collaborations, thought-provoking discussions, his support during the writing part of this thesis and his patience with my learning curve in Secure Shell. I would like to extend my gratitude to both former and current members of the vitrivr team, specially Dr. Ivan Giangreco, Ralph Gasser, Silvan Heller, Loris Sauter and Florian Spiess, for the pleasant time and inspiring discussions during the hackathons and conferences we participated together.

I would like to extend my sincere thanks to my colleagues from two different universities, specially Kevin El-Haddad for being the starting point of my PhD and introducing me to Prof. Dr. Stéphane Dupont five years ago. I additionally would like to thank Alexander Stiemer, for his genuine personal and technical support from the time I joined DBIS group.

A big thank you to Dr. Nada Ben-Lhachemi for her support during the most difficult time of my life. I would also like to thank DBIS and ISIA (former TCTS) members, specially Dr. Lukas Probst and Marco Vogt for all the engaging discussions, social events and attempts to teach me French and German languages.

I wish to specially thank Yvonne Wegmüller and Nathalie Durieux for their administrative support and magically solving all the difficulties I faced with the two affiliations in three countries throughout my PhD journey.

I would like to say a big thank you to all the women in the Department of Mathematics and Computer Science at University of Basel who inspired me both scientifically and personally and supported the endeavor of the "We are Computer Science" association. A special thanks goes to Claire Walzer who never ceased to surprise me with her wits, commitment and her hard work. I would like to extend my gratitude to Prof. Dr. Florina Ciorba, Dr. Gabriele Röger, Cristina Illi, Esther Mugdan, Jana Völker, for their enthusiasm which was always my driving motor during the last stages of my PhD. I would also like to thank Dr. Sonali Parbhoo and Dana Rahbani, for the sport classes and quality times we spent together.

Last but not the least, I would like to express my deepest gratitude to all my family without whom I would not have been here. I would like to thank my brothers and in-laws for their support, understanding and patience. A big thanks to my lovely nieces, Niku and Diba who were the source of joy in the cloudy days of my life. I cannot express how thankful I am to my mom, Hajar, for her endless love, unconditional support and encouragement throughout my life and giving me wings to fly even though it took me far from her and to my dad, Alireza, who is no longer among us today, but his loving memories always guided me like the North star in the sky. A special thanks goes to my best friend and husband, Dr. Christopher Scherb for believing in me and his support throughout the ups and downs of this journey.

Contents

List of Acronyms

AQD Asymmetric Quantizer Distance.

AVH Attention-based Video Hashing.

BSN Boundary Sensitive Network.

C3D 3D Convolutional Network.

CBVR Content-Based Video Retrieval.

CMP Convolutional Pose Machine.

CNN Convolutional Neural Network.

CSR Corrected Segmentation Rate.

DCG Discounted Cumulative Gain.

DNN Deep Neural Network.

DSTW Dynamic Space-Time Warping.

DTQ Deep Triplet Quantization.

DTW Dynamic Time Warping.

ELAN EUDICO Linguistic Annotator.

FC Fully Connected.

FCN Fully Convolutional Network.

FPN Feature Pyramid Network.

GRU Gated Recurrent Unit.

HMDB Human Motion Database.

HMM Hidden Markov Model.

HOG Histogram of Gradients.

I3D Inflated 3D Convolutional Neural Net.

ICM Iterated Conditional Modes.

IDT Improved Dense Trajectories.

IoU Intersection over Union.

JHMDB Joint Annotated Human Motion Database.

LSH Locality Sensitive Hashing.

LSTM Long Short Term Memory.

mAP mean Average Precision.

MFH Multiple Feature Hashing.

MJI Mean Jaccard Index.

MLP Multi Layer Perceptron.

NER Named entity recognition.

NLP Natural Language Processing.

NMS Non-Max Suppression.

NPH Neighborhood Preserving Hashing.

OIM Online Instance Matching.

PAF Part Affinity Field.

PCA Principal Components Analysis.

QbE Query by Example.

QOM Quantity of Movement.

R-CNN Regions with CNN features.

ReLU Rectified Linear Unit.

RNN Recurrent Neural Network.

RoI Region of Interest.

RPAN Recurrent Pose Attention Network.

RPN Region Proposal Network.

S-CNN Segment Convolutional Neural Network.

SGD Stochastic Gradient Descent.

SGM-Net Skeleton-Guided Multimodal Network.

SIFT Scale-Invariant Feature Transform.

SPDTH Similarity-Preserving Deep Temporal Hashing.

SST Single Stream Temporal Action Proposals.

SSTH Self-Supervised Temporal Hashing.

SVM Support Vector Machine.

TAG Temporal Actionness Grouping.

TSN Temporal Segment Network.

TURN-TAP Temporal Unit Regress Network for Temporal Action Proposals.

UDVH Unsupervised Deep Video Hashing.

VA-File Vector Approximation File.

VBS Video Browser Showdown.

VRS Video Relay Service.

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Introduction

Human communication consists of non-verbal modalities such as gestures along with verbal speech which in many cases complement each other. Nonverbal communication makes up to two-thirds of all communications among humans [15] and is one of the core components of human-machine interactions. However, our understanding of the co-occurrence of these modalities with spoken language is still limited. Gestures can be distinguished from other movements, segmented, and assigned a meaning based on their forms and functions.

Depending on the language, gestures emerge in all levels of linguistic structures. Typically, a gesture has rich cognitive dimensions in addition to its communicative dimensions [16]. Generally, gesture and speech are believed to be elements of the same cognitive process, and investigation of their connection is difficult due to the inexact co-occurrence. The study of this temporal proximity of gesture and speech identifies gestural and verbal patterns which happen systematically [17]. The hand gestures that accompany speech are called *co-speech* gestures and are generally different from cultural-specific gestures. These types of gestures are specifically important in providing semantic information beside the speech to disambiguate the content.

Many different research directions for example in neuroscience [18] and linguistics [19] are focused on analyzing the co-speech gestures. However, there are still limitations in the availability of annotated media contents which is essential for these analyses. In the meantime, projects such as NewsScape [6] have given researchers access to large volumes of news footage and talk shows, with a tremendous amount of hand gesture and pose instances. But so far the amount of annotated videos is limited due to the manual annotations of the documents.

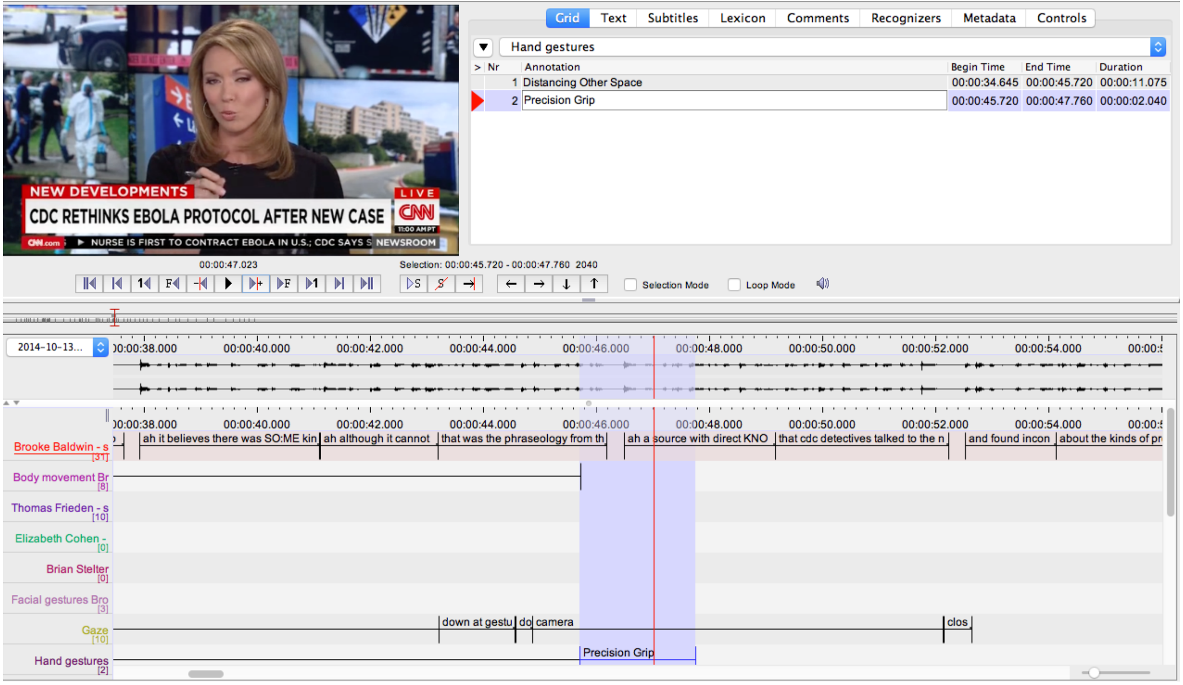
Meanwhile in the computer science domain, with the advent of machine and deep learning techniques, different algorithms and methods are developed which can be used to search in large-scale media collections. Such a search system can be extended to gesture modality and be used to explore the videos with gestural content.

1.1 Motivation

The main motivation of this work lies in the current gesture search methods in linguistics and the capabilities of the retrieval methods in computer science. In the following we explore the two sides of this topic.

1.1.1 ELAN and Gesture Search

A large amount of annotations in linguistics is done manually by an existing tool called ELAN which provides a professional interface to annotate audio and video with text. The manual annotation of videos requires a large amount of time and effort, due to the need of precise and accurate labeling of the videos with the right time stamps where the gesture starts and with the correct gesture annotations. An example of the interface of ELAN can be seen in Figure 1.1.



Hand gestures

>	Nr	Annotation	Begin Time	End Time	Duration
	1	Distancing Other Space	00:00:34.645	00:00:45.720	00:00:11.075
	2	Precision Grip	00:00:45.720	00:00:47.760	00:00:02.040

Figure 1.1. An example of the interface of theELAN tool for gesture annotation[1]. Beside the detailed annotation of timestamps each phrase is uttered, the orientation, motion and configuration of the fingers are specified in this tool.

The annotated gestures are described usually by their form, and sometimes with the function they have in the speech. For example in Figure 1.1 we see that the hand gesture is tagged as *precision grip* and the start/end time of it is specified. Sometimes the gestures do not have a specific name, therefore they are described with their form. For example, together with the speech:
"... SHE'S 8 MONTHS OLD, SHE CAN FEND FOR HERSELF. WHO INVENTED WATER? WHAT WOULD YOU SAY TO THAT? I'M WONDERING WHAT YOUR ANSWER IS. "
The hand gesture corresponding with the *What would you say* (shown in Figure 1.2) is labeled as: Flips hand to palm-up position.

When searching for the gestures, either one needs to look for the spoken phrase and explore the videos to see if the gesture actually happens, or search for the gesture annotation. However,

[1] Image from https://www.redhenlab.org/

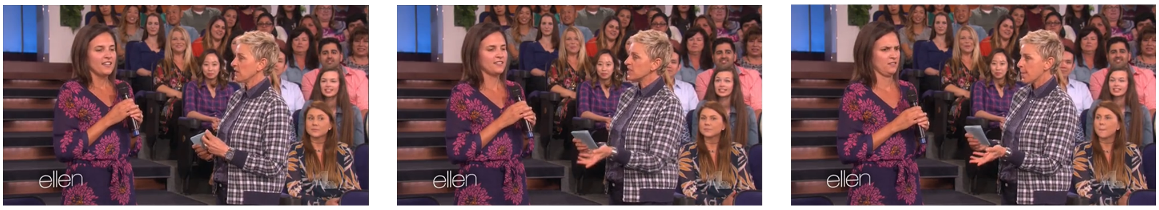


Figure 1.2. Sample frames from the gesture accompanying the phrase *What would you say* which is annotated as ”flips hand to palm-up position” in ELAN.

despite the high level of details in gesture annotations, the slight variations of the form of gestures in different situations would result in missing some of the objects of interest. Additionally, the semantic gap between the translation of a visual event into a text, would result in loss of information. Therefore, using the same visual query to perform the search to find the potential similar instances of the same gesture could possibly improve the returned results and help the linguistic researchers in analyzing the co-speech gestures.

1.1.2 Information Retrieval Systems

Information retrieval refers to the processes of searching for a specific piece of information, locating and retrieving that data from a collection. These processes are generally divided into an *online* and *offline* (Figure 1.3). The *offline* process is referring to extracting features of the collection documents and storing them into a database. The feature extraction is highly dependent on the collection and the need of information. For example in image collections these features could be color, shape, or Deep Neural Network (DNN) features. These features stored in the database will be used in the *online* retrieval process. The *online* part of the

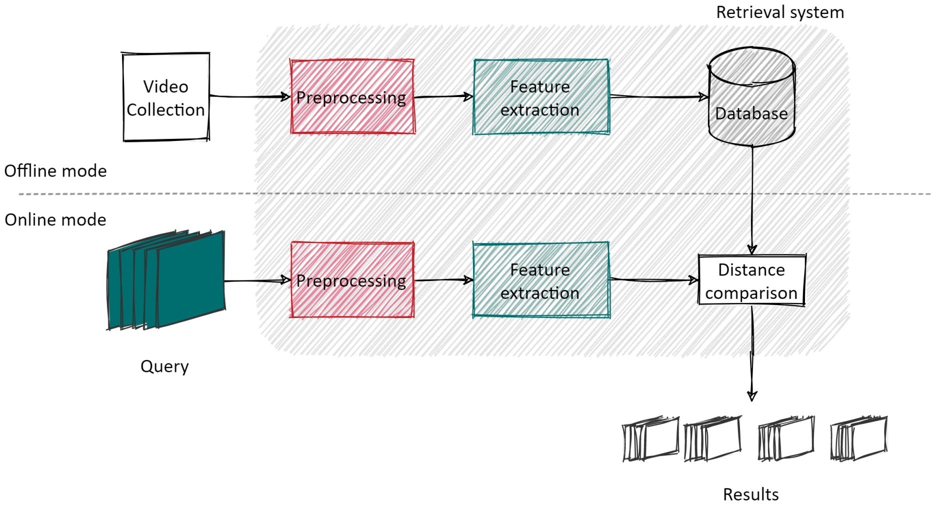


Figure 1.3. A high-level overview of multimedia information system separating the offline and online phase of retrieval.

retrieval includes the query formulation and performing the search in the database. A *query* is the formal statement needed by the information retrieval system to perform the search. In other words, the query is the description of the desired information by the user. This description can be expressed in different modalities such as text, audio, image or sketch or by providing an example to the system (query by example). The search is performed by comparing the feature vectors of the query and the collection (stored in the database) and the objective is to find the collection items with the smallest distance in the vector space with the query.

With the enormous amount of video being produced everyday, the video retrieval processes have gained more and more attention from the information retrieval community. However, despite the great achievements and competitive systems developed for in-the-wild video data, non-verbal communication has not made its way to these systems. Specifically hand gestures, which are very commonly used in human interactions, have no specific role in the video retrieval processes and hand gesture retrieval is still an under explored field in video search systems. Additionally, with the current query formulation techniques, the only mode of the search is query by example, which does not always find the related media.

1.2 Contributions

Motivated by the application in linguistics, this thesis opens an under-explored field in the video retrieval domain. The methods in this thesis are developed to address the challenges in real-world gesture videos, such as background clutter, occlusion, multi-person, cross-angle and untrimmed gesture scenarios. Additionally, these videos are only available in RGB, which limits the functionality of methods using auxiliary modalities, such as depth. The size and diversity of the real-world video collections is yet another challenge for gesture retrieval methods.

In the following we briefly outline the contributions of this thesis and how they address these challenges:

- A domain specific preprocessing method to overcome the challenges existing in real-world data. Our proposed preprocessing component will detect each individual hand gesture in a scene in multi person scenarios by creating clips of their presence in consecutive frames. This component additionally alleviates the susceptibility of the feature extraction process to background noise and occlusion by spatial segmentation throughout the video,

- Two gesture specific vision-based similarity learning and retrieval methods are proposed. The independence from depth modality or other sensor's signals is specifically important due to lack of availability of such data in large-scale real-world video collections. The first feature extraction and metric learning method, ROFI3D is a deep learning based method which projects the RGB and optical flow streams of the short video clips into feature embedding. The discriminative feature vectors are used not only for gesture recognition, but are also used for similarity learning between the gestures. The second proposed method, RKLSTM, benefits from the spatial attention mechanism in modeling the joint movements and embeds the body joints information and RGB data into a feature space,

- A step was taken toward the binary representation learning and exploring its potential for hand gesture video retrieval. We evaluated the approach and investigated the limitations and challenges of quantization in this domain,

- Proposed a new query type for hand gesture video retrieval, *query by gesture* to further bridge the semantic gap in the human interaction retrieval domain. We have performed evaluations with both proposed methods with this type of query and analyzed the results with quantitative and qualitative measures,

- Exhaustive evaluations and analysis is performed on available labeled dataset, as well as user studies with assessors in different fields to evaluate the quality of the results on large-scale real-world dataset. Additionally, an in-depth discussion on quantitative and qualitative results was presented. We further explored the limitations of our proposed methods, potential improvements and future work.

1.3 List of Publications

The following papers have resulted from some of the work presented in this thesis:

- *Gesture of Interest: Gesture Search for Multi-Person, Multi-Perspective TV Footage*
  Mahnaz Parian-Scherb, Claire Walzer, Luca Rossetto, Silvan Heller, Stéphane Dupont and Heiko Schuldt
  in Proceedings of Content-Based Multimedia Indexing, 2021

- *On the User-centric Comparative Remote Evaluation of Interactive Video Search Systems*
  Luca Rossetto, Ralph Gasser, Silvan Heller, Mahnaz Parian-Scherb, Loris Sauter, Florian Spiess, Heiko Schuldt, Ladislav Peska, Tomas Soucek, Miroslav Kratochvil, Frantisek Mejzlik, Patrik Vesely, Jakub Lokoc
  in IEEE MultiMedia, 2021

- *Are You Watching Closely? Content-based Retrieval of Hand Gestures*
  Mahnaz Parian-Scherb, Luca Rossetto, Heiko Schuldt, Stéphane Dupont
  in Proceedings of the International Conference on Multimedia Retrieval, 2020

- *Interactive lifelog retrieval with vitrivr*
  Silvan Heller, Mahnaz Parian-Scherb, Ralph Gasser, Loris Sauter, Heiko Schuldt
  in Proceedings of the Third Annual Workshop on Lifelog Search Challenge, 2020

- *Vitrivr-Explore: Guided Multimedia Collection Exploration for Ad-hoc Video Search*
  Silvan Heller, Mahnaz Parian-Scherb, Maurizio Pasquinelli, Heiko Schuldt
  in Proceedings of International Conference on Similarity Search and Applications, 2020

- *Combining Boolean and Multimedia Retrieval in vitrivr for Large-Scale Video Search*
  Loris Sauter, Mahnaz Parian-Scherb, Ralph Gasser, Silvan Heller, Luca Rossetto, Heiko Schuldt
  in Proceedings of International Conference on Multimedia Modeling, 2020

- *Retrieval of structured and unstructured data with vitrivr*
  Luca Rossetto, Ralph Gasser, Silvan Heller, Mahnaz Parian-Scherb, Heiko Schuldt
  in Proceedings of the Second Annual Workshop on Lifelog Search Challenge, 2019

- *Deep Learning-based Concept Detection in vitrivr*
  Luca Rossetto, Mahnaz Parian-Scherb, Ralph Gasser, Ivan Giangreco, Silvan Heller, Heiko Schuldt.
  in Proceedings of International Conference on MultiMedia Modeling, 2019

- *Towards good practices for image retrieval based on CNN features*
  Omar Seddati, Stéphane Dupont, Saïd Mahmoudi, Mahnaz Parian-Scherb.
  in Proceedings of International conference on computer vision workshops, 2017

1.4 Thesis Structure

This thesis is organized in 5 parts. After the current part which is aimed to motivate and introduce the problems, the structure of thesis is as follows:

- **Part II** introduces the fundamental concepts used in this thesis. The gesture taxonomy, background on Deep Learning methods and information retrieval are introduced in Chapter 2. In Chapter 3, we introduce the state-of-the-art in vision-based gesture recognition based on different preprocessing and feature extraction techniques. We further introduce the sparse literature in the gesture retrieval task and present a detailed description on different methods addressing the challenges in the gesture recognition and retrieval field.
- **Part III** Presents the methodology of solving the problems stated in the introduction. Chapter 4 introduces the dataset needed for training and later for evaluations. Chapter 5 introduces our proposed preprocessing techniques for overcoming the main challenges in human interaction settings, namely multi person, occlusion and cluttered background. Additionally, two methods to address the temporal localization of gestures are proposed. In Chapter 6 we present our feature extraction methods from the theoretical perspective for gesture recognition and retrieval and all the practical and implementation information reproduce the methods are explained in Chapter 7.
- **Part IV** provides a comprehensive evaluation to compare the usability of the proposed processes with the state-of-the-art when possible. Chapter 8 presents the recognition results of our proposed methods on relevant datasets. Due to the absence of methods specifically addressing the task of gesture retrieval, we performed qualitative and quantitative evaluation studies to view the problem of gesture similarity retrieval from different point of views in Chapter 9. We further discuss the results from both tasks and explore the limitations and existing challenges in the domain in Chapter 10.
- **Part V** concludes this thesis with a summary in Chapter 11 and outlines the applications and future directions in Chapter 12 to address the existing challenges in this domain.

Part II

Foundations and Related Work

Chapter 2

Foundations

Understanding the meaning of gestures, in addition to their relevance to the spoken language, sheds light on the challenges an automatic system can face while recognizing and identifying them. In Section 2.1, we give a brief overview over the gesture definition from the linguistic point of view which is the aspect we focus on and the different types of the gestures which exist, both from the visual and semantic perspectives. Since most of this work is based on *deep learning* methods, in Section 2.2 we review the fundamentals of this field in computer vision, which are being prominently used in our methods. Additionally we introduce the concept of information retrieval and the similarity learning and the modern methods in this area in Section 2.3.

2.1 Gesture Definition

Human communication is considered to be divided into two major components: content-filled verbal and affect-filled nonverbal [20]. Based on this division, expression of emotions, the presentation of one's personality and managing the turn taking among others belong to the nonverbal behavior which is explicitly separated from the verbal form of communication. However, in contrast to the traditional view of communication, researchers such as Kendon [21] suggest that at least one form of nonverbal behavior cannot be independent of verbal communication: *Gestures*.

By definition, gesture refers to the meaningful and spontaneous hand movements which have semantic co-occurrence with spoken utterances [22, 23]. Spoken language and gestures can be considered as a unified system, both originating from a single semantic and mental representation to communicate the speaker's idea and intentions [23]. This also explains why people gesticulate even when the listener cannot see them for example in telephone conversation. The *co-speech* gestures (gestures made while speaking) aid communicating the intention and transferring supplementary information in addition to that presented in speech [24]. For example, when describing a building, one may use hand gestures to explain the shape of the structure. In addition to conveying supplementary information, co-speech gestures are believed to happen as a result of cognitive processes to assist lexical access. According to Krauss [25], knowledge representation in human memory has multiple dimensions such as visual, spatial and motoric, and one memory can be stored via different representations. In other words, gestures which are considered to reflect spatio-dynamic representations, can help the word retrieval process from the memory [25].

Interpersonal non-verbal communication consists of different components, such as eye contact, body posture and hand movement, of which gestures are one of the key elements. There are numerous ways of classifying gestures, linguistically, by the form of the hands, by the interpretation of the meaning, etc. In the following we introduce the different classes of gestures which are being used in this thesis.

2.1.1 Gesture Taxonomy

There are different types of classifications of gestures available depending on the field they are being studied in. These categories vary in considering the interpretations of the gestures or the motion involved in the gesticulation. We describe two main ways of categorizing gestures, one from the communication perspective, the other from the state of the motion point of view.

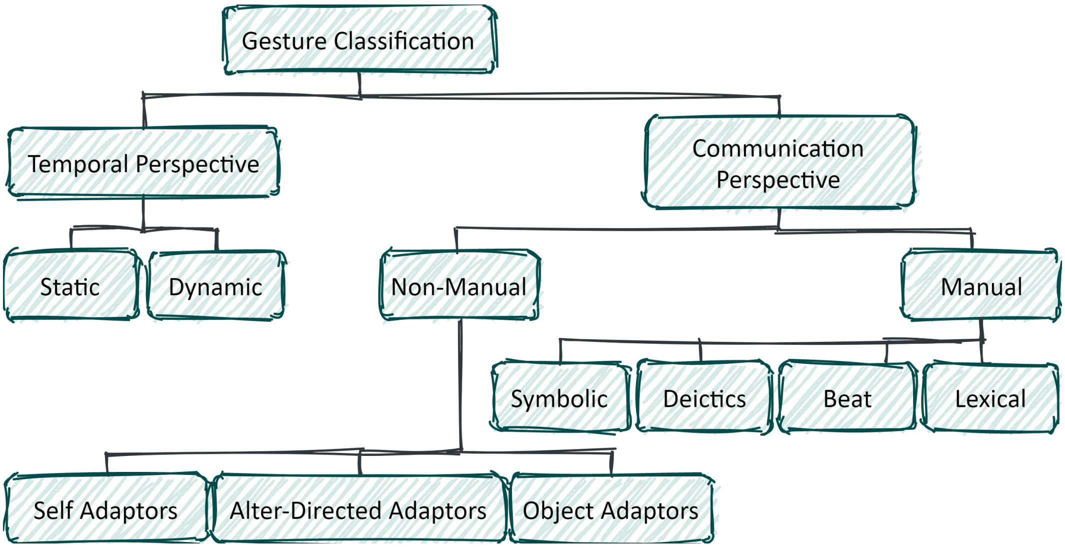


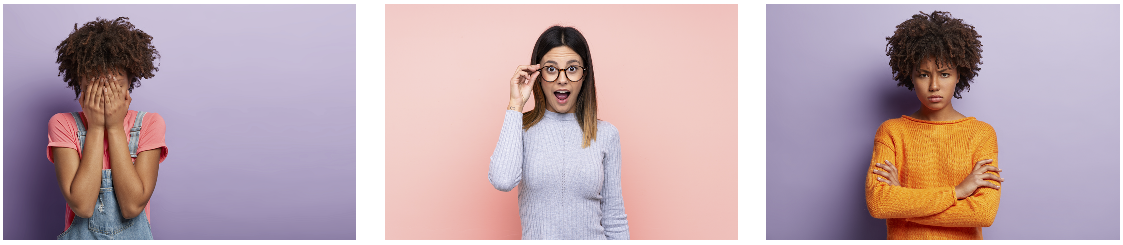
Figure 2.1. Diagram of different categories of gestures from communication and temporal point of view.

2.1.1.1 From a Communication Perspective

From the interpretation of the gestures and the semantic meaning, gestures are being classified to two main categories:

Non-manual Hand Gestures or Adaptors are referred to those gestures done when the person is in discomfort for example in stressful situation or experiencing anxiety. These gestures psychologically help the speaker to adapt to the situation and are unconsciously articulated and are less related to the spoken words. Adaptors are divided in three groups [26, 27]:

- *Self Adaptors* are gestures made for self-comfort, such as stroking the back of the head or touching the face (Figure 2.2a).
- *Alter-Directed Adaptors* are the unconscious gestures in reaction to another person. They are very similar to the self adaptors, with the main difference that the need of the articulation is triggered by another individual. An example of alter adaptors would be crossing the arms when someone enters one's private space as a defensive reaction (Figure 2.2c).
- *Object Adaptors* are the comforting hand movements involving an object such as glasses, paper, etc. Adjusting glasses or a hat are examples of object adaptors (Figure 2.2b).



(a) Self adaptor **(b)** Object adaptor **(c)** Alter-directed adaptor

Figure 2.2. Three examples of adaptor gestures[2]as unconscious reactions to situations: Self adaptor, Object adaptor and Alter-directed adaptor.

Manual Hand Gestures are the class of hand gestures which are communicating special messages and although sometimes articulated unconsciously. They support the conversation and speech. Illustrators are the class of unconscious gestures produced along with speech to create a visual image to support the words and an example of manual gestures. For example when giving directions, one also uses their hands to visualize the path. Manual gestures are divided into four main categories [28, 29]:

- *Symbolic* or emblematic gestures are hand movements that have conventional forms and direct verbal translation. These gestures occur both independent or concurrent with the speech. Emblems are culture dependent and have specific meaning among a class or a group of people. Examples of symbolic gestures are a wave of hand meaning "hello" or "goodbye".
- *Deictics* or indexical gestures are pointing gestures to an object, place or time and work the same way as "this" or "that". These gestures occur simultaneously with speech.
- *Beat* gestures are hand movements with the rhythm of the speech and do not convey any supportive content. They are usually used to emphasize a part of speech.
- *Lexical* or iconic gestures are spontaneous hand movements with more content and are used to elaborate and support the spoken language and are more universal in their usage than emblematics. They could visualize a mental representation of an image or air sketch

[2] Image credits: Figure 2.2a and 2.2c from wayhome-studio/stock.adobe.com and Figure 2.2b from luismolinero/stock.adobe.com

of path or thought. Iconic gestures are co-expressive with speech but not redundant and specially they are used in expressing motion.

However, these categories cannot be used globally since occasionally a gesture from a category is mixed with another, for example beat gestures are often combined with lexical and deictic gestures . In this case they usually are not put into separate categories, when not knowing which category is dominant [30].

2.1.1.2 From a Temporal Perspective

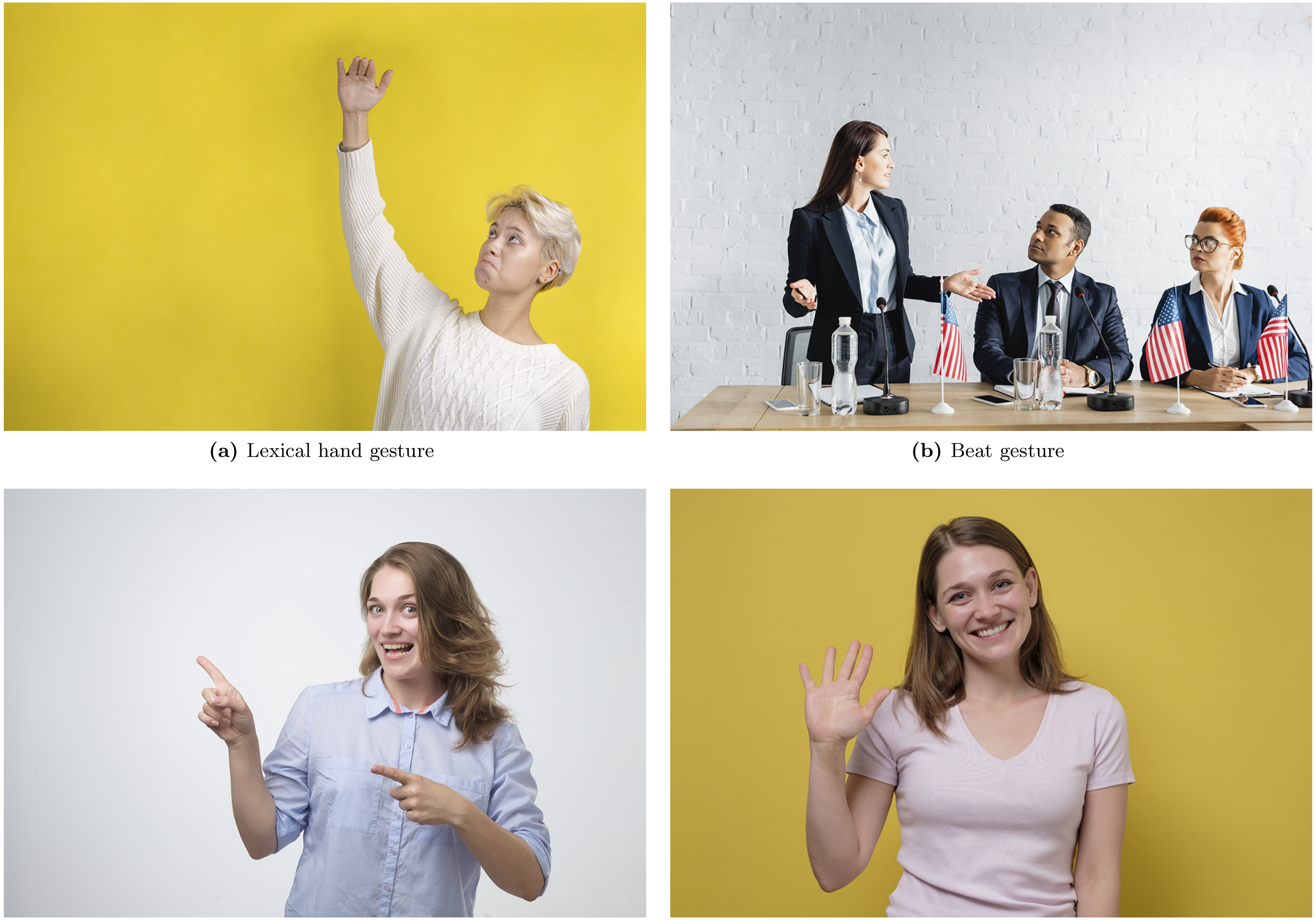
In addition to the communicative aspect of gestures, hand movements can be categorized based on their temporal relationships. This classification is specifically important from computer vision perspective, to decide if a single frame could represent the spatial and temporal information of a gesture, or multiple frames are needed to get the whole spectrum of a gesture. Gestures from temporal perspective are divided into two main classes:

Static Hand Gestures tend to remain almost unchanged over time and are known as hand postures. These type of gestures are formed from various shapes and orientations without any temporal variation. Static hand gestures are usually recognized base on the shape, orientation, and the spatial location of the hand toward the body. In addition to the arms, fingers and their angle to the palm of the hand, play an important role in identification of the static gestures. Many emblematic such as “okay” signs fall into this category of gestures. Static gestures can be analyzed by single image and do not require a sequence of frames to find the temporal dependence and motion of the hands.

Dynamic Hand Gestures are those types of gestures which the hand moves over time. This type of gestures are composed of a sequence of hand poses with their motion information at each timestamp. The motion trajectory of the gestures in addition to the shape of the hands play and important role in identifying these type of gestures. There are three main motion phases in dynamic hand gestures: *preparation*, *stroke*, and *retraction*. The main message of the dynamic gestures linguistically are stored in the stroke phase. Most of the co-speech gestures as well as non-manual gestures are inherently dynamic. To identify such gestures, a sequence of frames are required to extract the spatial and temporal dependency of hand movements during the gesticulation.

2.1.2 Natural Language Processing (NLP) and Its Role in Gesture Search

The co-speech gestures, as visible by their names, are accompanying spoken language which has been studied widely through auditory and verbal channels. A big chunk of the work on understanding the speech is performed on transcribed text from video or audio which provides insight on correlation of gestures with the uttered words. Natural Language Processing (NLP) is essentially a part of artificial intelligence that models human-machine interactions using natural language.



(a) Lexical hand gesture

(b) Beat gesture

(c) Deictic hand gesture

(d) Symbolic hand gesture

Figure 2.3. Examples of manual hand gestures which communicate special messages(Image credits: Figure 2.3b from `lightfield-studios/stock.adobe.com`, Figure 2.3a from `woodpencil/stock.adobe.com` and Figures 2.3c and 2.3d from `koldunova_anna/stock.adobe.com`).

The aim of NLP algorithms is to convert the language data into a form that computers can understand. Among these algorithms, synthetic and semantic analysis are the main approaches to model these relationships.

Syntax The synthetic analysis refers to the understanding of correlation between the grammatical rules with the natural language. With this type of analysis one can derive a relative understanding from the language via computer algorithms. There are different syntax methods which are frequently used, for example:

- **Lemmatization** is the process of grouping the inflected forms of words together for easier analysis,
- **Morphological segmentation** divides the words into individual units called *morphemes* which are the smallest units of a language,
- **Word segmentation** involves segmenting long texts into smaller units (words) for easier analysis,
- **Part of speech tagging** assigns the specific parts of speech (nouns, verbs, etc.) to individual words,
- **Parsing** is the technique of extracting grammatical structure of a text by analyzing constituent words based on their underlying grammar,
- **Sentence breaking** is the process of determining the boundaries of a sentence in a long text, and
- **Stemming** is a technique which reduces a word to its root that affixes to suffixes and prefixes.

These techniques are used iteratively on an input text to provide synthetic analysis. This reiteration is important as the produced segments can be grammatically correct, while not making any sense.

Semantics Semantics refer to the meaning that is conveyed by a text. The process of understanding this meaning is an unconscious procedure in humans which relies on our knowledge and intuition about the language. However, extracting this meaning by computers requires a different technique which enables them to partly understand the meaning of a language.

To extract the interpretation of the words, there are different semantic techniques, such as:

- **Named entity recognition (NER)** is a technique which categorizes parts of an unstructured text into pre-defined groups, such as names, locations, quantities, etc.,
- **Word sense disambiguation** is concerned with giving meaning to the words in a sentence based on the context. This is specially important because words typically have different meaning, and it is important which of those meanings are used in certain context,
- **Natural language generation** essentially generates texts that describe input data in a human language automatically. As a branch of artificial intelligence, Natural language generation is important in converting large amounts of data into written narrative.

These two analyses are the main components in supporting human machine interactions in textual form.

2.2 Deep Learning Fundamentals

With tremendous advances in data capture and data storage devices, the massive stored information in memory makes analysis of data for different tasks such as natural language processing, computer vision and information retrieval a tedious task which is accompanied with errors. Especially the existing heterogeneous and large-scale collections make the conventional trial-and-error-based methods obsolete. As an alternative, machine learning offers data-driven algorithms and models to analyze and process huge volumes of data. Applications of machine learning models are so tightly integrated in our daily lives that we might not notice the process of data analysis and decision making were done by these methods. Deep learning essentially is a subset of machine learning which produces representations of data through transformations by numerous hierarchical layers. These algorithms represent the input data differently at each of these layers and compute more abstract representations based on the previous layer's output. Such algorithms are called *Deep Neural Network (DNN)* based on the attempt to imitate the function of the human brain in processing the data through different levels of abstraction.

In the following we introduce the main concepts of deep learning which are used in this thesis and review the literature in different subtasks of gesture recognition and retrieval.

2.2.1 Multi Layer Perceptron (MLP)

The Multi Layer Perceptron (MLP) is one of the basic components of an *Artificial Neural Network* which essentially consists of a perceptron [31] which takes multiple inputs $x_1, \ldots, x_n$ and computes the output $(\mathbf{y})$ through $\mathbf{y} = \mathbf{W}^T\mathbf{x} + \mathbf{b}$ where $\mathbf{W}$ and $\mathbf{b}$ are weight and bias (Figure 2.4).

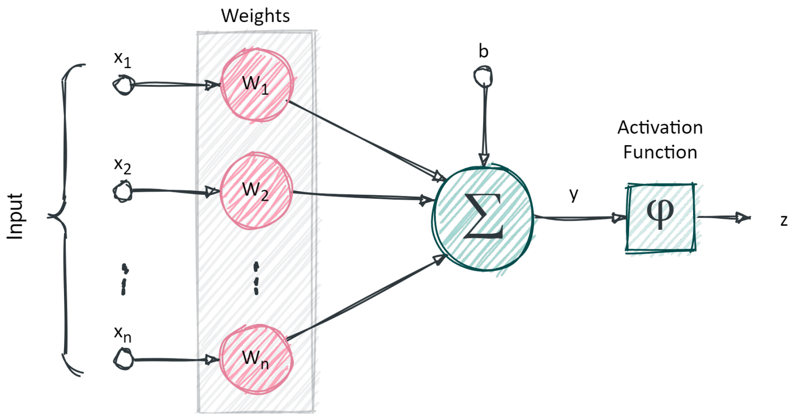


Figure 2.4. The details of a single neuron in MLP: The weighted inputs together with the bias elements pass through the activation function to create the output of a neuron.

The weights indicate the relative importance of the associated input compared to the others. While limited in capacity alone [32], a perceptron can be used as a building block for a more complex model, MLP which is a *feed forward network* consisting of an *input layer*, one or more *hidden layers* and an *output layer*. Essentially, in MLP, each layer's output is the next layer's input and is computed through a linear combination

The output of this linear combination is passed to a nonlinear function, namely *activation function* with the purpose of adding non-linearity to the output of the neurons and impart the capability of processing the complex relationship in data. The output of the activation function $z^{(l)} = \phi\left(y^{(l)}\right)$ is then passed to the following layers. A schema of a MLP is shown in Figure 2.5).

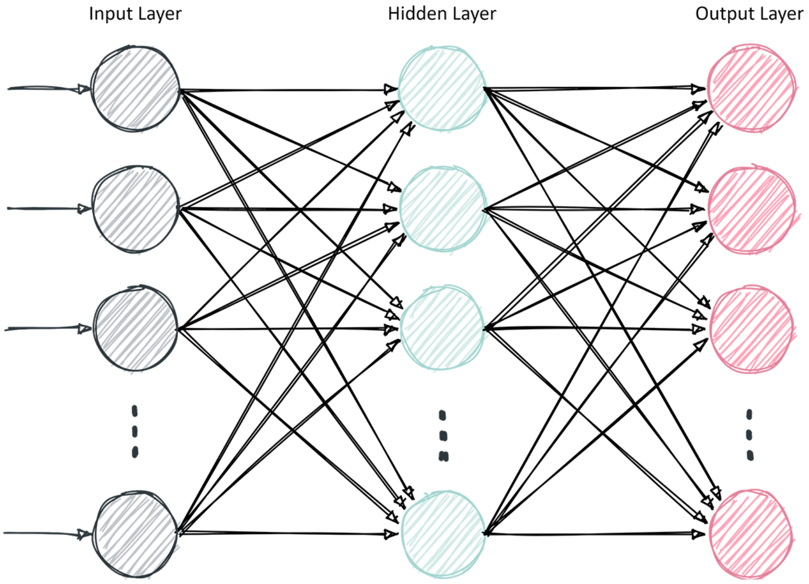
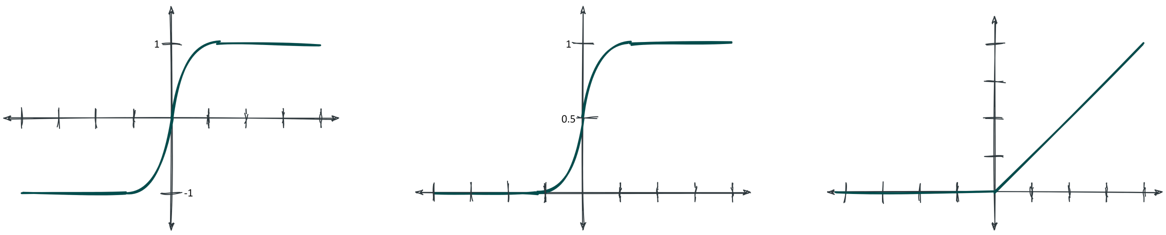


Figure 2.5. An example of the interconnections in a three layer MLP with an input layer, one hidden layer and an output layer where each of these layers consists of multiple neurons.

There are several commonly used activation functions in computer vision such as:

1. **Sigmoid** is specifically used where a probability needs to be calculated since the output of the function is in range of 0 and 1 (Figure 2.6a). The function has the mathematical form as $\sigma(x) = 1/\left(1 + e^{-x}\right)$ and is differentiable and monotonic.
2. **Tanh** is practically scaled Sigmoid and squashes the output to the range -1 and 1. Mathematically it has the form of $\tanh(x) = \left(2/1 + e^{-2x}\right) - 1$. Unlike Sigmoid, it has a zero-centered output, as shown in Figure 2.6b.
3. **ReLU** which is short for Rectified Linear Unit and computes $f(x) = \max(0, x)$ and has the output in range of 0 to ∞ (Figure 2.6c). This function compared to Sigmoid and Tanh is less computationally expensive and shown to accelerate the convergence during training [33]



(a) Sigmoid activation function **(b)** Tanh activation function **(c)** ReLU activation function

Figure 2.6. The illustrations of three of the most common activation functions in deep learning: Sigmoid, Tanh and ReLU.

4. **Softmax** is a generalized logistic function usually used to generate a vector of probability and commonly is used at the end of the network. It is defined as follows:

$$p = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} \text{ where } p_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}. \tag{2.1}$$

The objective of an MLP is to learn the relationship between the input data and the target which is possible thanks to the *back-propagation* algorithm [34]. Back-propagation computes the partial derivatives $\partial\mathcal{L}/\partial w$ and $\partial\mathcal{L}/\partial b$ where $\mathcal{L}$ is the loss function. These partial derivatives are computed one layer at a time and iterated backwards, which ultimately update the parameters to reduce the error between the predicted output and the target. One of the most common optimization algorithms used to train the parameters, is the *gradient descent* algorithm [35]. Employing the gradients is one way to reach the local minimum, which is the least error between the prediction and target. The update at each iteration heavily depends on the value of a hyper parameter called ‘*learning rate* which determines how big or small a step toward the local minima should be. Figure 2.7 shows how gradient descent works and how learning rate could affect the optimization procedure.

2.2.2 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are tightly associated with computer vision due to the seminal work of Krizhevsky et. al [33] which was originally introduced in 1989 [36, 37]. This type of deep learning models relies on the mathematical operations called *convolutions* which are specifically useful for data with grid-like topology such as images. In the context of MLP, a convolution is a linear operation which multiplies a set of weights with an input, which in this case is an array. The weights form a two-dimensional array, which is called filter or kernel. The primary purpose of CNN is to extract features from the input data. CNNs primarily consist of three different types of layers:

1. The Convolution layer performs convolution operations as rectangular grid and computes a linear feature map (see Figure 2.8). Give an output y of previous layer $\ell - 1$ with size

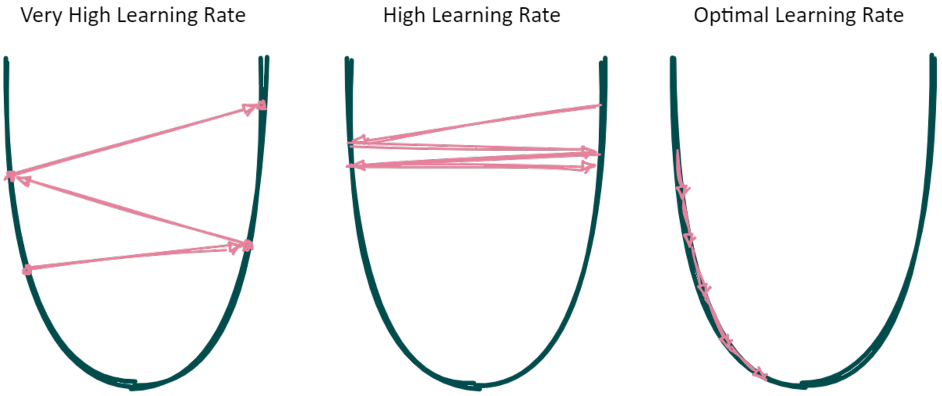


Figure 2.7. A simple explanation on how gradient descent with small and big learning rate works. Selecting improper learning rate during training would cause the divergence from local minima.

$M \times N$, a linear feature map fm^{ℓ}_{ij} is calculated by:

$$fm^{\ell}_{ij} = \sigma(\sum_{a=0}^{m-1}\sum_{b=0}^{m-1} w_{ab} y^{\ell-1}_{(i+a)(j+b)}) \tag{2.2}$$

where w is the learnable filter. Then a non-linear activation layer is applied on fm^{ℓ}_{ij} as:

$$y^{\ell}_{ij} = \sigma\left(fm^{\ell}_{ij}\right) \tag{2.3}$$

with $\sigma(.)$ being one of the activation functions explained above (see also Figure 2.6). In addition to the dimension of the convolutional filter, *stride* denotes the number of pixels the filter window moves after each operation. Here an example of a two-dimensional convolution with stride $s = 1$ is shown in Figure 2.9. The convolutional filter dimension

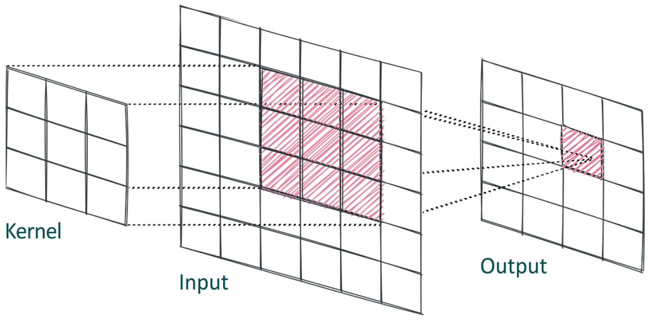


Figure 2.8. 2D convolutional layer in a nutshell: The 2D Kernel moves as a sliding window over the input and generates the output.

is not limited to two and can be generalized to three and one case (Figure 2.10). A

1D convolutional filter slides along one dimension, such as audio and text data [38]. Three-dimensional convolutional filter is important for 3D input data, such as videos [1].

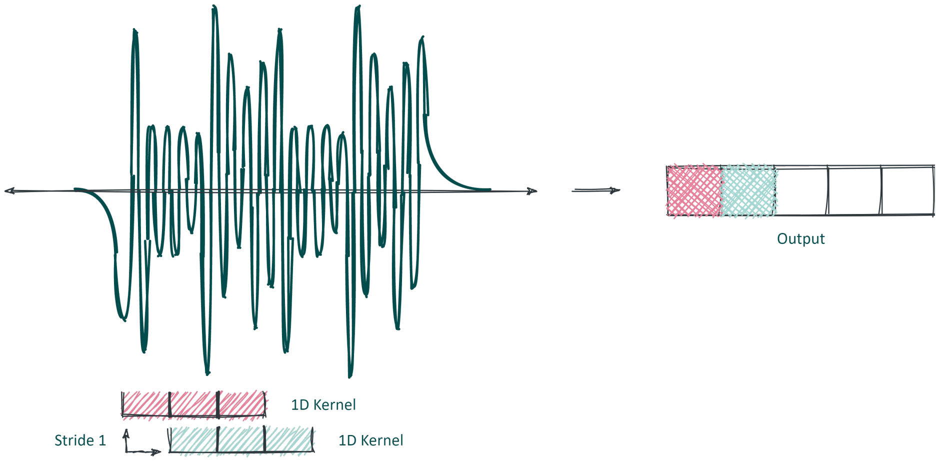


Figure 2.9. 1D convolutional layer in a nutshell: The 1D Kernel window scans the input with the stride one and generates the output elements.

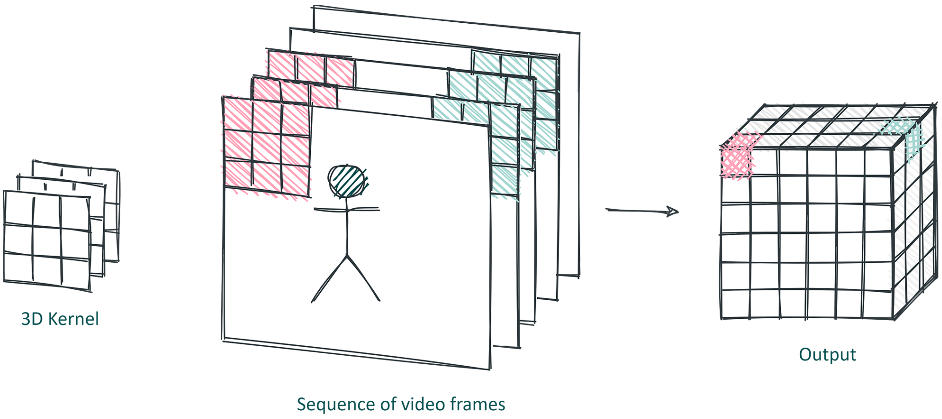


Figure 2.10. 3D convolutional layer in a nutshell: The 3D Kernel is applied on the 3D input such as video and generates the output.

2. The Pooling layer or down-sampling layer reduces the spatial size of the feature maps generated by the convolutional layer to decrease the computational complexity of the network as well as introducing spatial invariance. Max and Average pooling are most common pooling layers used in CNNs where the maximum and average value of the filter window are selected respectively. A Pooling layer also can be generalized to three dimensions, Figure 2.11 visualizes the pooling.

3. The Fully Connected (FC) layer is essentially an MLP used to convert the two dimensional feature map to a one-dimensional vector. This layer usually is used towards the end of the network and can be used as the optimization of prediction of the target.

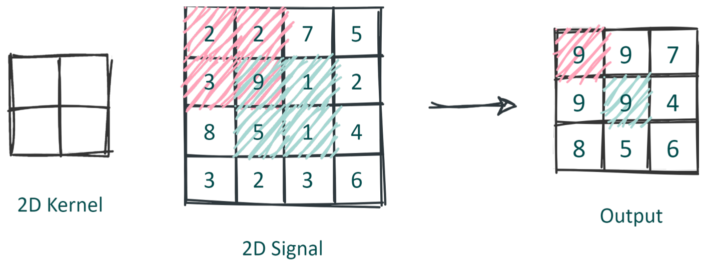


Figure 2.11. 2D Maxpooling as one of the most common pooling operations: The output of each Kernel is the highest element in the Kernel frame.

Using these layers, a CNN transforms data – for example an image – to a prediction which for example could be a class score. The convolution layer and fully connected layer's parameters will be trained with *gradient descent* which minimizes the output error. Figure 2.12 shows an example of a CNN (Alexnet) which is used for handwritten digit recognition. The network consists of five convolutional layers and three FC layers. As it can be seen, the output of the layers at the beginning of the network are more representing the low level features such as edges, while the deeper the network gets, the representations become closer to the actual target.

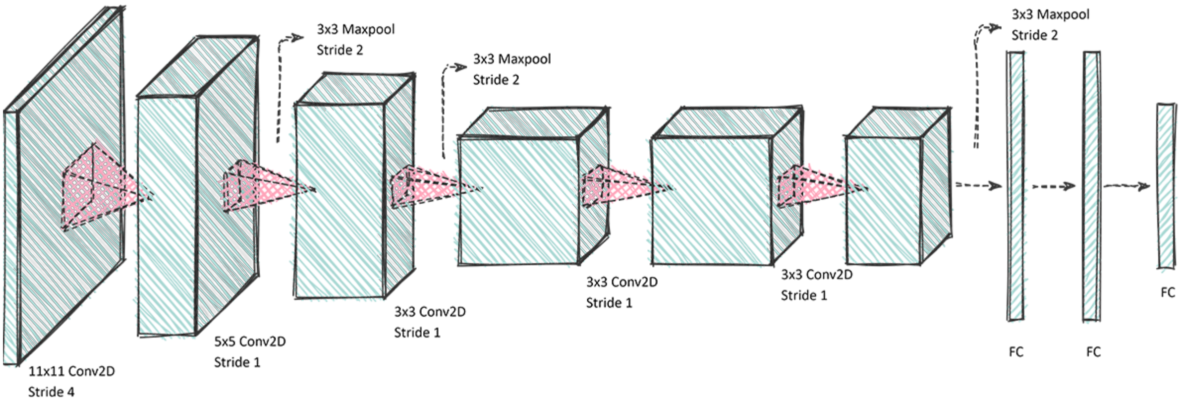


Figure 2.12. An overview on the Alexnet [1] architecture diagram consisting of convolutional, pooling and fully connected layers.

2.2.3 Recurrent Neural Network (RNN)

RNNs are another type of neural network with the ability to *remember* the input. The history of the algorithm goes back to 1980's [34] but the actual potential of them was discovered only in the recent decade with increase of computational power and data availability. These networks with an internal loop-wise connection, allow the information to persist inside the network and be especially helpful in tasks such as speech recognition and sequence modeling. One of the special types of RNN is LSTM [39] which has proven useful for computer vision tasks with sequential data. The original idea of RNNs could infer a prediction based on recent information, however, once the information for this prediction is not as recent anymore, the RNN fails to connect the information. In other words, the gradients which are used to update the weights according to the predicted error, diminishes when propagated backwards through the network. This problem is referred to as *vanishing gradients* and makes the network harder to train. LSTM is specifically designed to overcome this situation by remembering long periods of information. RNNs consists of a repeating module which can be a simple structure of a neural network (Figure 2.13).

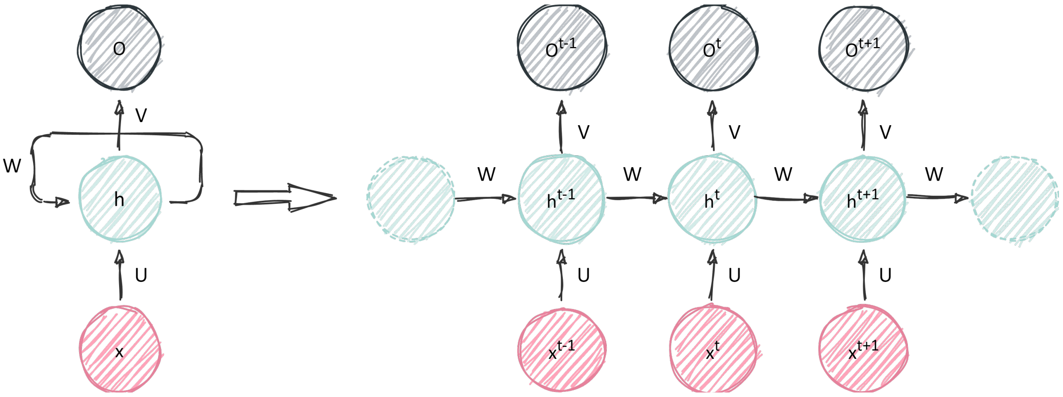


Figure 2.13. A simple structure of RNN. Each hidden layer at each timestamp *t* generates an output as well as input to the next hidden layer.

However, in LSTMs this neural network structure is rather different and more complex, as shown in Figure 2.14. It contains different *gates* to ensure access to the memory, either by adding or removing it. The three gates namely input, forget and output, allow LSTM to let new input, delete the information or let it impact the output. These gates, practically, are layers of neurons with Sigmoid activation functions which in essence are differentiable and enable the training through back-propagation.

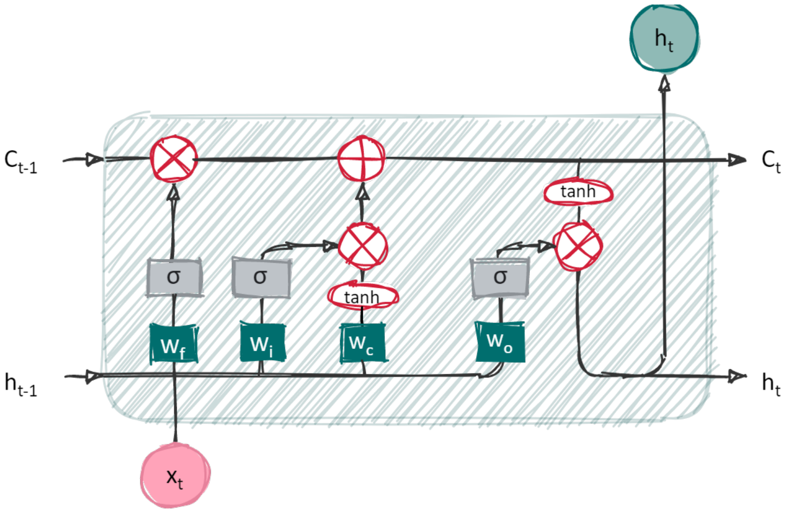


Figure 2.14. Architecture diagram of LSTM unit containing the input, forget and output gates.

The information arriving at a LSTM unit first passes through a forget gate. The hidden state of the previous timestamp $\mathbf{h}_{t-1}$, together with the input at the current timestamp $\mathbf{x}_t$ are fed to this gate which is essentially a layer of neurons with Sigmoid activation function to output a value between 0 to 1, where the latter is the pass ticket to cell state $\mathbf{C}_{t-1}$. In other words, the cell state is the memory of an LSTM unit while the hidden state is the output of the unit. Mathematically speaking:

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f\right) \tag{2.4}$$

where $\mathbf{W}_f$ is the weight matrix, $\mathbf{b}_f$ the bias and $\sigma(.)$ is the Sigmoid function. The next step is to indicate the new information which is going to be added to the cell state via the input gate. Here also layers of neurons with Sigmoid function, $\mathbf{h}_{t-1}$ and $\mathbf{x}_t$ as input decides if the values are updated. Then, a new vector is created to be added to the cells state:

$$\begin{aligned}\mathbf{i}_t &= \sigma\left(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i\right)\\ \tilde{\mathbf{C}}_t &= \tanh\left(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C\right)\end{aligned} \tag{2.5}$$

where $\tilde{\mathbf{C}}_t$ is the vector of new values,$\mathbf{W}_i$ and $\mathbf{W}_C$ are weights matrices and $\mathbf{b}_i$ and $\mathbf{b}_C$ are biases vectors. At this stage, $\mathbf{f}_t$, $\mathbf{i}_t$ and $\tilde{\mathbf{C}}_t$ are used to update the cell state $\mathbf{C}_{t-1}$ to $\mathbf{C}_t$ as:

$$\mathbf{C}_t = f_t * \mathbf{C}_{t-1} + i_t * \tilde{\mathbf{C}}_t \tag{2.6}$$

where $*$ is the Hadamard product. The last step at the LSTM unit is to produce the next hidden state which is dependent on the updated cell state as well as $\mathbf{h}_{t-1}$ and $\mathbf{x}_t$:

$$\begin{aligned}\mathbf{o}_t &= \sigma\left(\mathbf{W}_o\left[\mathbf{h}_{t-1}, \mathbf{x}_t\right] + \mathbf{b}_o\right)\\ \mathbf{h}_t &= \mathbf{o}_t * \tanh\left(\mathbf{C}_t\right)\end{aligned} \tag{2.7}$$

where $\mathbf{o}_t$ is the output of the output gate, $\mathbf{W}_o$ is the weight matrix and $\mathbf{b}_o$ is the bias vector. There are different improved types of LSTM such as bi-directional LSTM which enables the network at every point to have complete sequence information about all points either before or after it [40].

2.2.4 Attention Mechanism

One of the models where RNNs are prominently used is in the *seq2seq* model which transforms a sequence of input to a sequence of output, both with arbitrary length. This model originally was introduced for language modeling [41] and extended to machine translation and computer vision as well. Such models have an encoder to produce an embedding and decoder to generate the output sequence which are both RNN models. While *seq2seq* models are increasing the RNN capabilities, the memory that they can access from a context is not unlimited. Therefore, in long sequences of input important information would get lost. To reduce the impact of loss of information in long sequences, the concept of looking at different elements of the sequence was suggested by [42] which is the equivalent notion of what we call *attention* today.

Although there is an explicit way of forcing the network to focus on a part of a sequence and weigh its sensitivity to the input, we should not forget that very deep neural networks are already a form of implicit attention [43]. A trained model over different layers learns to ignore some parts of the input or prefer some parts to the others, for example in images, human bodies or poses [44].

The *explicit attention*, which from now on will be referred to as *attention*, has two main types: soft and hard attention. In the encoder-decoder RNN, given a source sequence $\mathbf{x}$ of length n, output sequence $\mathbf{y}$ of length m,

$$\begin{aligned}\mathbf{x} &= [x_1, x_2, \ldots, x_n] \\ \mathbf{y} &= [y_1, y_2, \ldots, y_m]\end{aligned} \tag{2.8}$$

the hidden states of the encoder and the decoder are $\mathbf{h}_i$ and $\boldsymbol{s}_t = f\left(\boldsymbol{s}_{t-1}, \boldsymbol{y}_{t-1}, \mathbf{c}_t\right)$ respectively. The attention score is calculated based on the alignment model of the pair of (y_t, x_i):

$$\alpha_{i,j} = \text{align}\left(y_t, x_i\right) = \frac{\exp\left(\text{score}\left(\boldsymbol{s}_t, \boldsymbol{h}_i\right)\right)}{\sum_{j'=1}^{n} \exp\left(\text{score}\left(\boldsymbol{s}_t, \boldsymbol{h}_{j'}\right)\right)} \tag{2.9}$$

The set of $\alpha_{i,j}$ are the attention weights defining how much of the input hidden state should be considered in the output. There are different alignment score functions which are being used in different applications. One of the common functions are additive [45] which is defined as following:

$$\text{score}\left(\mathbf{s}_t, \boldsymbol{h}_i\right) = \mathbf{v}_a^{\top} \tanh\left(\mathbf{W}_a\left[\boldsymbol{s}_t; \boldsymbol{h}_i\right]\right) \tag{2.10}$$

where $\mathbf{v}_a^{\top}$ and $\mathbf{W}_a$ are weight matrices to be learned.

2.2.5 Transfer Learning

An important property of deep learning methods is their ability to generalize a model designed for one task, to another one [46,47]. One of the popular methods is to use a pre-trained model as a starting point for a different task and help the model to generalize to the new task [48]. Transfer Learning is a technique to fine tune previously trained DNN parameters for a new target task. Practically, this method only works when the model features of the first task is general and can be optimized for the second task [46].

The key for a transfer learning method to work, is the careful selection of the source task and the dataset which the model is trained on. The selected source task should have some similarity to the task in hand, to be able to use the learned representations. For example, a CNN trained on cars, can be used to detect trucks as well. Additionally, the dataset that the original task is trained on, should be large enough to help to enable the model to generate general and diverse representations.

The process of transfer learning starts with a pretrained model. This model can be either trained from scratch or be of the existing models available from other researches on different datasets. When reusing this model, the objective needs to be adjusted to the task in hand. The features generated are more generic in early layers and they become more specific the deeper the network gets. Therefore, it is important to change the last layers and adjust them to fit our objective and train them. As a rule of thumb, if the new dataset does not have more than a few samples per class, only the new layers are re-trained and the rest of the network remains untouched. However, if the second dataset contains a large number of samples per class, the whole model will be retrained, using the initial weights from the original model.

2.3 Information Retrieval

Information retrieval is a broad term referring to the methods which are designed to obtain a subset of information from a collection of documents. The process starts with a user specifying a *query* and the information retrieval model searches the *database* of stored documents to find the items *relevant* to the query and match its specifications. The *documents* in an information retrieval can have any modality. The earlier information systems were mostly concerned with text, as the means to search in the written documents, however, the modern information retrieval has extended this to other media types and modalities.

The search process ideally would find the exact match to the specified query , however, in multimedia information retrieval, this is not always possible. Therefore finding the *similar* documents is the next best result of the process. While similarity sounds like a simple concept because we use it in our everyday life, when it comes to information retrieval, it gets ill-defined [49]. When talking about actions, one might think skiing on snow or grass are similar, since both share almost the same equipment, however for someone else they would be two completely different actions. Due to this ill defined notion of similarity, it is most useful to quantize the similarity with a certain function, so it can be comparable.

In the following we explain the retrieval mechanism, specifically how the query is formulated and encoded. Additionally we introduce how the similarity is quantized and learned by deep neural networks.

2.3.1 Query Formulation

A query in an information retrieval system is the expression of the mental image one has for the desired search result. Depending on the capabilities of the retrieval system, and the information needed, the query can have different forms.

Keyword-based queries is a type of query in which the user describes the desired information using text and the results will be a textual subset of the collection. As an example, one might be interested in reading the manual of a camera, and would search the stored documents by the keyword “Nikon D7500”. Once the collection is not textual itself, the need for some sort of meta-data or annotations would be necessary. Following the same example, one might be interested in finding the pictures taken by “Nikon D7500” in a gallery which is stored in the meta-data of the images [50].

The annotations for the documents can be manually or automatically generated for the specific type of document. For example in gesture video collections, linguist experts would go over the documents and describe the gestures within the video. However, thanks to the recent developments of computer generated text for images and videos, one can generate the annotations automatically [51]. An example for such a situation would be a picture where the camera body is shown, and the generated caption would include the brand and the model written on the device.

Query by example is another way to formulate the queries which could possibly reduce the semantic gap which mostly occurs in translating one's mental perception of the desired information to the query domain. Essentially the query is an example of the desired information with the same modality [52]. For example in this case one could search for the Nikon D7500 camera by using the picture of this device as an example.

Query by Sketch is yet another type of query formulation where the user tries to illustrate the mental image of the desired information as a sketch. This type of query also can potentially lower the difficulties of expressing one's idea about the query [53].

Others: The query formulation does not limit to the three above-mentioned categories and can be extended to the need when developing an information retrieval system. For example one can express the music they are looking for by singing or by performing an activity such as jumping to search for similar action.

2.3.2 Vector Space Mapping

Once the query is formulated, the next step is to find a common representation between the modalities of the query and the collection to find the similarity. This representation modeling, which also is referred to as encoding, is a step to map the information from one domain into a vector space. The generated vector is used to describe the document. This mapping is done using inherent features of that domain, for example in textual documents word frequency is used to represent the document in vector space [54]. This representation in vector space depends on the type of media, for example for images, color features can be used to describe the document. Alternatively, the previously mentioned feature extraction methods based on deep learning can be used in this step as well.

2.3.3 Similarity Comparison

Once the collection is represented in vector space based on some defined features, they are stored in a database and will be used for the comparison. When the query is formulated, it also is represented to the feature space, analogous to the collection space, and the vectors are used to find the relevant instances in the collection. This comparison is done via a distance metric in the vector space, and according to the distance, the similar items of the collection would have a smaller distance to the query.

In addition to comparing the features which are extracted from the query and the collection solely based on the inherent characteristic of them, There are *similarity learning* methods which map the documents to a feature space according to their similarity with other documents in the collection. This method of extracting features would enhance the result of similarity comparison and retrieval.

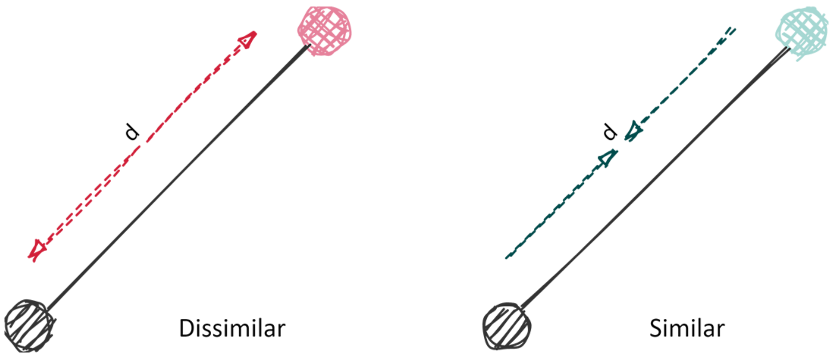


Figure 2.15. Similarity preserving in embedding via contrastive learning. The similar pairs are mapped in closer distance points and the dissimilar pairs will have larger distance from each other in vector space.

Deep metric or similarity learning is a task in different domains, including computer vision, which aims to learn a similarity function to identify how similar two instances are. This learning procedure contains three main steps:

- Feature extraction which projects the data into an embedding. This step sometimes is referred to as encoding phase and the extracted feature is the latent vector. This part could be a CNN or RNN architecture, depending on the input data. There are two main architectures to generate such latent vectors: Siamese network [55] and Multimodal Auto encoders [56].
- Comparison with the collection according to a distance metric. The distance metric used for this comparison is usually the Manhattan distance or Euclidean distance:

$$\text{Manhattan}\left(e^{(1)}, e^{(2)}\right) = \sum_i \left| e_i^{(1)} - e_i^{(2)} \right| \tag{2.11}$$

$$\text{Euclidean}\ \left(e^{(1)}, e^{(2)}\right) = \sqrt{\sum_i \left(e_i^{(1)} - e_i^{(2)}\right)^2} \tag{2.12}$$

- Classification of the results. The distance measure determines which two pieces of data are *similar* or *dissimilar*. This can be done by setting a threshold and labeling the distances above the threshold as dissimilar and vice versa which generates a ranked list based on the similarity differences between the data points. A more efficient way is to use a classifier such as logistic regression to learn this similarity.

In the following we explain the specifics of the Siamese network and different types of it. Siamese network architectures are essentially two identical networks which share the parameters and weights. In contrast to classification networks whose objective is to predict a target label for a data point by minimizing a cross entropy loss function, in Siamese networks other loss functions such as contrastive loss and Triplet loss are used.

Contrastive loss is a distance based loss function used to learn the embeddings in a way that two similar data points have smaller Euclidean distance than two dissimilar data points [57] (Figure 2.15). The loss function in this case is defined as:

$$\text{loss}(d, \text{sim}) = \frac{1}{2} \times \text{sim} \times d^2 + (1 - \text{sim}) \times \frac{1}{2} \times \max(0, m - d)^2 \tag{2.13}$$

where d is the distance between the feature vector of an output of a network, sim is the similarity label of the input and m is the margin. In the training process, the embedding of the dissimilar inputs are learned in a way that the distances between them are larger than margin m. In this case, $Y = 0$ and the loss function will be:

$$\text{loss}(d, 0) = \frac{1}{2} \times \max(0, m - d)^2 \tag{2.14}$$

However, for the similar data points, the embeddings are learned to have smaller distance. In this case, $Y = 1$ the loss becomes:

$$\text{loss}(d, 1) = \frac{1}{2} \times d^2 \tag{2.15}$$

Triplet loss Differently from contrastive loss, the triplet loss function takes three input data at each time to compare their distances [58]. In contrast to contrastive loss, where the data points were randomly chosen, the samples for triplet loss are intentionally chosen from similar or dissimilar data points: *Anchor* is the point of comparison between two other samples, a *positive* sample $\mathbf{s}^+$ which is the data point similar to the anchor and a *negative* sample $\mathbf{s}^-$ which is the dissimilar data point to the anchor. The loss function in this case is defined as:

$$D(\mathbf{s}, \mathbf{s}^+) < D(\mathbf{s}, \mathbf{s}^-) \tag{2.16}$$

$$\mathcal{L}(\mathbf{s}, \mathbf{s}^+), \mathbf{s}^-)_{\text{triplet}} = \max(D(\mathbf{s}, \mathbf{s}^+) - D(\mathbf{s}, \mathbf{s}^-) + m, 0) \tag{2.17}$$

where $d_1 = D(\mathbf{s}, \mathbf{s}^-)$ is the distance between the anchor and the negative sample and $d_2 = D(\mathbf{s}, \mathbf{s}^+)$ is the distance between the anchor and the positive sample and m is the margin constant as shown in Figure 2.16. The training process involves minimizing the distance between the anchor and positive sample, and maximizing the distance between the anchor and negative sample.

2.3.4 Retrieval

Either of the two above mentioned methods, or any other feature extraction can be used as a similarity preserving encoding method to populate the database for the retrieval. Once these features are available, the query is also mapped to the same space and according to the distance to the features in the database, the closest instances to the query are retrieved. These retrieved results would include *true positives* if they are representing similar content to the query or *false positives* when they are not related to the query. Depending how big is the data collection, and how much of the results are selected to be shown to the user, there will be *false negatives*, which are the results related to the query but not shown to the user

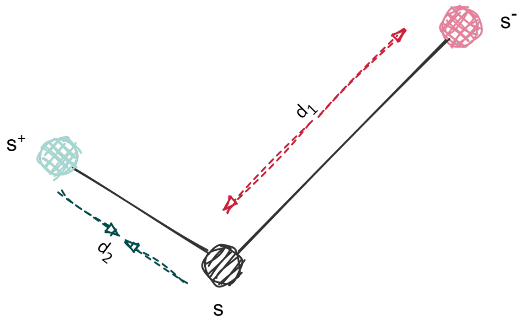


Figure 2.16. The concept of triplet similarity learning: during training, the network tries to get the anchor and positive sample closer to each other, and at the same time anchor and negative sample further from each other.

or are ranked at the bottom of the list, or *true negatives*, which are actually not related to the query and should not be shown to the user.

According to the number of true positives, and the ranking presented by the retrieval system, one can judge the performance of a retrieval system, which is a topic to be discussed in the evaluation chapter.

Chapter 3

Related Work

In this chapter we explore the background research in the areas related to gesture recognition and retrieval. Specifically, in Section 3.1 we review the existing literature in vision-based hand gesture recognition in computer vision including various preprocessing methods (Section 3.1.1), and the techniques for classifying and identifying gestures (Section3.1.2). In Section 3.2 we explore the research done in the field of retrieval. Although there is numerous literature in hand and body posture understanding, retrieving these gestures, which inherently have different challenges compared to recognition, has not been as much in the spotlight and remained an under-explored field in computer vision. In Section 3.2.1 we review the related work in the field of video retrieval (because of the common media type with gesture retrieval) and explore the methods and challenges in this task.

3.1 Vision-Based Gesture Recognition

Vision-based hand gesture recognition is mostly referred to as finding the re-occurrence of spatio-temporal patterns representing hand gestures. The recognition of dynamic hand gestures necessitates a different approach than static gestures, as in the former there is a temporal dependency between the gesture frames, and one keyframe cannot represent a gesture. A gesture recognition system consists of three main components; Preprocessing, Feature Extraction and Identification. In this section we explain the role of each component and explore the literature in each module.

3.1.1 PreProcessing techniques

The main difference between actions and gestures are the body parts being involved. In the actions, usually the whole body is performing the action, but in gestures, tend to be localized and performed by specific body parts. The action recognition methods are highly dependent on the environment in which the action takes place as well, however, for gesture recognition, the content of the gesture is not related to the environment. Therefore using the action recognition methods for the gesture recognition task results in poor performance.

3.1.1.1 Instance Segmentation

Image classification in computer vision is the task of specifying a class label for an image based on what is visible and shown. For example Figure 3.1 illustrates a classification algorithm which tags the image as car. While classification can answer the question *what*, it cannot answer the question *where*. Object detection is another task which specifically is designed to answer the question *What is where in the image?*. Object detection locates the presence of objects in an image or sequences of images (videos), based on the label of the object using bounding boxes around the object of interest.

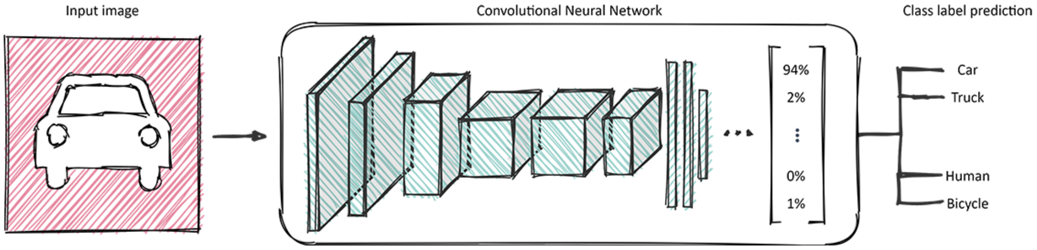
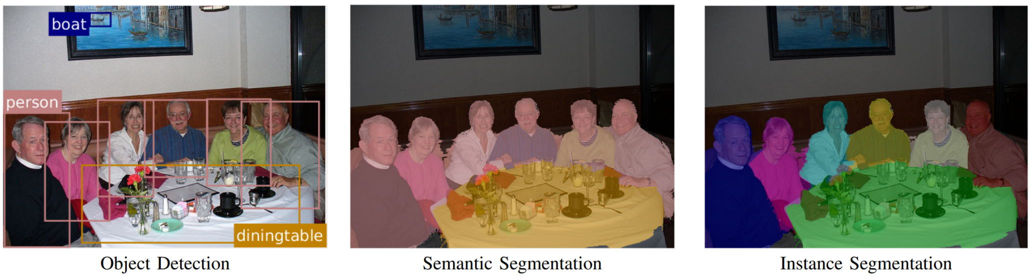


Figure 3.1. Using deep learning for classification tasks. the output of the network is a prediction of a label with a probability score.

Object Instance Segmentation

While object detection methods can locate and label single or multiple objects in an image, image segmentation identifies every pixel in an image with a specific label. Generally speaking, there are two main approaches with major difference in image segmentation as shown in Figure 3.2:

- **Semantic segmentation** refers to the process of identifying each pixel with a class label. By making dense prediction over an entire image, semantic segmentation creates a fine-grained mask over all instances of one object.
- **Instance Segmentation** – in contrast to semantic segmentation – will identify pixels with the same label for each instance of an object. In other words, if there are multiple instances of the same class in an image, instance segmentation would separate them.



Object Detection Semantic Segmentation Instance Segmentation

Figure 3.2. Comparison of object recognition, semantic segmentation and instance segmentation (Image from [2]).

Both approaches of image segmentation have numerous applications and there is a rich literature using different methods to accomplish these tasks with higher accuracy. Traditional image segmentation algorithms used histograms [59], edges [60] and clustering [61] among other methods for pixel-wise identification of objects and scenes. Despite their simplicity, memory efficiency and speed, the conventional methods of image segmentation require a lot of specific tuning and have limited application in complex scenarios.

Deep learning based approaches for image segmentation serve as a function, where the input image goes through a process of feature extraction using a model which is trained to identify which features of an image are important for this task. A general Image segmentation algorithm can be thought of as an encoder-decoder architecture where the encoder is responsible to extract discriminative features and identify the objects with their locations, and the decoder projects this representation to the pixels and maps it to the regions of the image.

Region based Methods One of the widely used approaches for semantic segmentation are *region* based methods with *segmentation through recognition* approach. The pipeline involves selecting regions of an image and performing feature extraction on each region, and finally classifying them. The region mask usually will be projected on the original image by labeling pixels according to the highest scoring region (see Figure 3.3).

R-CNNs [62] as one of the prominent methods in this group extracts multiple region proposals using selective search [63] and employs CNNs as the feature extractor of each proposal. The extracted features which have a fixed size –thanks to affine image warping– are fed to a category-specific linear classifier. Although R-CNN can be built on top of different CNN architectures and significantly benefits from the discriminative CNN features, it specifically suffers from the lack of spatial information in the features as well as region boundary generation.

Fast R-CNN [64] is a successor of R-CNN for object detection which uses *RoIpool* to generate one Region of Interest (RoI) per image by extracting a fixed length vector from the feature map. This extension considerably improved the accuracy and speed of the algorithm. While specifically designed for object detection, it found a base for image segmentation methods such as MultiPathNet [65] which uses segment candidate proposals from [66] to refine the RoIs from Fast R-CNN.

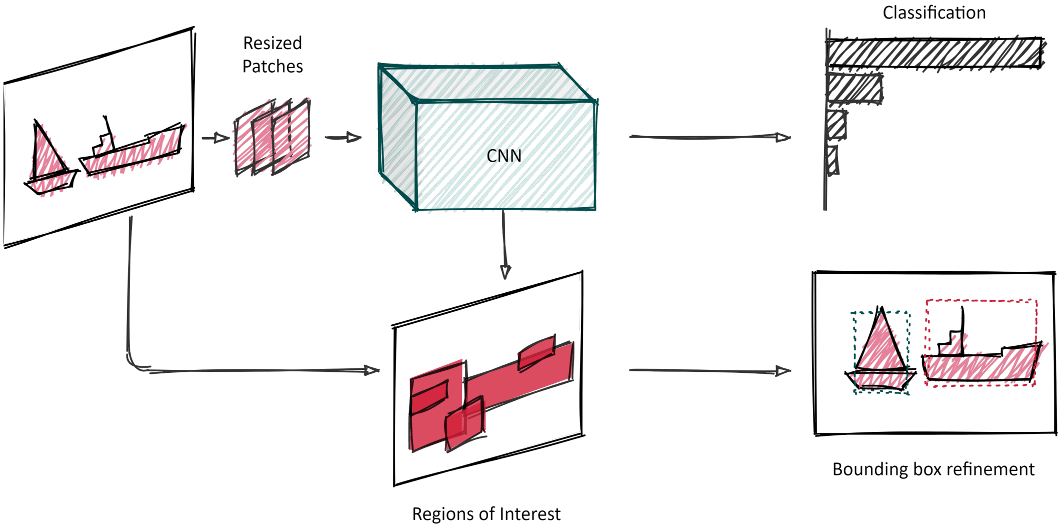


Figure 3.3. The architecture diagram of R-CNN

As another extension to the R-CNN family for object detection, Faster R-CNN [67] uses Region Proposal Networks (RPNs) instead of selective search to extract object proposals directly from CNN feature maps. The RPN uses a sliding window to detect anchor boxes, and predict the probability of the box containing an object, and if yes adjust the bounding box to fit the object.

Fully Convolutional Network (FCN) based methods Another popular line in the image segmentation field is FCN-based approaches. These methods learn a pixel to pixel mapping circumventing RPNs. Originally, CNNs required fixed size image input due to the fixed Fully Connected (FC) layers. FCNs overcomes this limitation by removing the FC layers and using only convolutional and pooling layers which gives the network the flexibility to infer predictions based on arbitrary sized inputs. One of prominent methods using FCN is Mask R-CNN [68] which is an extension of Faster R-CNN to predict the segmentation mask. This method essentially is using a FCN [69] on top of the feature extraction and generates a binary matrix with 1 on locations where the pixel belongs to the recognized object and 0 elsewhere.

One of the main drawbacks of the earlier works with FCN was the low resolution of the object boundaries due to the multiple downsampling and pooling layers in FCN. To overcome this issue SegNet [70] uses an upsampling technique by storing the max-pooling indices which results in improved segmentation boundary delineation. Unet [71] is another alternative to overcome the information loss in FCN. The authors propose to have direct connections to the upsampling layers directly before pooling layers. The additional information about the details of the input image will result in fine grained segmentation boundary prediction.

Human Instance Segmentation

Human Instance segmentation is a subcategory of Instance segmentation, which due to the human body characteristics, and the human interactions, imposes challenging difficulties. Especially, in interactions where the co-speech gestures mostly happen, the occlusion and multi person scenarios are pronounced challenges. Using pose estimation for tailoring the segmentation map to the human body is one of the popular multi person segmentation methods. There are two types of pose estimation approaches:

- *bottom-up* approach, detects body key-points and then generates the segmentation mask based on these key-points. Next, it groups the key-points into body joints to generate body pose estimation [72]
- *top-down* approach uses human detection methods to obtain proposals followed by pose estimation for single person [73, 74].

Multi-person scenario: In this case the bottom-up approaches are often adopted to generate instance proposals. Tripathi et al. [75] in Pose2Instance condition semantic segmentation on human key-points and generate a pose-instance map, even in occluded situations. PersonLab [76] also follows the same process by associating semantic segmentation of humans and human poses in the entire image by forming a geometric embedding. However, both of these methods highly depend on the performance of the joint grouping which makes them comparable with generic semantic segmentation methods. PoSeg [77] eliminates the low recall rate effects of segmentation based methods, by using local and global refinement blocks to tailor the segmentation map to the human pose.

Pose2Seg [8] is another bottom-up approach using body key-points to estimate the pose and then transform them by the affine-align operation to map to pose templates. This approach extracts the spatial features from the detected human instances in the video frame and skeletal features from pose key-points. The skeletal features are formed by the Part Affinity Fields [10] and confidence maps which indicated the pairwise relationship of the body parts and the probability of the existence of the body part in the predicted location, respectively. Lastly, the segmentation model named *SegModule* is applied which creates final masks of human segmentation.

3.1.1.2 Temporal Localization

Similar to the object segmentation, segmenting and localizing a specific action in video requires specific search and is crucial for fields such as action and gesture recognition. In videos containing gestures, there is a sequence of frames which form a gesture and often times, during a conversation, other gestures as group of frames will follow. This is illustrated in Figure 3.4. To correctly recognize these gestures, we need to accurately localize and segment them temporally.

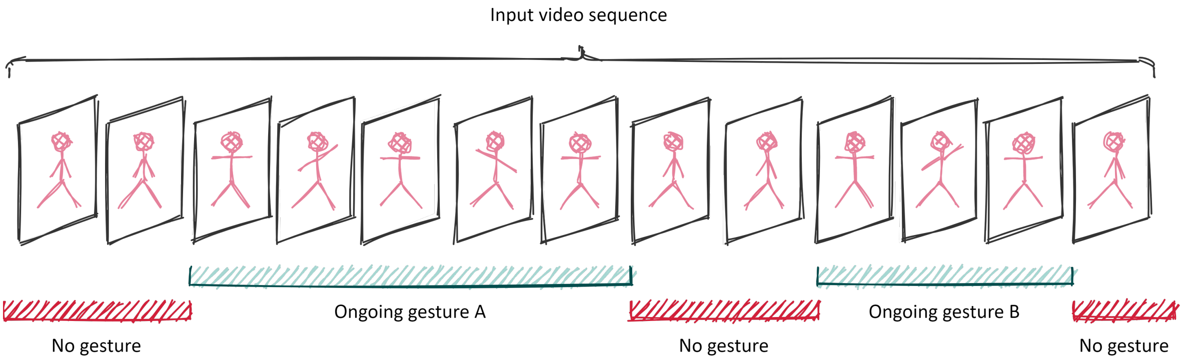


Figure 3.4. A simple illustration of temporal localization task where there are sequences of video without gestures.

The temporal localization methods are tightly connected with action detection and many approaches exhaustively tackle this problem in two stages: classifying and sliding window. The simplicity of implementation made these methods a popular approach in action and gesture recognition tasks. However, covering all the video with different window sizes, imposes computational costs and are considered inefficient. Another category of two stage methods is focusing on generation action agnostic temporal proposals and pass these trimmed clips to classifier for prediction. A temporal action proposal is a trimmed video clip which may contain events defined by start and end of the clip.

Anchor-based methods: Anchor-based methods generate a set of proposals using multi-scale anchors. Segment Convolutional Neural Network (S-CNN) by Shou et al. [78] used multi-scale anchors with regular temporal intervals and passed them to a C3D [79] to generate a binary classification for generating the proposals. However, the high overlap between the anchors are computationally expensive and as long as they are not high enough, the temporal proposal boundaries remain inaccurate [80]. Temporal Unit Regress Network for Temporal Action Proposals (TURN-TAP) [81] inspired by faster R-CNN overcomes this problem by constructing clip pyramids and performing temporal coordinate regression on adjacent windows.

In contrast to S-CNN which needs video inputs of the same length due to use of CNN, Temporal Actionness Grouping (TAG) [82] proposed a flexible input length temporal action proposal method based on binary classification using Temporal Segment Network (TSN) [83]. Once the classification based on the *actionness* –the likelihood of containing a generic action– is finished, high scored snippets are grouped and form a trimmed video clip. Since this method is highly dependent on the binary classification, in case of error in this part, parts of the temporal proposal would be missed.

Boundary-based methods: Boundary-based methods eliminate the need for a sliding window for temporal localization. Single Stream Temporal Action Proposals (SST) [84] generate action proposals using a single stream visual and sequence encoder. Gao et al. [85] and the Boundary Sensitive Network (BSN) [86] predict the start and end of an action interval using local to global fashion, to generate precise boundaries. Lin et al. further improved BSN [87] by confidence evaluation of densely distributed proposals.

Depth-based methods: The Temporal Action Proposal methods are widely used in gesture localization in long videos as well. However, due to the existence of different modalities, such as depth in gesture videos, some other methods are prominently used as well. One of the most used methods in gesture spotting is based on the assumption that the hands, after finishing one gesture, return to their resting position. Based on this assumption Lie et al. [88] used the visual cues of location of hands to temporally segment the gestures. Another simple yet popular method is using QOM measure to locate the hands when they complete the gesture and return to the origin. Wang et al. [89] segmented the continuous gestures into isolated ones by QOM measure for depth data. Cihan Camgoz et al. [90] used a silence class to classify continuous gestures jointly with training a C3D by extracting windows from a prior distribution of gesture-probable regions. FOANET [91] uses the temporal fusion over the CNN geatures generated by sliding window over video frames.

3.1.2 Gesture Recognition Methods

A gesture recognition system comprises three main modules as shown in Figure 3.5:

- Preprocessing: which prepares the input the data for the feature extraction,
- Feature extraction: which maps the input gesture sequence to an embedding space,
- Classification: which maps the extracted features to category labels.

Among these three, the main component of a gesture recognition is feature extraction and encoder. There are numerous methods which specifically are designed to tackle the challenges of video recognition and specifically hand gestures and extract the discriminative features. The approaches are mainly divided into pre and post-deep learning eras.

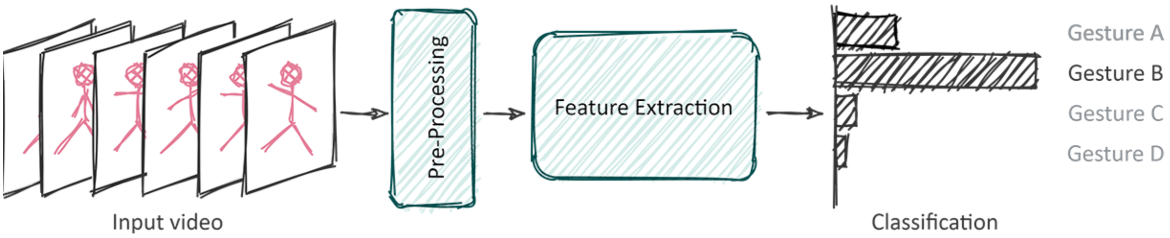


Figure 3.5. A general gonfiguration of network for gesture recognition task.

3.1.2.1 Pre-Deep Learning Era

Many pre-deep learning methods used the appearance of the hands to extract the image features. Features are meant to represent information about gesture position, orientation and temporal progression. In order to identify the gesture, these features were compared with the features extracted from the collection using a pattern classification module [92]. Many conventional models for gesture recognition use a separate feature extraction component followed by a classifier to identify the hand gestures.

Color-based feature extraction: The appearance based feature extractors depend on the visual cues of hand gestures in 2D images. Color features is one of the most common representations used in gesture recognition which are extracted based on the hand skin color and due to their fast and simple implementations [93, 94] were a popular approach. However, their sensitivity to the lighting conditions as well as robustness in the identification of hands when skin color objects are present in the image frame, limits their usability in the gesture recognition context.

Motion-based feature extraction: Motion features were essentially extracted by frame to frame comparison of the gesture videos to detect the position and motion of the hand gestures. Binh et al. [95] used Kalman filter to predict hand location based on the previous frame and tracked the hand using the skin color region. Scale-Invariant Feature Transform (SIFT) features also were used for gesture recognition due to the features being invariant to scaling.

Improved Dense Trajectories (IDT) are the current state-of-the-art hand crafted feature based action recognition method on RGB videos [96]. IDT is an efficient video representation method [96] which samples dense points from each frame with several spatial scales and tracks these dense points based on the displacement information from optical flow.

Machine learning-based feature extraction: Machine learning based methods also have an important share in gesture and human activity recognition. These methods infer the mapping between the features and type of gestures and do not need human supervision to define the rules to identify the gestures.Hidden Markov Model (HMM) is one of the most popular machine learning models that classifies gestures via a stochastic process [97, 98]. Lu et al. [99] proposed identifying gestures by maximum likelihood estimation with HMM and Histogram of Gradients (HOG) descriptors over the whole body. Gorelick et al. [100] used silhouette motion volume for extracting space-time saliency and orientations as well as performed gesture classification using nearest neighbor algorithm. In addition to HMM, other machine learning approaches such as decision trees [101, 102], syntactic grammars [103] and Bayesian networks [104, 105] were used to classify hand gestures.

Quantization of features: Since these features are high dimensional, they need to be quantized or use methods such as bag-of-features representation to reduce the dimensionality [106, 107]. Although Bag of features methods have been used in action recognition, identifying a range of basic to complex range of actions [108–110], these methods suffer from a lack of correlation between the spatial and temporal domains due to using local dense features. Ikizler et a.l [111] extended the bag of features method by including the spatial orientation information in local features and [112, 113] used probabilistic latent semantic analysis to capture semantic and structural information for identifying the type of actions.

3.1.2.2 Post-Deep Learning Era

After the success of deep learning methods in image classification and object detection tasks, it became a popular approach in the human activity recognition field as well.

Temporal agnostic deep feature extraction

One of the main streams of work in human activity and gesture recognition considers videos as a collection of frames and processes them individually and models the temporal dependency between these frames in the second step. The simplest way to extend these architectures to support videos, is to run a CNN on each frame and then average the *softmax* scores for video classification. Karpathy et al. [114] used a 2D-CNN for extracting features from each frame and fused them to classify the videos.

Another method to extend CNNs for videos is to run a CNN on each frame of the video, extract the features, and then pool the features. A fully connected layer on top of the pooled features can be used for video classification. GoogLeNet introduced an inception module [115] which applies multiple convolution filters of different receptive field sizes (1×1, 3×3, 5×5) to capture information at different levels of granularity. This architecture has 12× less parameters than AlexNet and VGGNet. GoogLeNet got further extended to Inception-3 [12] and Inception-4 [116].

Temporal-aware deep feature extraction

RNN-based methods: The frame-based methods generally ignore temporal structure. Another approach to model temporal sequences is to add a RNN after the last pooling layer (in the place of a fully connected layer) of a CNN. The RNN can model temporal structure and capture long-range dependencies. RNNs are usually placed after the last convolution/pooling layer so that the CNN can act as a feature extractor while the RNN models temporal structure [117]. [118] used RNN to classify dynamic gestures from sequences of video frames.

However, RNN originally suffers from the vanishing gradient problem [119], therefore, LSTM was used in [120] to leverage this problem.

3D convolution-based methods: 3D Convolutional Networks are a natural extension of 2D-CNN that can create hierarchical representations of spatio-temporal data. Tran et al. [79] proposed a deep 3D Convolutional Network (C3D) that models appearance and motion simultaneously. The C3D takes the 16 frame long video clips as input and with 8 $3 \times 3 \times 3$ convolution filters and extracts the spatio-temporal features of the input. Molchanov et al. [121] used C3D to classify gestures in depth and RGB modalities in consecutive frames of videos. Zhang et al. [122] used the C3D and LSTM networks to capture the full temporal dependencies of dynamic gestures. C3D got further extended to Res-C3D [123] to increase the speed of inference and decrease the size of the representations and used a Support Vector Machine (SVM) to classify the gestures.

Vision-based feature extraction

Multi stream methods: The increased amount of parameters of 3D CNN based networks makes them difficult to train and usually the training would take a very long time. Inspired by a hypothesis about the human visual system [124], Simonyan and Zisserman [125] proposed a two stream convolutional network architecture for action recognition in videos. Their architecture consists of an RGB stream to capture information about scenes and objects in a video, and an optical flow stream to capture motions of the camera and objects.

Although the two stream architecture use both motion and appearance information, it does not register spatial cues with temporal cues (what is moving where). Feichtenhofer et al. [126] extended the two stream architectures by fusing the temporal and spatial streams after the last convolutional layer. The authors observed that after fusion of motion and spatial stream, the motion stream should be used again at the end to improve the performance. These two stream fusion methods were extended by using residual networks [127] and injecting residual connections from the motion stream into the spatial stream at multiple levels. This architecture was further extended to multiplicative gating for fusion instead of addition [128].

In line with inception networks, Carreira and Zisserman proposed the I3D [13] which is a deep architecture based on GoogLeNet with 3D Convolutional kernels, initially for action recognition. This network architecture by two stream optical flow and RGB input and inflated 2D kernels of inception blocks to 3D kernels.

In addition to the natural modality of videos, which is RGB, depth videos also play an important role in many of the methods tackling the gesture recognition challenging task. Depth, RGB and optical flow modalities were often used in literature [122, 123, 129, 130] to extract features via a C3D network and used a different fusion scheme to classify them via SVM [123] or *softmax* scores.

Attention-aware methods: Attention mechanism had a great impact on computer vision tasks. Larochelle and Hinton proposed a biologically inspired human vision system for object recognition [42] which used a retina that only has enough high resolution pixels to cover a small area of the image. The model must therefore learn to focus on the relevant parts of the image. Commonly in video representation methods, by default there is no distinct priority on any spatial part of the frame, therefore often the frames are resized to fit the input dimension of the network by center cropping. However, dynamic hand gestures happen in different locations of the frame and a center crop of the image, critically decreases the performance of the gesture recognition method. Therefore, in addition to using the entire spatial information in video frames, the attention in gesture recognition should be localized on a semantic object.

Soft attention feature extraction is based on weighting the average of features and is focusing on different parts of the frame [131] which resulted in improving the baseline in action recognition. VideoLSTM [132] learns the sequential features with motion-based attention, which provides better guidance towards relevant spatio-temporal locations. Due to the dependency on supplementary information, the model incurred substantial costs and the lack of human pose integration reduced the flexibility of the method in human-specific feature extraction.

Transformer networks: Jaderberg et al. introduced a spatial transformer module that can be inserted into existing CNN architectures and spatially transform feature maps without extra training supervision [133]. The spatial transformer module consists of three parts. The first part is a localization network that consists of a number of hidden layers followed by a regression layer. The second part is a grid generator that creates a sampling grid based on the predicted transformation parameters. The third part is the image sampler that takes a feature map and the sampling grid as inputs, and produces the output map sampled from the input at the grid points. It should be noted that the spatial transformer module [133] is different from the Transformer network introduced in [134] which is built on self-attention and uses key-value pairs in seq2seq models.

Recent works on the *Video Action Transformer Network* [135] uses a modified *transformer architecture* [133] to classify the action of a target person. Their model uses the I3D network [13] for extracting the base features and a region proposal network [67] for a sampling mechanism to localize people performing actions. Their attention mechanism learns to extract features with emphasis on meaningful body parts for action recognition, such as hands and the face.

Pose-based feature extraction

In addition to different methods to increase the accuracy and performance of gesture recognition, using different modalities can benefit the recognition results. Unlike depth modality, which requires special devices, skeletal information can be extracted from RGB videos and different tasks such as human activity and gesture recognition can benefit from this modality to improve the results.

Multi-stream methods: In contrast to vision based models, they are less computationally intensive and more robust against a complex background, viewpoint variation, motion speed, and changing body scales. Cao et al. [136] presented an attention model, which predicts spatio-temporal key-points in 3D convolutional feature maps. Most of the methods in this area use the multi-modal input, namely visual and skeletal and possibly other modalities to extract discriminative gesture representations. Neverova et al. [137] proposed a multi-modal method with RGB, depth, audio stream and body skeleton data to capture several spatial information. Their method identifies the label for the activity based on the final label of a sequence of frames labels, computed by majority vote.

Two-stream RNNs were also proposed to model spatial and temporal information using skeletal data for activity recognition [138]. LSTM has shown better performance regarding modeling temporal dependencies of the activity frames. Du et al. [139] used bidirectional LSTM to model the temporal dimension of the human action by dividing human body pose in five meaningful parts. The main drawback on these kinds of models is that the overall accuracy depends on the precision of the pose estimation. Additionally to capture the motion direction of skeletal joints, [140] used joint trajectory maps, which are projected on three planes and then used for classification using CNN.

Attention-aware methods: Attention aware methods also became popular in pose based activity recognition over the past few years. Liu et al. [141] proposed using a context aware attention LSTM network to process and update the weight of the important body joints in an action. Similarly [142] used LSTM to model the temporal dependencies and attention mechanism to focus on outputs of different frames. Liu et al [141] proposed a view invariant skeletal projection in 2D images to extract spatio-temporal features of skeletal joints.

SkeletonNet [143] extracts pairwise relative positions between skeletal joints and using the cosine similarity measure, 10 representations were concatenated and used as input to a two stream CNN. Skeleton-Guided Multimodal Network (SGM-Net) is another multi-modal network using skeletal information to emphasize on corresponding RGB components and enhance their importance [144]. RPAN [14] resize their joint coordinates to a 2D map to feed it into a pre-trained CNN. The long-term dependencies and semantic information of the body structure are captured by the large receptive fields of deep neural networks and the correlation of the joints are considered by dividing the human skeleton into semantically correlated parts for modeling the dependencies. The architecture used to extract the spatial features is based on the TSN [83] using optical flow and visual cues as inputs.

Actor-Transformer [145] similarly to [133], belongs to the family of transformer-based methods based method which models the actor specific representations based on pose information and features extracted from I3D network. The two stream of optical flow and RGB extract the dynamic representation from multiple frames, and the pose information extracted from a keyframe contributes as the static representation.

3.2 Vision-based Gesture Retrieval

Gesture retrieval refers to the search of similar instances of dynamic hand gestures in videos based on features extracted by different methods. The search process includes a query and a set of results which are retrieved from the collection based on their similarity score with the query.

Gesture retrieval is a very useful method in search systems to find hand gestures and also annotating unlabeled gestures in large non-annotated datasets. However, this field is rather under explored. In a broader sense, gesture retrieval falls into the Content-Based Video Retrieval (CBVR) which is a popular field of computer vision.

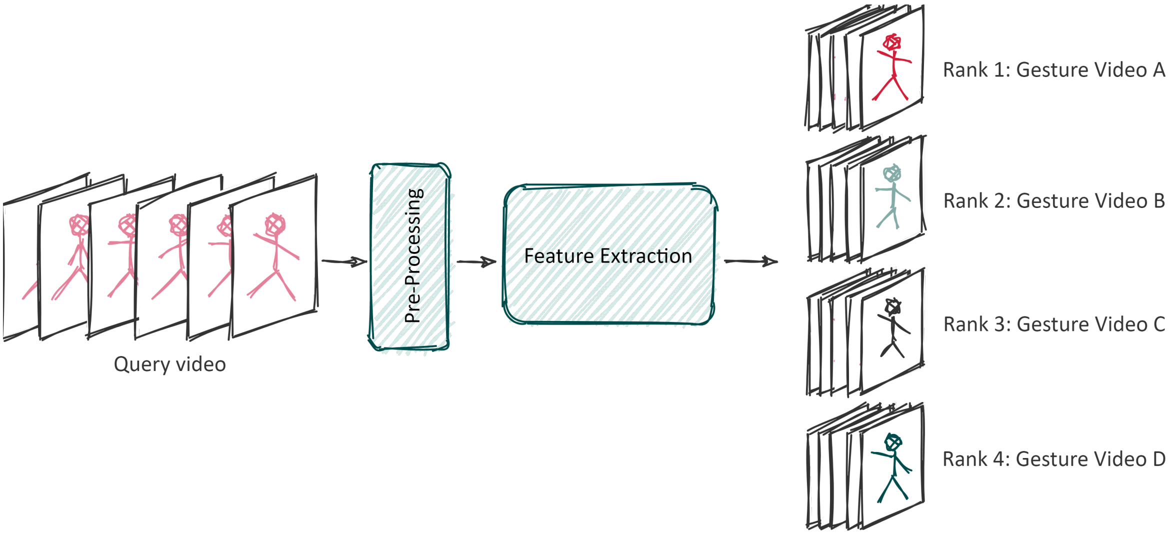


Figure 3.6. A general structure of a gesture retrieval task where the output is a ranked list of similar results to the query.

3.2.1 Content based Video Retrieval

With the tremendous increase in the digital contents due to the advances in recording devices technology, finding the desired video clip among the huge amount of stored data is a tedious task. CBVR is associated with searching for specific content in a collection of videos. This search needs a query, which can have different approaches, such as Query by Example (QbE), sketch, image, text and audio [146]. The QbE takes a video clip and performs the search to find similar instances to this example. Sketch and image queries are used to search in the key frames of the video for similar content as the query [52]. Text queries also can be used to search within the metadata of the videos (if available), the spoken language [147] or the written text in the frames [148].

A great deal of video retrieval methods use the deep learning methods explained in Section 3.1.2.2 to extract features to augment or replace the hand-crafted features. The sketch based retrieval has been developed over the years from edge and color based feature extraction of sketches [149] to semantic-based pixel-wise labeling of frames [150]. The activity recognition in CBVR systems usually results in providing textual labels to perform text-based queries.

One of the main steps in CBVR is the shot segmentation which essentially divides a long video into smaller clips containing a related sequence of frames [151]. Segmenting the videos into shots is an extensive research topic and different methods based on ongoing activity or camera operation [152].

The result of the search is usually displayed as candidates which match the query. The result videos are usually ranked by a relevance criterion or users feedback. To deal with the curse of dimensionality and efficient search in high dimensional feature vectors, methods such as indexing and dimensionality reduction are used.

Dimensionality Reduction

Dimensionality reduction is one of the methods to increase the efficiency of the retrieval process by increasing the speed for the search within feature collections. There are different ways to reduce the dimensionality such as using Principal Components Analysis (PCA) to map the original dimension to a new one [153] or feature selection to remove irrelevant and redundant features [154].

Indexing is another alternative which aims to collect, parse, and store data to facilitate the information retrieval procedure. There are numerous index structures for the problem of similarity search and information retrieval. Two of the famous indexing structures are forward and inverted index: a forward index is the list of documents and the keywords associated with them, while inverted index is the list of keywords and the documents in which these keywords appear.

Fingerprints or signatures are another type of indexing which are quite robust to errors. Signatures are compact representations of the document which are used to ease the retrieval process of data. All these three are the terminology which are emerged from database background and some are also used in machine learning literature [155].

Hashing

Normally, the index keys are stored as binary codes which are called *hash* codes and have different lengths based on the need in retrieval systems. The shorter the code, the less information it can store but the faster the retrieval by the search engine can be. The hash codes are stored in a data structure called *hash table* to map the code to its symbol. The ideal hash code is generally compact and is easily computable. Hashing is the terminology which is used both in the context of database systems and machine learning and has common definition in both fields.

Essentially hash-based video retrieval consists of two stages of extracting the features and embedding the hash codes. In traditional methods, namely *non-learning based*, the loss of information in the process is very likely and is not reversible [156]. On the other hand *learning-based* methods learn the representations by preserving the similarity between the data points, in a short hash code.

Approximate Nearest Neighbor search: In retrieval settings, finding the exact Nearest Neighbors is very time consuming. That is the reason the Approximate Nearest Neighbor searches became popular.

Tree based methods such as Hierarchical methods [157] and K-means clustering [158] proved to lose its performance when the dimension of the database increases. Conversely Vector Approximation File (VA-File) [159] method finds approximations in two stage processes by firstly selecting candidates for k-Nearest Neighbor search, and in the next step calculates the exact neighbors.

On the other hand, hashing based methods such as Locality Sensitive Hashing (LSH) [160] aim to solve the Approximate or exact Near Neighbor Search in high dimensional spaces which ranks the search results based on their relevance to the query. Over the past decades, different variants of theLSH algorithm developed to deal with the limitations of the original LSH. A family of LSH algorithms were developed with different similarity measures [161] such as Hamming LSH [160] which uses Hamming distance to approximate the neighborhood. Another variant of LSH families deal with the theoretical limitations of it such as the characteristics of the similarity measure [162] or focus on improving search efficiency to optimize indexing and search [163].

Frame-based deep hashing: Although many different end-to-end representation learning methods for CBVR methods exist, majority of the hash-based video retrieval is built upon image hashing methods [164,165]. Majority of the hash-based CBVR methods use supervised learning to extract binary representations for images and generalize them to each frame in the video [166]. Such methods consider videos as the sequence of images, and extract the short representations based on the similarity between the frames of the video. Disregarding the temporal relationship between the frames and the generated hash codes it does not preserve the temporal similarity.

Multiple Feature Hashing (MFH) [167] extended image hashing to video hashing without considering temporal structure of the video. Another approach to encode the temporal information in short hash codes is using pooling [168]. Similarity-Preserving Deep Temporal Hashing (SPDTH) proposed using stacked Gated Recurrent Unit (GRU) to model the temporal dimension of CNN features of frames and learning the hash code representations without the mapping layer [169]. Zhang et al. proposed Self-Supervised Temporal Hashing (SSTH) which uses a binary auto-encoder to learn the hash-based representations and Song et al. [170] extended their method to include the neighborhood similarity in the hash codes. Unsupervised Deep Video Hashing (UDVH) [171] was proposed to enhance the previous hashing methods by balancing the variations of dimensionality by using a LSTM network. It additionally got extended to replace LSTM with TSN to enhance the feature modeling and retrieval performance [172].

Attention-based hashing: Recently, neighborhood attention mechanisms have been employed in hashing methods to incorporate the spatio-temporal information of the neighboring segments, when representing the video clips. Neighborhood Preserving Hashing (NPH) used the RNN-based reconstruction module to enhance the similarity perseverance of the hash codes. Attention-based Video Hashing (AVH) [173] also uses the attention mechanism together with CNN and LSTM network to learn short binary codes representing the structural information of the video frames.

3.2.2 Gesture Similarity retrieval

Compared to the different retrieval tasks, relatively little work has been done in comparing the similarity of hand gestures for the purpose of retrieval, i.e., the search of video sequences on the basis of the gestures that appear in these sequences.

Sign retrieval: Sign language look up systems [174] refers to this task partly by finding the meaning of a sign by querying the video database. One of the methods used in the literature is Dynamic Time Warping (DTW) which aligns sequences of frames in temporal dimension to compute the matching score. These two sequences are the query and the reference, and the results indicates whether the query is similar to the reference or not [175]. Stefan et al. [174] used this method and its extension, Dynamic Space-Time Warping (DSTW) [176] to compare the automatic and non-automatic sign search in relatively small dataset of 933 signs. Although their proposed method fits the scale of the sign search in the database, it lacks the learning ability therefore, does not generalize for larger scale problems.

Control gestures retrieval: In the context of human-machine interaction, Yousefi et al. [177] proposed a gesture search system for 3D hand gestures to control a gesture based interaction interface. The global orientation of hand gestures extracted from a motion capture device is used in the dataset to be compared with the query. The query processing consists of extracting the edge and associated angle interval of the hand and comparing it with the stored reference. The system is based on the database recorded from the motion capture device to track the 3D hand gestures which is not widely available. Additionally, the collection of gestures the target problem inherently does not involve many different classes and diversity in types of gestures, therefore, computing the angle intervals can be representative enough for the similarity search.

Pose retrieval: Pose search [178], is another closely related method which retrieves similar human body poses to the query in large video collections based on skeletal data extracted from pose estimation. Their method contains a HOG-based descriptor for retrieving similar single frame poses and has promising results on Hollywood Movie dataset [179]. However, their method is not capable of retrieving poses in motion, and the trajectory is limited to the body pose at a snapshot. Recently [180], proposed a system which takes a dynamic action as a query and retrieves similar pose sequences. The similarity measure is the numerical pose sequence distance between two same length clips. However, their method cannot handle occlusion and multi-person scenarios and the evaluation results do not suggest any scalability in large real-world dataset.

Part III

Methodology

Chapter 4

Datasets

One of the main parts of the deep learning-based methods is *data*. Most of the existing methods in this area are dependent on a large amount of data, which are manually or automatically annotated to serve as the source for training the deep learning networks. Usually, this annotated data is taken together into *datasets*. In the following chapters when discussing the methodologies, some datasets will be mentioned, on which the networks are trained on, or evaluated. Therefore, in this chapter we present an overview of these datasets for the future references.

4.1 Chalearn Isolated Gesture Dataset

The Chalearn Isolated gesture dataset [3] is one of the largest annotated hand gesture datasets available. There are approximately 36 000 videos for training which are divided into 249 classes of hand gestures.The videos of this collection contain only one gesture and are performed by 21 different individuals (see Figure 4.1).

The characteristics of the videos in this dataset poses several challenges for the gesture recognition task:

- The rather large group of people presented in the dataset requires the methods be person-agnostic and do not learn the gestures in relation with people,
- In most of the videos, there is heavy background clutter, which leads the methods to have lower accuracy in recognizing the gestures,
- The videos have low resolution with a large amount of noise, which also makes the results obtained from methods developed and tested on this dataset prone to error.

We will use this dataset for training our methods due to: firstly, the large number of samples per class, secondly, the large number of classes available. As the goal of this thesis is to be able to perform in the real-world data collection and application, it is important that our training data has a large number of classes. The gestures articulated by the subjects are not specifically communication gestures and range between Indian Mudra, to Chinese numbers and Italian hand gestures.

All videos in the dataset come in RGB and depth video format, with a total of 47 933 videos which are split into 35 878 videos (∼42 hours) for the training set, 5 784 videos (∼7 hours) for the validation set, and 6 271 videos (∼8.5 hours) for the test set. A summary of these numbers are listed in Table 4.1

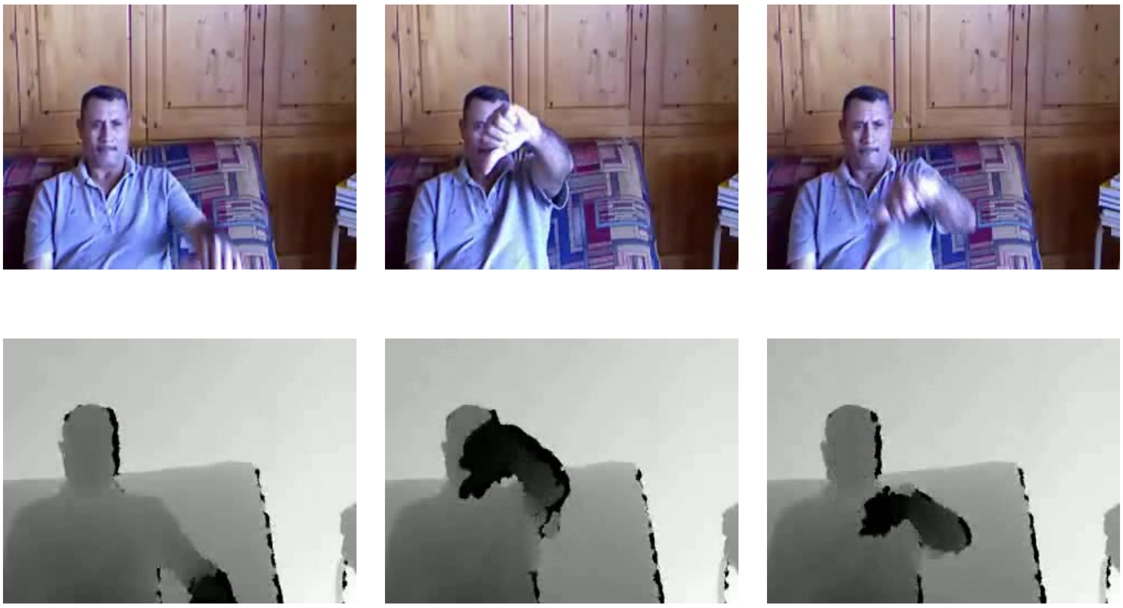


Figure 4.1. Sample frames from one video in the Chalearn Isolated Gesture dataset [3] in RGB and depth modality.

Table 4.1. General information about the Chalearn Isolated gesture dataset as the number of labels, videos and subjects in training, validation and test sets.

	Classes	**Videos**	**Subjects**
Training	249	35 878	17
Validation	249	5 784	2
Test	249	6 271	2
All	249	47 933	21

4.2 Chalearn Continuous Gesture Dataset

The ChaLearn Continuous gesture dataset [3] is another variation of the ChaLearn datasets which exhibit more than one gesture per video. The collection has RGB and depth video format with nearly 48 000 gesture instances in 22 535 videos. The ChaLearn Continuous gesture dataset shares the 249 gesture labels with the Chalearn Isolated gesture dataset.

In addition to the challenges mentioned for the Isolated variation of this dataset, the presence of multiple gestures per video requires a temporal detection method for the correct recognition of the hand gestures. Table 4.2 summarizes the information about this dataset.

Table 4.2. General information about the Chalearn Continuous gesture dataset as the number of labels, videos and subjects in training, validation and test sets.

	Classes	Videos	Gestures	Subjects
Training	249	30 442	14 134	17
Validation	249	8 889	4 179	2
Test	249	8 602	4 042	2
All	249	47 933	22 535	21

The Chalearn datasets are the largest annotated gesture dataset available and are widely used in gesture recognition tasks for training and evaluation.

4.3 Jester

Another widely used gesture dataset for task of recognition is Jester [4] which is a uni-modal and annotated collection of gesture videos in RGB format. The videos of this dataset are recorded by 1376 actors via the webcam in different lengths, when people perform one of the 27 pre-defined gestures, including “no gesture”. The gestures in this dataset are a mix of static (thumbs up or stop sign) and dynamic gestures (swiping right or left) which are commonly used in human machine interfaces. An example of these gestures can be seen in Figure 4.2.

The total number of videos is 148 092, divided into training, validation and test set. However, only the annotations for training and validation sets are publicly available. For testing the methods on the test set of this dataset, one must send the results to the organizers to calculate the accuracy.

One of the features of this dataset is the presence of class with “no gesture” label, where in the videos, the actor does not perform any gesture. This category is commonly used in deep neural network methods to train the temporal gesture detectors.

4.4 JHMDB

Joint Annotated Human Motion Database (JHMDB) is a subset of HMDB [5] which originally contains 7 000 videos divided into 51 action categories. JHMDB with 21 action classes, which are selected to represent single person actions such as brush hair, climb stairs and golf, is closer to our task. There are minimum of 100 clips per action available in HMDB dataset which are collected from different sources, such as movies while this number is 45 on average for JHMDB.

The actions in HMDB related to facial expressions such as smiling, laughing and talking. Therefore, we will use the JHMDB to benchmark parts of our experiments. There are different annotations such as visible body parts, camera motion and angle available for HMDB which

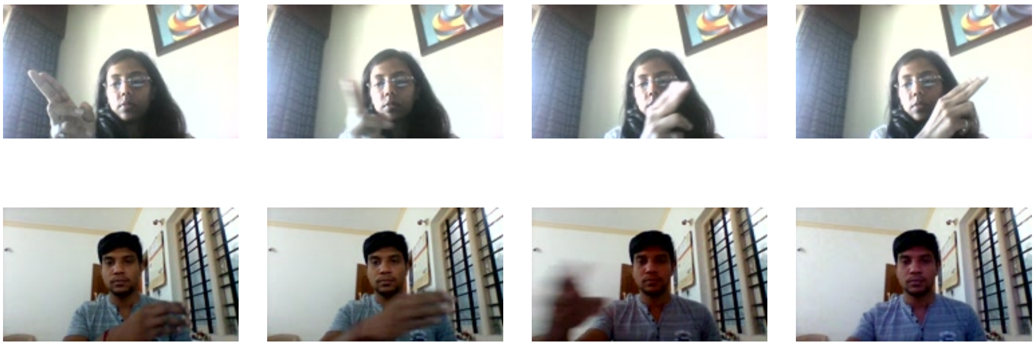


Figure 4.2. Example of frames from two videos in the Jester dataset with two types of swiping gestures. [4].

we use instead of the pose annotations of the JHMDB. An example of this dataset is shown in Figure 4.3.

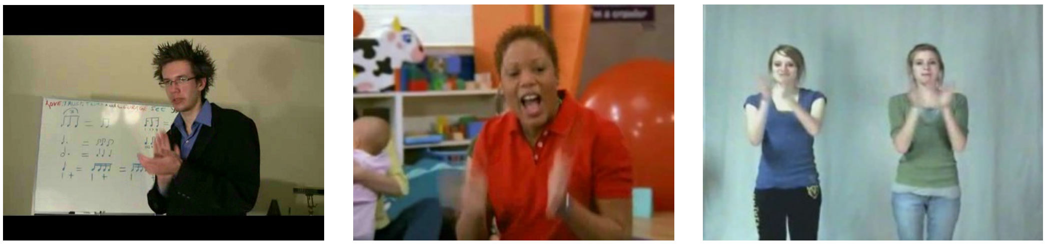


Figure 4.3. Examplee keyframes from three videos in the HMDB dataset with clapping action. [5].

4.5 Newsscape: UCLA Recorded News Library

The UCLA Library NewsScape [6] contains digitized television news programs collected from international broadcasting channels and online sources from 2004 to the present. The collection includes more than 400 000 recordings and hundreds of thousands of hours of videos, which are available with closed captions and meta-data.

Despite the enormity of the dataset, and all the benefits it can have for the computer vision community, one of the main challenges of using this dataset are the annotations. Due to the very large number of shows and videos, the labor-intensive task of manual annotation takes time to provide enough data to train deep learning models. However, the real-world setting of the videos and uncontrolled human activities, can serve as an invaluable testing medium to qualitatively observe the performance of the developed methods. An example snippet of this dataset is shown in Figure 4.4.

NewsScape introduces various challenges for computer vision methods, aiming to recognize or search for actions or gestures within this dataset:

- There is a large amount of occlusions in the scenes with different sources such as person-person, person-object and banners and subtitles.
- In most of the shows, there are multiple people present in the scene, where in some cases a large crowd with more than 50 people are shown.
- Since the actors in the shows are not "asked" to perform specific actions or gestures, the actions and gestures are not controlled, meaning they come with various speed and orientation. Additionally in most cases two consecutive gestures follow each other with little or no pause.
- The videos in this dataset are recorded from multiple cameras which introduces large variation of viewpoint and perspective on the human of the interest. The camera has a large amount of motion along different axes such as zooming.

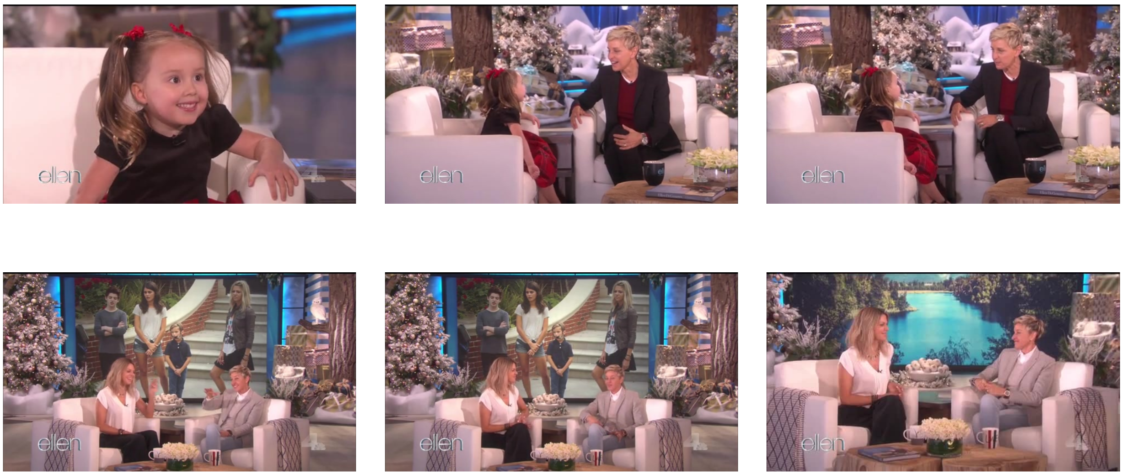


Figure 4.4. example frames from seperate segments of a video in NewsScape dataset from Ellen DeGeneres show. [6].

For our evaluations, we have selected a subset of this collection, from the *Ellen DeGeneres show* throughout the year 2017. We have used approximately 250 hours of videos with non-unique shows. Due to the nature of the talk show, this collection exhibits numerous co-speech and communication gestures which is of interest for the linguists and cognitive scientists.

Chapter 5

Gesture Spatio-Temporal Preprocessing

As a primary goal, this thesis aims to develop a method which can encode gesture information in videos into an embedding and further use this in gesture retrieval settings. Along the way, we learn that to reach this goal, we can also develop hand gesture recognition, where the encoded representations would classify the gestures which are visible in the videos. However, the gesture recognition and retrieval comes with some domain specific challenges which require the special pipeline to overcome them. In this thesis we propose an approach consisting of two main components, namely preprocessing and feature extraction. An overview of this pipeline is shown in Figure 5.1.

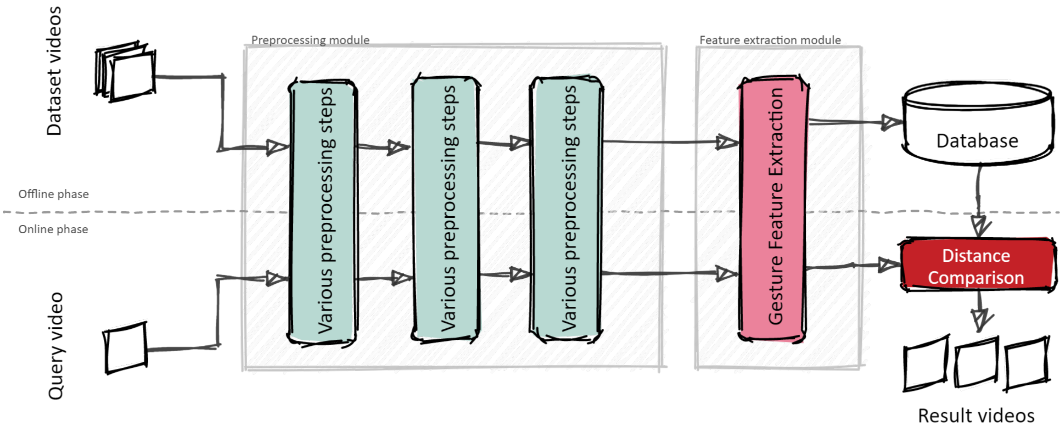

Dataset videos
Preprocessing module
Various preprocessing steps
Various preprocessing steps
Various preprocessing steps
Feature extraction module
Gesture Feature Extraction
Database
Offline phase
Online phase
Distance Comparison
Query video
Result videos

Figure 5.1. Illustration of different components of the pipeline for gesture video retrieval.

Preprocessing is an inseparable step of machine and deep learning models which is essentially referring to preparing the data to feed the neural network, either for training or inference. The type of preprocessing heavily depends on the task and data. For example, in image classification tasks, the model usually requires an input image with a certain dimension, therefore, all the images are resized before feeding to the network. However, in some tasks the preprocessing step is changing a color image to gray-scale, extracting some additional data, or even adding some, to help the model in processing the input. Some of these procedures are independent computer vision tasks which standalone can solve different problems, however, when used with other tasks, could be considered as a preprocessing step.

Many categories of human activities are strongly dependent on the objects and surrounding environment, which can help the identification of an activity. For example *playing ice hockey*, see Figure 5.2 action, heavily depends on one or several sticks and the ice field being visible in a frame. Therefore, the existence of the white field could help the recognition model to identify the action. Unlike activities such as ice hockey, hand gestures can happen in conversation in any situation and environment. In other words, the background of a person rarely has meaningful correlation with the hand gesture and removing it can help the recognition model to focus only on the hand motion. Thus, removing the background from video frames is a crucial preprocessing step in gesture-related tasks.

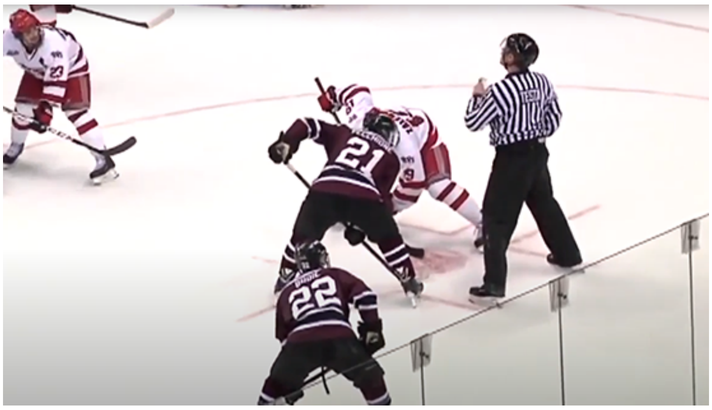


Figure 5.2. A snapshot of the action “playing ice hockey” from the Activitynet [7] dataset. The action heavily depends on the white field of ice and the sticks of the players which helps the recognition model to identify the action “playing ice hockey”.

Additionally, to ease the video processing, usually the videos are segmented based on where the camera view changes which is referred to as *shot segmentation*. However, this type of segmentation could cause the gestures to be divided into two different shots causing the incomplete gesture not to be identified properly. Therefore, a special form of video segmentation as a preprocessing step is necessary for long video collections.

Another computer vision task used in our gesture recognition model as a preprocessing step, is temporal localization. This component is responsible to detect the gestures temporally and determines where a gesture starts and ends. This module is specifically important in long videos where multiple gestures exist and lack of temporal segmentation based on presence of the gestures, would result in loss of information.

In this chapter we explain in detail the spatio-temporal preprocessing module which has a critical role in the gesture recognition and retrieval results. In Section 5.1 we describe the cross-angle spatio-temporal human segmentation in recorded scenes with multiple persons present. This module is necessary to identify a specific gesture when different people are gesticulating. In Section 5.2 two methods are proposed to estimate the start and end of the gestures and prepare them for feature extraction and learning.

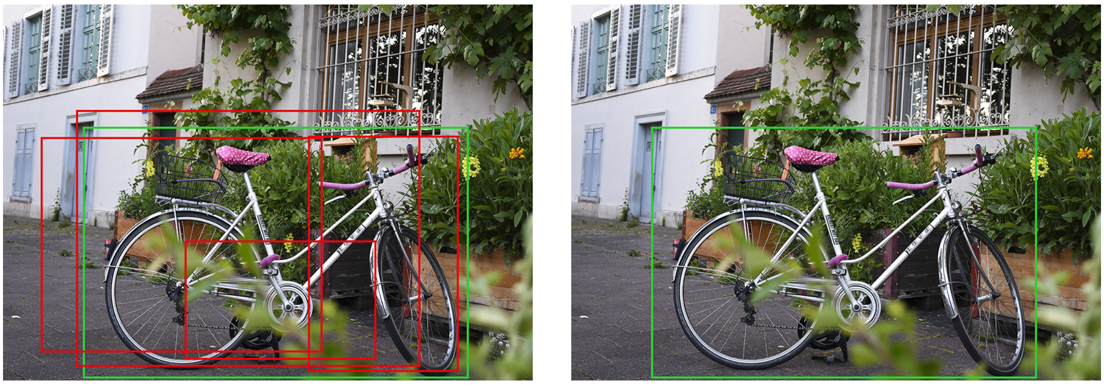
5.1 Cross-angle Spatio-Temporal Human Segmentation in Multi-Person Scenes

The spatial human segmentation falls into the category of semantic segmentation which we use as a preprocessing component to remove the cluttered background. On one hand, the combinations of the colors and objects in the scenes where the gesture articulations happen, can affect the feature extraction and by including this information, the feature vector does not exclusively represent hand gestures. On the other hand, when multiple persons are present in one or multiple frames, the extracted features will not be informative about one individual hand gesture made by one person.

Additionally, the reappearance of each person in the adjacent frames, needs to be recorded, to make sure the gesture feature of the correct person is extracted. For this purpose we use the re-identification method, which allows tracking each person through the entire video as well as the camera shots. Together with spatial segmentation, each person's sequence of gestures is fed to the feature extraction component.

5.1.1 Pose-based Human Instance Segmentation

The main intuition of using a pose-based human instance segmentation is its difference with the general object segmentation method. The majority of instance segmentation methods relies on the selection of region proposals generated using NMS as shown in Fig.5.3.



(a) before NMS

(b) After NMS

Figure 5.3. An example of the output of an object detection method before and after NMS.

One major disadvantage of such methods is the case where there is a large overlap between the same class objects and the NMS will eliminate one of these instances as redundant instances. Therefore, in scenes with a large overlap between the same class objects, for example in a talk show with multiple persons sitting beside each other, such methods fail in detecting the human instances properly.

However, the human category can be defined by a special characteristic such as pose skeleton, which can help the detection and segmentation algorithms. The pose skeleton information can provide unique information about each individual human instance and can help with the highly overlapping instances.

The idea of using the body joint keypoints to create a skeletal model and segmentation mask of a person –which is referred to as bottom-up approach– is an effective method in contrast to top-down methods where the human skeletons are detected based on the bounding boxes identified in a scene. For this reason, we use one of the human instance segmentation methods which has the bottom-up structure and with the unique segmentation module, identifies and masks human instances even in complex environments (see Figure.5.4).

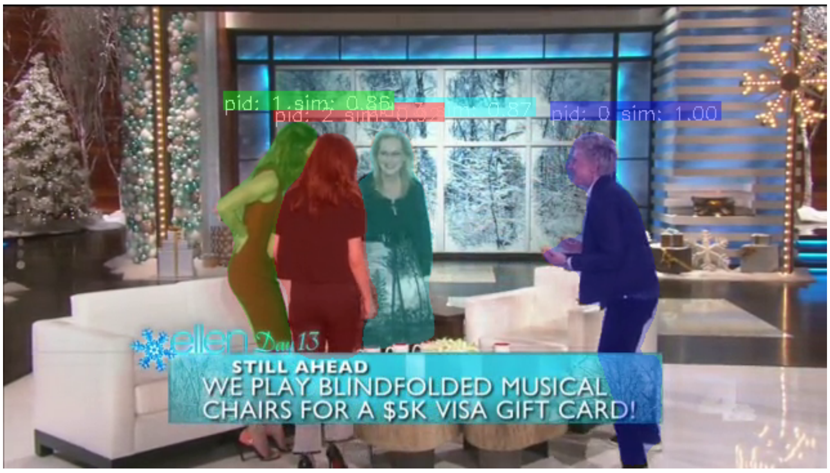


Figure 5.4. The output of the spatial segmentation model with a heavily occluded input.

The method we use is Pose2Seg [8] and its network consists of three main parts: Affine-Align, Skeleton Features and SegModule. In the following we explain this segmentation module in detail.

The overall network structure is shown in Figure 5.5. The network essentially has two types of input: RGB frames and a human pose skeleton.

The pose skeletal data can be obtained with any pose estimation method. In this thesis we use openpose [10] which is based on the encoder-decoder model and generates heatmaps which are representing the likelihood of a keypoint. The exact coordinates are obtained based on the highest likelihood of presence of a keypoint.

The inputs to the network consists of a sequence of N frames with the dimension of $m \times n$ and sequence of human poses in each frame. The pose of an individual person is a list of vectors with the form:

$$P = (kp_1, kp_2, \dots, kp_l) \in \mathbb{R}^{l \times 3} \tag{5.1}$$

where l is a dataset related parameter representing the number of parts in a pose (17 in COCO [181]), $kp_i = (x, y, v) \in \mathbb{R}^3$ is the vector containing the coordinates of keypoints (x, y) and the visibility of the keypoint is defined as:

$$v = \begin{cases} 0 & \text{if the keypoint is not in the frame} \\ 1 & \text{if the keypoint is in the frame but not visible} \\ 2 & \text{if the keypoint is clearly visible} \end{cases} \tag{5.2}$$

The RGB stream of data is used to extract features using FPN [9]. The features generated by this network have rich semantics in all levels and use a single input image scale. Essentially

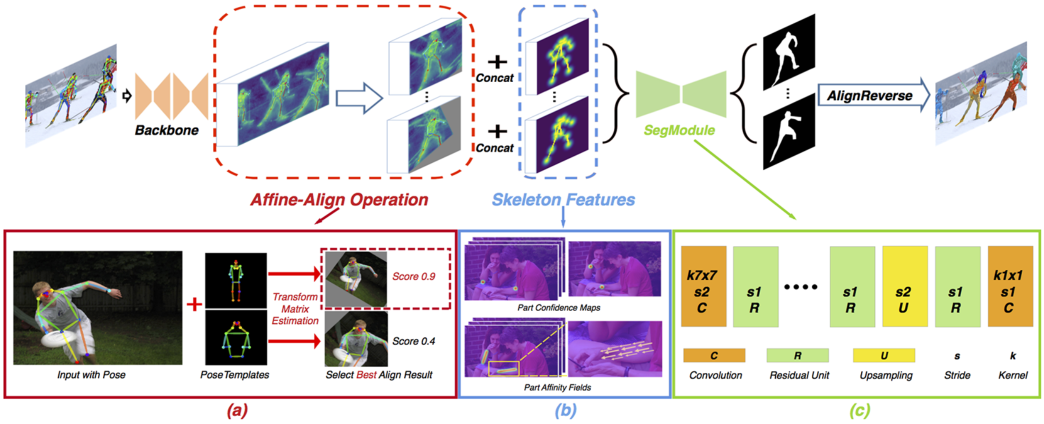


Figure 5.5. Architecture diagram of Pose2Seg [8] pipeline for human instance segmentation. The affine-align operation translates the pose templates ad with skeleton features the human segments are formed.

the architecture is designed to combine low-resolution, semantically strong features with high resolution, semantically weak features.

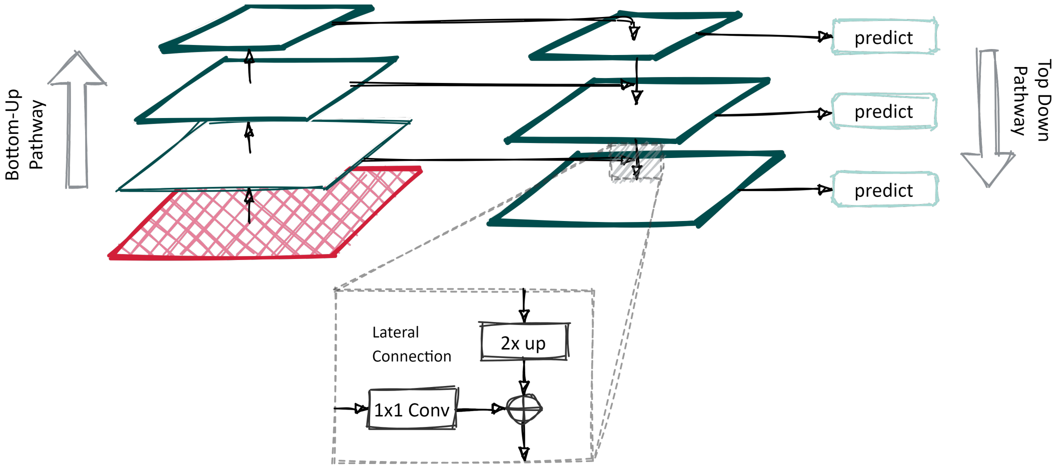


Figure 5.6. The mechanism of the FPN network. The independant prediction per level, leverages the semantic gaps caused by the different layers predictions. The figure is redrawn from [9].

Inspired by RoI-pooling in Faster-R-CNN and RoI-Align in Mask-R-CNN, the authors proposed *Affine-Align* operation which instead of aligning the human bodies to the bounding boxes, aligns them to template human poses. These template poses are a collection of most frequent poses in the COCO dataset, which are also in line with real-world observations. (Figure 5.7)

The pose template is used as a reference to assign a score to each input pose after comparison. This comparison is made by aligning the template to the input pose to find the transformation matrix H and choosing the one with the highest score:

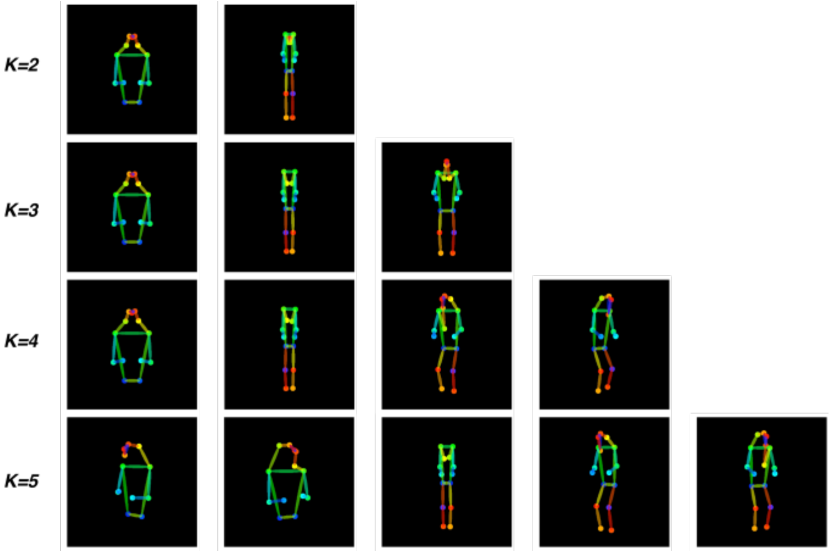


Figure 5.7. The template poses obtained from clustering the most common human poses in COCO dataset using K-means. Figure from [9].

$$
\begin{aligned}
H^* &= \arg\min_{H} \parallel H \cdot P - P_\zeta \parallel \\
\text{score } &= \exp(- \parallel H^* \cdot P - P_\zeta \parallel)
\end{aligned}
\tag{5.3}
$$

where P_ζ is the pose template, P is the input pose and H^* is the affine transform matrix for the best chosen template. The H^* associated with the best score for each input pose is applied on the image features (Figure. 5.8)

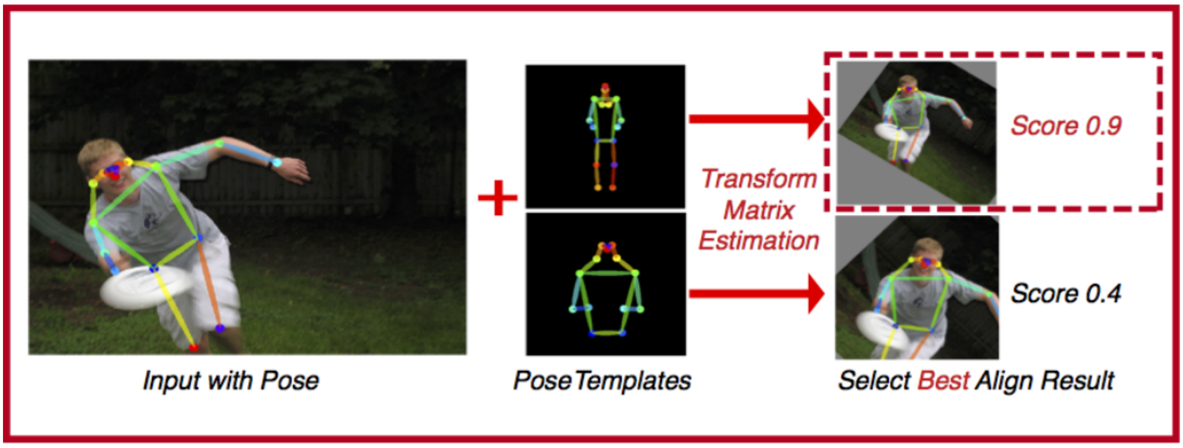


Figure 5.8. Affine-align transformation of the input image based on the reference pose. Figure from [9].

In addition to the affine-align operation, the input pose is also used to extract *skeleton features*. These features are essentially the PAFs which are 2-channel vectors encoding the location and orientation of each skeleton in the human pose. Additionally, the part confidence map is used to highlight the importance of keypoint regions.

The final stage in the pose2seg method is to map the segmentation to the actual frame. This is done with a CNN architecture with a 7×7 convolutional layer followed by several standard residual units on the concatenated skeleton and image features. A bilinear up-sampling unit is used to restore the resolution and the final mask is predicted using a 1×1 convolutional layer.

5.1.2 Person Tracking in Multi-Person Scenarios

In single person scenarios, by removing the background clutter from the scene, the segmented human body can be forwarded to the feature extraction component for gesture recognition. However, in multi person scenarios, a scene with multiple human instances, occasionally performing gestures simultaneously cannot be processed for gesture recognition as is. Additionally, processing long videos requires a form of temporal segmentation into parts which detects abrupt or gradual transition effects in videos. However, in scenarios where people are talking and co-speech gestures happen, these abrupt transitions often lead to cutting the gesture before it ends. One example of such a scene can be viewed in Figure 5.10 where Ellen in the first image is in a close-up shot, and the next frame shows her from another camera with a different angle with guests. A normal shot segmentation method would cut the scene between these two frames based on the abrupt transition, even though the gesture is continued over this transition.

To overcome the multi-person processing challenge and guarantee a gesture friendly shot segmentation, we propose a framework to segment the videos based on the presence of each individual in a sequence of frames. The method is based on re-identification of each individual in each scene and segmenting the video into short clips containing that person.

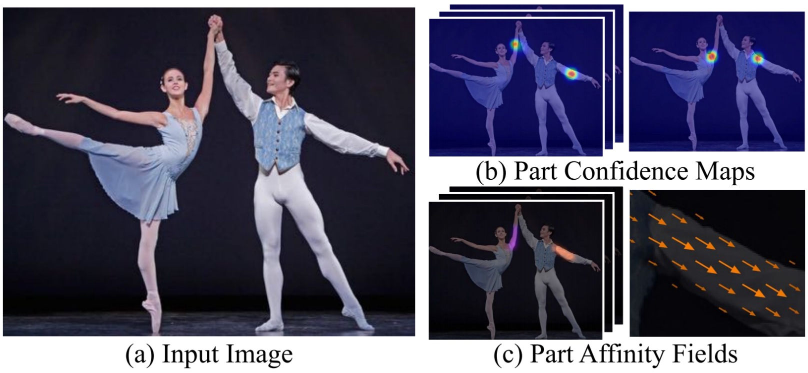


Figure 5.9. General overview on PAF which essentially indicates the connection between the joints. Figure from [10].

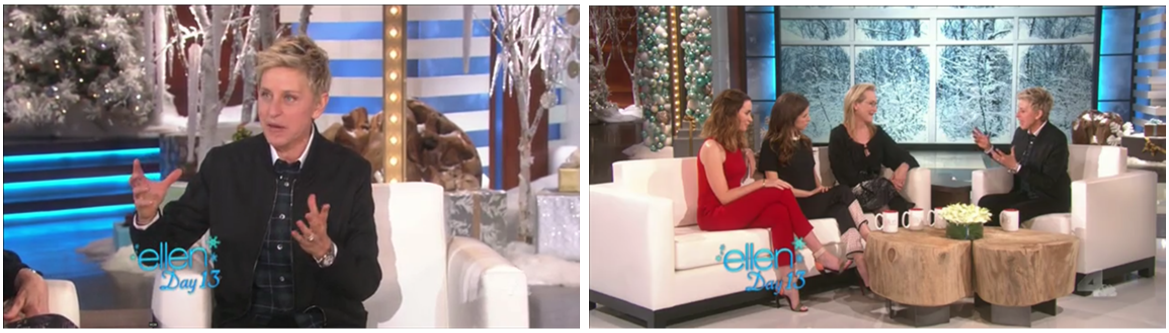


Figure 5.10. An example to show the importance of the person tracking and cross-angle segmentation. Only one of the four people present is performing the gesture and the abrupt camera transition is cutting the Ellen's hand gesture

Our proposed solution is inspired by the person re-identification method in [11]. This task refers to the problem of searching the collection of individuals, matching and identifying them according to a reference image.

The method originally has a detection and re-identification part. In the following we explain the mechanism of the method, and afterwards the variations made to adapt it to our problem. An overview of the method is shown in Figure 5.11.

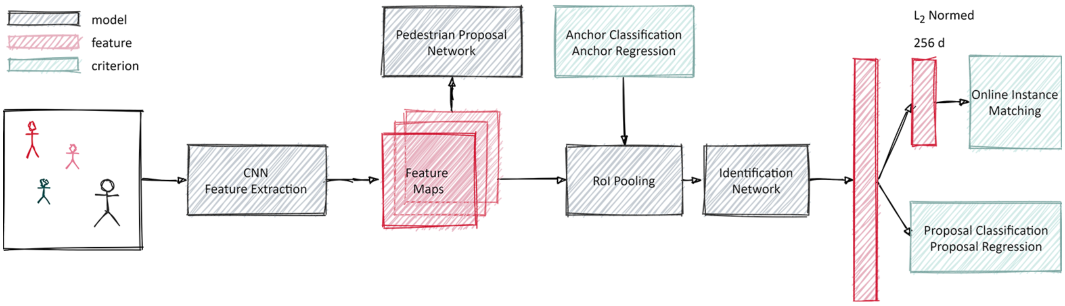


Figure 5.11. Architecture diagram of the re-identification component. The figure is redrawn from [11]

The input to the network is an RGB image which is fed to a CNN for feature extraction where only the first four layers are used. The extracted feature maps are used to detect *pedestrian* instances by passing through the RPN from Faster R-CNN, yielding 128 pedestrian proposals. To identify the person of interest, a RoI Pooling layer is applied on each proposal and they are passed through the remaining layers of the CNN and the final feature vector is obtained. The re-identification process will include projecting the 2048 dimensional feature to ℓ_2 normalized 256 dimensional space and similarity learning process using Online Instance Matching (OIM). The training process also includes suppression of non-person proposals and reducing their spatial misalignment.

In short, the goal of OIM is to reduce the distance between the features of the reference with the correct target person, while maximizing the distance between the reference and other people. OIM is essentially minimizing the number of samples that need to be compared to each other by maintaining two look up tables. One of these tables is used for the labeled identities $\mathbf{V} \in \mathbb{R}^{d \times L}$ where d is the feature dimension and L is the number of classes and initially consists of the positive sample identities. The other table is for the list of identity features which are not in the list of references $\mathbf{U} \in \mathbb{R}^{d \times Q}$ where Q is the length of the list. These samples are used as negative samples for the training. In the forward path, the cosine similarity between the mini-batch samples $\mathbf{x} \in \mathbb{R}^d$ and the items in $\mathbf{V}$ is computed. In case of the matching class ID with i, the i-th column of the lookup table $\mathbf{V}$ is replaced with $\mathbf{v}_i \leftarrow \gamma \mathbf{v}_i + (1 - \gamma)\mathbf{x}$ where $\gamma \in [0, 1]$ is the momentum.

For the negative samples in lookup table $\mathbf{U}$, a circular queue is defined, where after computing their cosine similarity with the samples feature $\mathbf{x}$ from the mini-batch, a new feature vector is added to the queue and the previous one is removed.

The probability of a sample feature $\mathbf{x}$ being identified with class ID i is calculated by a softmax function as:

$$\mathbf{p}_i = \frac{\exp\left(\frac{\mathbf{v}_i^T \mathbf{x}}{\tau}\right)}{\sum_{j=1}^{L} \exp\left(\frac{\mathbf{v}_i^T \mathbf{x}}{\tau}\right) + \sum_{k=1}^{Q} \exp\left(\frac{\mathbf{u}_k^T \mathbf{x}}{\tau}\right)} \tag{5.4}$$

where τ is the temperature parameter determining the softness of the distribution. The probability of a sample feature $\mathbf{x}$ being identified as an identity in the circular queue is calculated as:

$$\mathbf{q}_i = \frac{\exp\left(\frac{\mathbf{u}_i^T \mathbf{x}}{\tau}\right)}{\sum_{j=1}^{L} \exp\left(\frac{\mathbf{v}_j^T \mathbf{x}}{\tau}\right) + \sum_{k=1}^{Q} \exp\left(\frac{\mathbf{u}_k^T \mathbf{x}}{\tau}\right)} \tag{5.5}$$

The OIM maximizes the following expectation:

$$\mathcal{L} = \mathrm{E}_x \left[\log \mathbf{p}_t\right] \tag{5.6}$$

This loss function is an optimal choice for the person re-identification task or similar ones, where there are large numbers of classes but each class does not have more than several samples. This formation leads to the conventional classifiers not being learned efficiently.

In inference mode, the features of each frame are extracted from the CNN and the RoIs are extracted. The detected persons in the scene are framed by bounding boxes and passed through the rest of the feature extraction process and are compared with the gallery of the references. Based on the similarity of the target person with the references, the gallery instances are ranked.

Person tracking variant: To track each individual in a sequence of frames, we exploit the re-identification method explained above with some alteration as follows.

The main difference between the task of person tracking and re-identification is the presence of reference instances with which the target can be compared to. In re-identification methods, the gallery consists of multiple unique instances of individuals and the target can be identified based on the similarity between them. In person tracking in a large collection of videos, such as a gallery, does not exist and forming it would be a costly effort due to the lack of annotations and the large amount of human instances in the videos. Additionally, the RoI selection using the bounding boxes is an inefficient method in occluded scenes with multiple people interacting with each other.

We adapt the person search network by replacing the RoI extraction component with the output of the instance segmentation method explained in Section 5.1.1. Replacing the bounding box extraction with the pixel-wise human instance segmentation would increase the robustness of the re-identification in occluded settings. Given the input sequence as $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots, \hat{\mathbf{x}}_N\}$, where $\hat{\mathbf{x}}_n$ is the masked input video frame. The first step is to extract the features of the segmented instances as $\mathbf{FV} = \{\mathbf{fv}_1, \mathbf{fv}_2, \dots, \mathbf{fv}_R\}$ where $\mathbf{fv}_r$ is the extracted feature associated with the r-th person in the frame.

Since there is no reference available, we initialize the gallery with the first instance of a person in the first frame as pid_1. If there are multiple persons in one frame, both are added in the gallery as $\mathbf{g} = \{pid_1, pid_2\}$. As soon as the new frame is fed to the feature extraction pipeline, the features of each masked human instance is extracted and is compared with the gallery as the query image. According to the similarity score, either the person in the frame will get the same pid as the reference or will be added as a new pid_3 to the gallery. The similarity score threshold σ is set to 0.5 (see Figure 5.12).

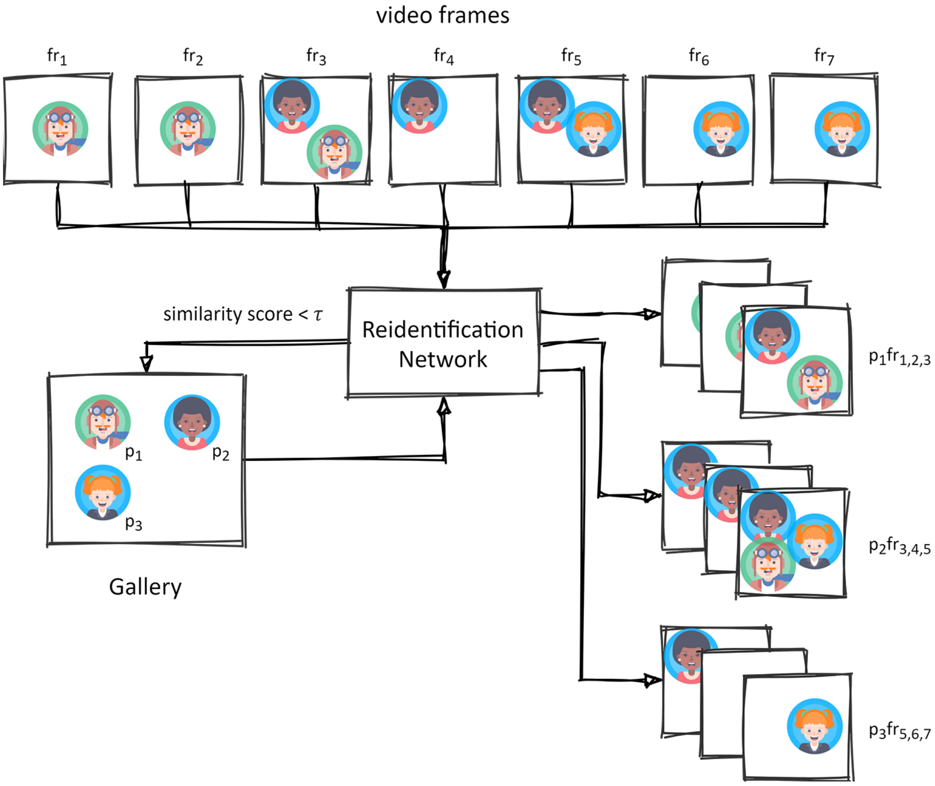


Figure 5.12. An illustrative configuration of the cross-angle person tracking module. The gallery is collecting the first instance of each person to use as the reference when the new query image appears. The output are short clips based on the continuous presence of a person in adjacent frames.

After the identification of each instance in the frames, and adding the queries with similarity score $< \sigma$ to the gallery, we stitch the consecutive frames in which one *pid* is present. Given $\mathbf{S} = \{\mathbf{fr}_1, \mathbf{fr}_2, \ldots, \mathbf{fr}_N\}$ as the sequences of video frames, $\mathbf{Sc}_{n,i}$ the clip containing the pid_i is formed as:

$$\mathbf{Sc}_{n,i} = \{\forall \mathbf{fr}_j \in \mathbf{S} \mid pid_i \in \mathbf{fr}_j \wedge (pid_i \in \mathbf{fr}_{j-1} \vee \mathbf{Sc}_{n,i} = \emptyset)\} \tag{5.7}$$

In other words, the first masked instance of pid_i in a video initiates a short clip where the next frame is stacked onto, in case the pid_i is present in that frame. This process continues until the person instance is not in the frame, and the clip is ended. Therefore the output will be multiple sequences of clips containing each person in the video.

5.2 Temporal Gesture Segmentation

In general, we can divide the type of videos we encounter into two main categories considering the number of gestures articulating in there (as shown in Figure 5.13):

- Isolated gesture videos, where the video clip only contains one gesture instance,
- Continuous gesture videos, where the video clip contains more than one gesture instance.

The algorithm to detect and recognize the gestures should either inherently be able to detect the gesture instances temporally during feature extraction, or a mechanism in preprocessing should segment these gestures before the feature extraction.

Usually in real-world applications, there rarely is only one gesture in the entire video, that is why separating the gestures from each other, i.e. finding the start and end frame number or time of a gesture is an important task. We have proposed two different methods of temporal segmentation to parse multiple gestures happening in one video in the preprocessing part.

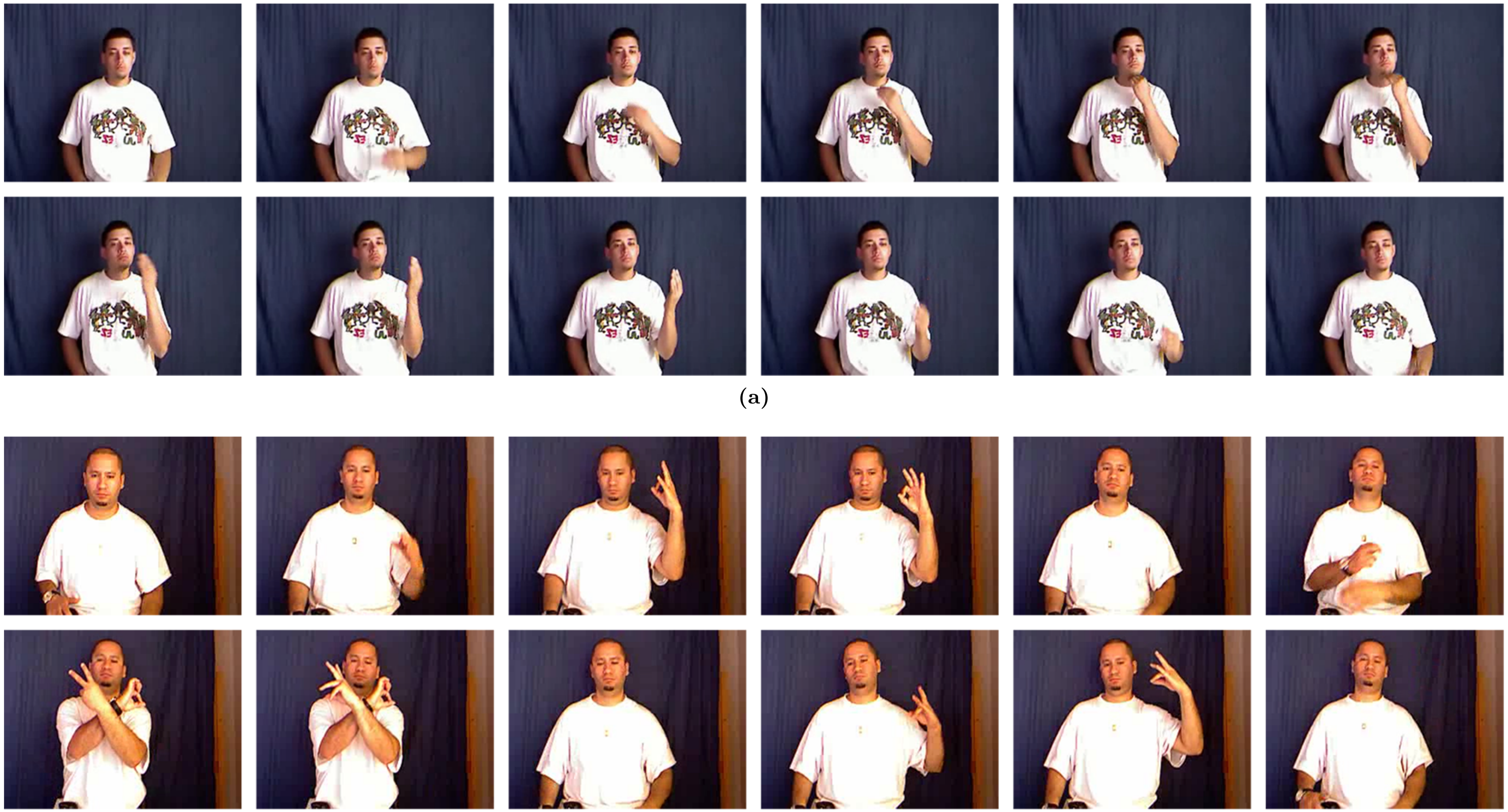
5.2.1 Binary Classification

Since the temporal localization is added to the rest of preprocessing steps, we would like to keep this step lightweight and with low computational cost. Therefore, inspired by [182] we design a lightweight binary classifier with the goal of segmenting gestures temporally before feature extraction. We denote the sequence of frames in the video as $\mathbf{S} = \{\mathbf{fr}_1, \mathbf{fr}_2, \ldots, \mathbf{fr}_N\}$. We introduce the window $\mathbf{Wn}$ as:

$$\mathbf{Wn}(a, b) = \ \mathbf{fr}_i, i \in [a, b] \tag{5.8}$$

which is a subset of $\mathbf{S}$ containing all the frames between $\mathbf{fr}_a$ and $\mathbf{fr}_b$ inclusively. The sliding window $\mathbf{S}_{wn}$ is defined as:

$$\mathbf{S}_{Wn} = \mathbf{Wn}(i - m + 1, i) \tag{5.9}$$



(a)

(b)

Figure 5.13. A set of frames from examples of isolated (5.13a) and continuous (5.13b) gesture videos from Chalearn Dataset

where m is the length of the sliding window and a predefined parameter. For the binary classification we use a 3D convolutional network which has the advantage of capturing the spatio-temporal dependencies between the video frames.

Following [182], we build the network with three 3D convolution, two maxpooling and three fully connected layers. The training is performed with cross-entropy loss minimization for 27 labels of Jester [4] dataset. At the the inference, 26 labels related to hand gestures were assigned to 1 (including "*doing other things*") and one label which indicates the absence of gesture ("*no gesture*") is mapped to 0.

The input to the network is the sliding window $\mathbf{S}_{Wn}$ on the entire sequence of $\mathbf{S}$. The output of softmax activation is:

$$f(fc_i) = \frac{e^{fc_i}}{\sum_j^L e^{fc_j}} \tag{5.10}$$

where fc_i denotes the elements of the output of fully connected layer. The cross entropy loss for the L classes is defined as:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{l} t_i \log\left(f(fc_i)\right) \tag{5.11}$$

where t_i is the correct label.

To determine the intervals where the gesture is not present in inference mode, we assign the "*no gesture*" label with the probability p as a binary class to the window of frames $Wn(a, b)$. With each stride of the window $i + 1$ a new class label is given to the window frames with the probability of p_{i+1}. This process continues until the window covers the entire sequence of frames. The final label is assigned to each label by averaging the probability of each frame collected by each slide of the window $\{p_i\}_i^{i+m}$ i.e.,:

$$p_{fr_i} = \frac{\sum_{j=i}^{i+m} p_j}{m} \tag{5.12}$$

where p_{fr_i} is the average probability of the frame fr_i. With defining the threshold τ we can assign label 0 to frames with $p_{fr_i} > \tau$ which have the label "*no gesture*". To make sure the hand movements which are forming a meaningful gesture are taken into account, we only consider sequences with more than 10 consecutive frames having label "*gesture*". Figure 5.14 illustrates the temporal segmentation using the binary classification.

5.2.2 Quantity of Movement (QOM)

The second method employed for segmenting gestures temporally, is inspired by [183] to measure the inter-gesture movement based on the initial position of the hands. Originally depth video frames were used to calculate this statistical measure. Since our method is strictly developed without depth information available we alter the QOM measure by using grayscale

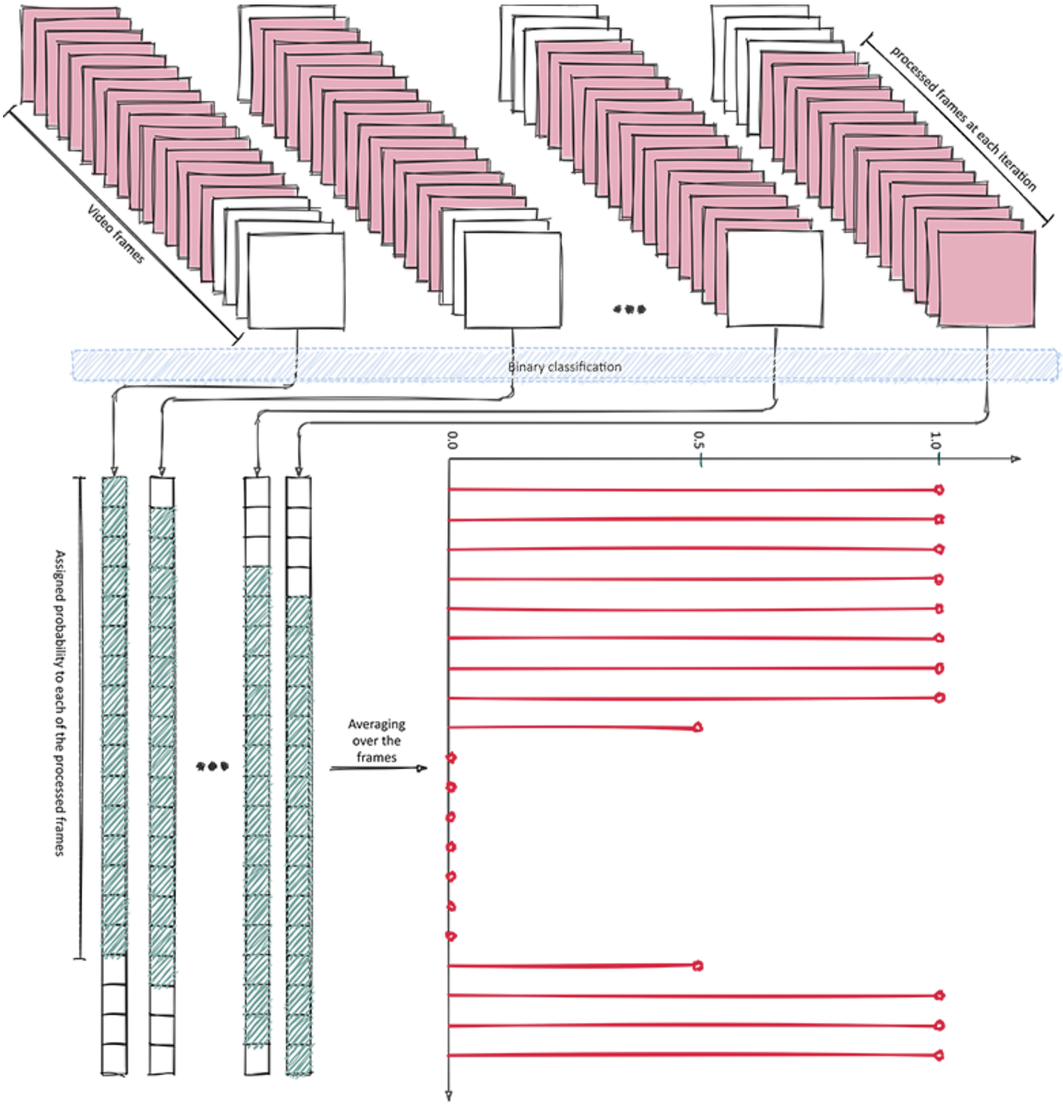

Video frames
processed frames at each iteration
Binary classification
0.0
0.5
1.0
Assigned probability to each of the processed frames
Averaging over the frames

Figure 5.14. The gesture temporal segmentation using binary classification which assigns binary label 1 or 0 to a window of frames based on the probability of presence of a gesture.

images instead of depth frames. The QOM has two individual components to measure the local and global movements of frame **fr** in sequence of frames **S**:

$$QOM(\mathbf{S}, \mathbf{fr}) = [QOM_{\text{Local}}\,(\mathbf{S}, \mathbf{fr}), QOM_{\text{Global}}\,(\mathbf{S}, \mathbf{fr})] \tag{5.13}$$

The $QOM_{\text{Local}}\,(\mathbf{S}, \mathbf{fr})$ component measures the movement of the frame **fr** compared to adjacent frames as:

$$QOM_{Local}(\mathbf{S}, \mathbf{fr}) = \sum_{m,n} \delta(\mathbf{S}_{fr}(m, n), \mathbf{S}_{fr+1}(m, n)) \tag{5.14}$$

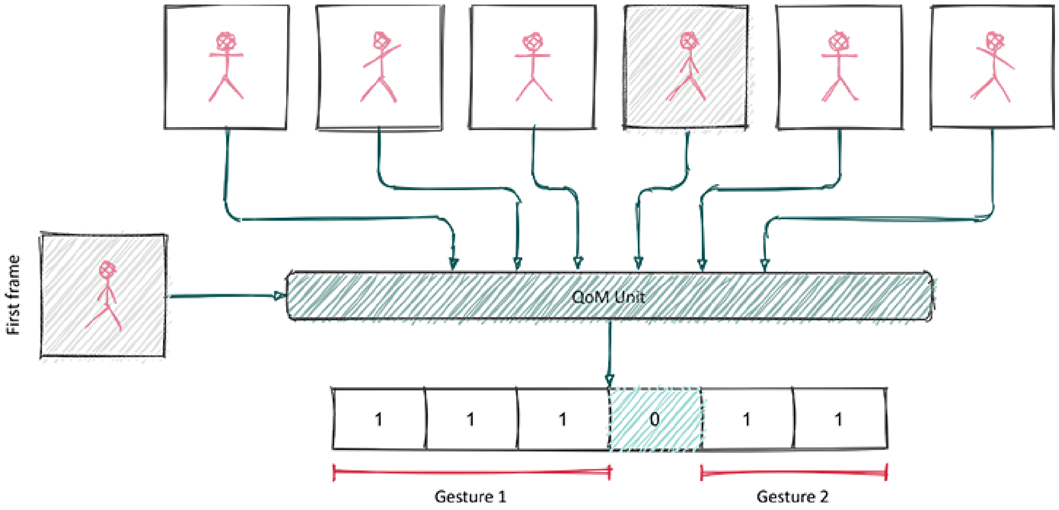


Figure 5.15. QOM gesture temporal segmentation localizes gestures based on their relative location to the reference position.

and the QOM_{Global} $(\mathbf{S}, \mathbf{fr})$ is calculating the movement quantity of frame **fr** compared to the first frame as:

$$QOM_{Global}(\mathbf{S}, \mathbf{fr}) = \sum_{m,n} \delta(\mathbf{S}_{fr}(m,n), \mathbf{S}_1(m,n)) \tag{5.15}$$

where $\mathbf{S}_{fr}$ is the frth frame in sequence $\mathbf{S}$, (m, n) are the pixels of the frame and $\delta(;\dot{)}$ is an indicator function defined as:

$$\delta(x,y) = \begin{cases} 1 & \text{if } |x-y| \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{5.16}$$

where τ is a hyper parameter empirically determined as 40. To mitigate the effect of similar background color interfering with the detection of the hand skin color, we use the spatial segmentation prior to this step and feed the masked frames to the QOM unit. The idea of the QOM_{Global} comes from the assumption that in the beginning of the video there is a neutral pose, where the hands are in the resting position. This position is used as a reference to calculate the amount of movements of the hands. We infer a binary classification using the QOM_{Global} for each frame as:

$$\text{label}_{QOM}(\mathbf{fr}) = \begin{cases} 1 & \text{if } QOM_{\text{Global }(\mathbf{S},\mathbf{fr})} \geq \eta \\ 0 & \text{otherwise} \end{cases} \tag{5.17}$$

where η is the intra gesture threshold, calculated by sum of the average and double the standard deviation of QOM_{Global} . Using the $\text{label}_{QOM}(\mathbf{fr})$ we create candidate subsets of the sequence S where the gestures are more likely to occur:

$$\mathbf{S}_{ca} = \{\mathbf{fr} \mid QOM_{\text{Global}}\ (\mathbf{S}, \mathbf{fr}) \leq \tau \wedge d(\text{label}_{QOM}(\mathbf{fr}))/dt = 0, \mathbf{fr} \in \mathbf{S}\} \tag{5.18}$$

where $d(\text{label}_{QOM}(fr))/dt$ is the derivative of $\text{label}_{QOM}(fr))$ specifying where the label toggled from 0 to one or vice versa. The $\mathbf{S}_{ca}$ contains the frames with $\text{label}_{QOM}(fr) = 1$ and that their adjacent frame also has the same label. These candidates are used as segmented clips in the next step for gesture feature extraction (see Figure 5.15).

5.2.3 Remarks on Temporal Localization Methods

The two proposed methods for detecting hand gestures temporally are sufficiently low cost from computational point of view and can be integrated in the preprocessing module. However, there are some advantages and disadvantages for each method as:

- Since the QOM method is originally proposed for depth modality, it is prone to misdetect gestures which have motion toward the camera
- The binary classification method distributes the probability of the presence of gestures over a window of frames, which eventually reduces the accuracy of the method.
- Both methods have difficulty to recognize gestures which are consecutively performed by the person without the resting period.

We will further discuss the performance of the two methods in Chapters 8 and 10

Chapter 6

Gesture Representation Learning

Feature extraction is considered as the core of a gesture recognition and retrieval method and is responsible for extracting the most informative representation of the video clips containing humans performing gestures. The input to this part is the output of the preprocessing step which has prepared the data to be mapped to the feature space.

In this chapter we describe three individual feature extraction methods that we used in the gesture recognition and retrieval system and further used for evaluation:

- RGB-based gesture feature extraction,
- Pose-based gesture feature extraction,
- Dimensionality reduction for gesture feature learning

The entire process is designed to be independent of depth modality and to rely only on RGB input videos. Our proposed models use additional information obtained from RGB data, such as optical flow and pose information to aid the feature extraction process. As a suggestion for reducing the computational complexity of the gesture feature extraction, we explore using the content based binary representation learning method for retrieval.

6.1 RGB-Based Gesture Feature Extraction

When talking about videos, there are two individual aspects that need to be considered: the spatial dimension of the frames, which is represented by pixel values, and the temporal dimension of the video, which is the relationship between the content of the frames in time. To describe a video in whole, we need to model both the temporal and the spatial dimensions. This is the reason it is essential for a video feature extraction method to be able to extract spatio-temporal features from the input data.

One way to model the spatial and temporal dimensions of a video, is to extract features using independent streams of information. One of the methods which has proven to be useful in modeling temporal dimension along with the image frames in the field of action recognition is I3D [13]. This network uses 3D kernel convolutions and is based on the InceptionV1 [12] or GoogLeNet model.

In hand gesture related problems, the large spatial variations of the gestures as well as the high temporal dependency between the frames, requires a network deep enough to extract discriminative features, and at the same time, being able to model the coarse hand shapes and motions. The InceptionV1 with different sizes of kernels is a great candidate to be used to extract discriminative features from real-world hand gestures. The architecture of this model is shown in Figure 6.3.

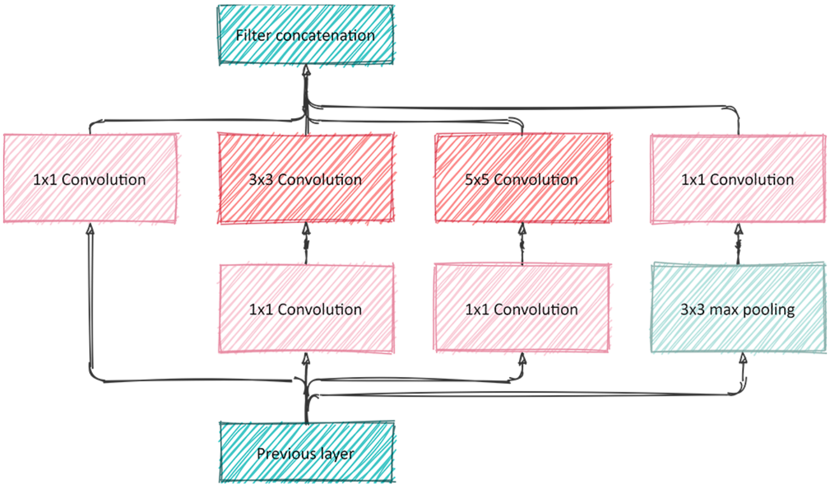
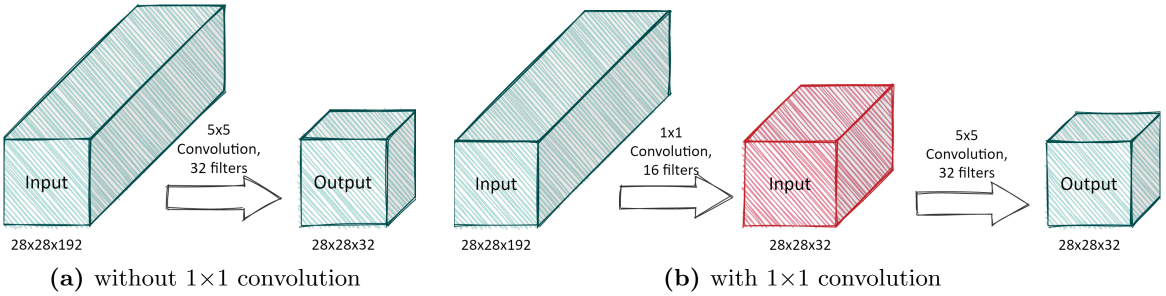


Figure 6.1. The network configuration of the Inception module used in the InceptionV1 [12].

InceptionV1 is the first of the trilogy of the Inception models with 27 layers. The network name comes from a module used in this network called *inception*. This module is essentially a combination of some individual components as:

- 1×1 Convolutional layer, as the dimensionality reduction element (due to the lower number of filters),
- 3×3 Convolutional layer, capturing local details,
- 5×5 Convolutional layer, capturing higher abstractions.

The "network in network" idea in the inception model reduces the computational complexity by using the 1×1 convolutions as dimensionality reduction unit (see Figure 6.2). The output of this block is concatenated and used as the input for the next block.


Input
5x5 Convolution, 32 filters
Output
28x28x192
28x28x32
Input
1x1 Convolution, 16 filters
Input
5x5 Convolution, 32 filters
Output
28x28x192
28x28x32
28x28x32

(a) without 1×1 convolution **(b)** with 1×1 convolution

Figure 6.2. The difference between 1×1 convolutional blocks and how they reduce the dimension of the input.

I3D takes the inception network one step further by inflating its convolutional filters to extract spatio-temporal features from video input. The filter inflation is essentially transforming all the pooling and convolutional kernels from $N \times N$ to $N \times N \times N$.

Additionally, I3D introduces the reuse of pretrained model's weights of the GoogLeNet for the video input by bootstrapping the parameters. The bootstrapping is essentially considering a video made of one repeated image and repeating the weights of the 2D filters N times. The model has two streams of data as input, the RGB and optical flow. We build our feature extraction method based on the network architecture of I3D, following the author's suggestion to use optical flow to add a form of recurrent flow of information (due to the iterative optimization for the flow fields) to capture the temporal dimension.

The original I3D network is trained for classification of actions. Using transfer learning, we change the objective of the method, by using the pre-trained weights and altering the architecture to fit our purpose of extracting features from gesture videos. For this purpose, we use the original model with the pretrained weights on Imagenet [184] and Kinetics 400 [185]. Since the kinetics 400 dataset is rich in activities involving humans, using the pre-trained model is a good starting point for fine-tuning the network for a gesture recognition task. Generally, neural networks tend to extract more local features in the lower layers of the networks and the deeper they get, the more semantic and abstract features are extracted [186]. In hand gesture recognition, the interpretation and the semantic meaning of hand movements usually comes with text and speech and does not necessarily exist in pure video frames. Therefore, we believe mitigating the conceptual inference of the network by extracting features from intermediate layers of the network could help to leverage the existence of local descriptors and to model compact spatio-temporal video representations.

The feature extraction begins with a uniform sampling of video frames $\boldsymbol{S} = fr_1, \ldots, fr_N$, to create a subset $\boldsymbol{S_s}$ of the video frame sequence $\boldsymbol{S_s} \subseteq \boldsymbol{S}$ with 40 frames. Following the recommendation of the authors of I3D, we use RGB $\boldsymbol{S}_s^{rgb}$ and the optical flow $\boldsymbol{S}_s^{of}$ modalities as input. To compute the optical flow, we use the TV-L1 algorithm [187] which effectively detects the displacement of pixels in two consecutive frames. For our training dataset, Chalearn Iso [3] we have a solid assumption that the background is stationary and the only motion in the videos are the result of hand and body movement.

The two streams of RGB and optical flow are processed through multiple modules of the I3D network. To select the layer from which we use the output to generate the video representations, we made five divisions of the I3D network and introduced five modules which are essentially combinations of convolutional layers or inception blocks followed by a pooling layer as shown in Figure 6.4. We have selected the output of the maxpool of the fourth module for each stream with their output feature maps fm_4^{rgb} and fm_4^{of} fused to make a single feature map as fm_4. In order to decide whether early or late fusion results in most descriptive and discriminative features from different streams, we have observed the accuracy and loss of training and validation set of the ChalearnIso dataset (Table 6.1). The two configurations of the network for early and late fusion are shown in Figure 6.5a.

To select which module output is most effective for our objective, we ran several experiments to measure the accuracy and loss on training and validation set of the ChalearnIso dataset. Table 6.2 shows the accuracy and the loss on training and validation sets obtained by the output of different stages of the I3D network for individual (RGB and optical flow) as well

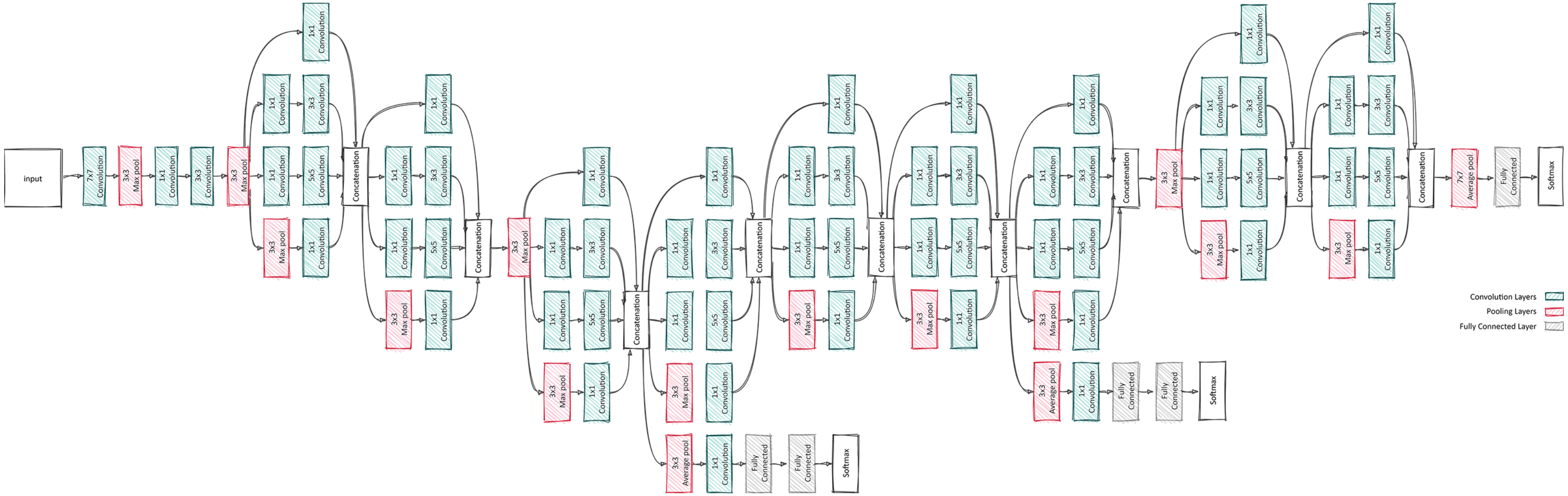


Figure 6.3. The Complete network architecture of InceptionV1, also called GoogLeNet. (The Figure is redrawn from [12])

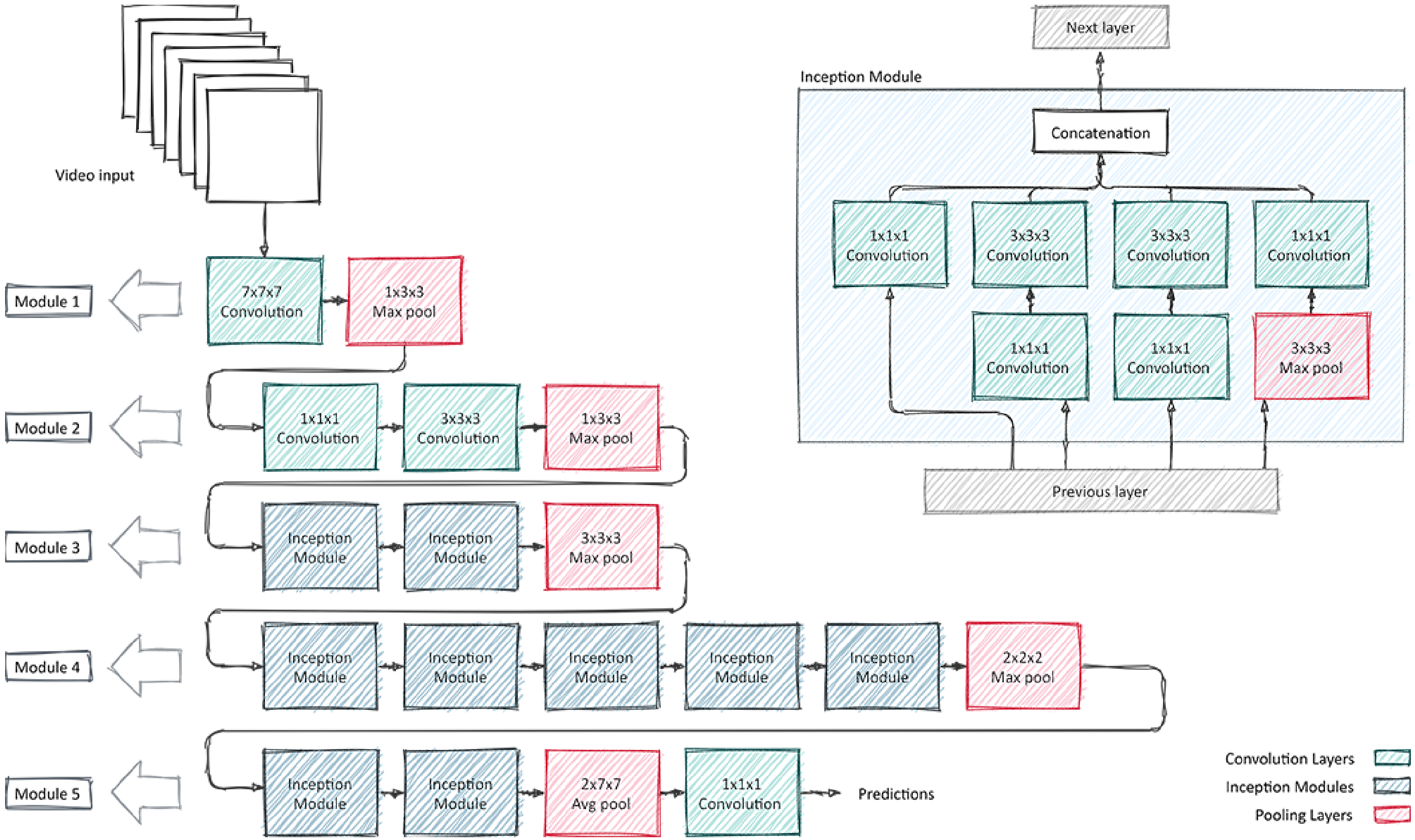
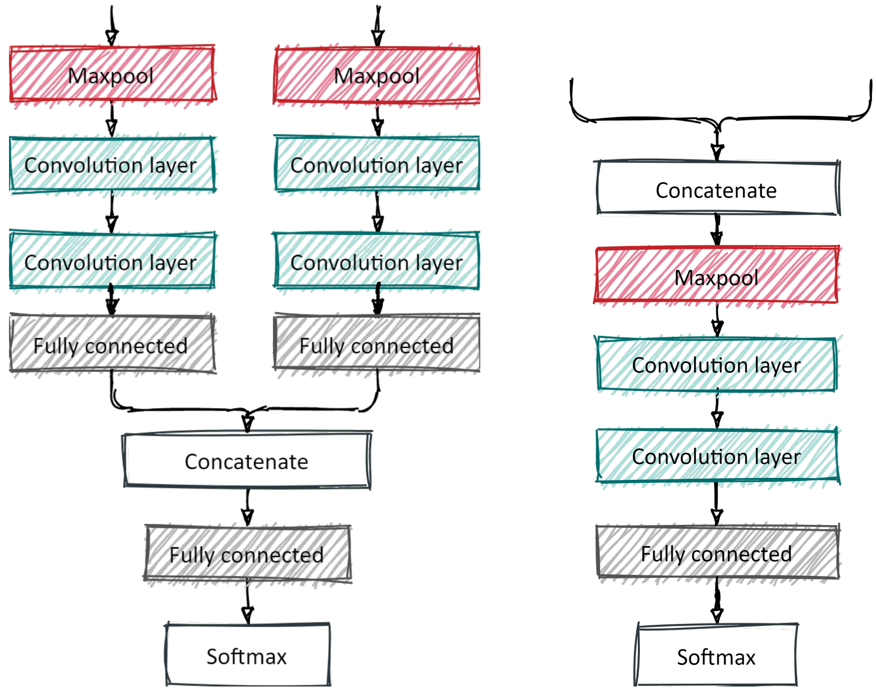


Figure 6.4. The network architecture of I3D with inflated filters (The figure is redrawn from [13]).



(a) Configuration of the added layers with late fusion

(b) Configuration of added layers with early fusion

Figure 6.5. Two different network configurations used for I3D with early and late fusion techniques.

as the fusion of two modalities. The results are averaged between five runs of the training at the end of 20 epochs.

Table 6.1. The accuracy and loss on training and validation set of ChalearnIso dataset with the output of fourth module of I3D network using early and late fusion. The best value of each modality at each set is shown in boldface.

	Accuracy (training)	Accuracy (validation)	Loss (training)	Loss (validation)
Early Fusion	**98.7%**	**77%**	**0.02**	**2.1**
Late Fusion	88%	53%	3.45	8.7

The results obtained by the output of the fourth and fifth modules are relatively close to each other for each individual modalities (RGB and optical flow), but the fusion results work best with the extracted features from the fourth I3D network module. It is worth noting that the accuracy of the last layer for the RGB modality for both the training and the validation set was slightly better than module four, but when testing, the results were not comparable, and the accuracy on test set for the RGB modality dropped significantly to 40%, which indicates the lack of generalization of the model for the RGB modality.

Table 6.2. The accuracy and loss on training and validation set of the ChalearnIso dataset with the output of each module of I3D network for each individual modality and the fused streams. The best value of each modality at each set is shown in boldface.

	Module	**RGB flow**	**Optical Flow**	**Fusion**
Accuracy (training)	Module 1	9.2%	13.1%	17.2%
	Module 2	15.9%	18.5%	30.4%
	Module 3	43%	48.8%	71.3%
	Module 4	91.8%	**96.9%**	**98.7%**
	Module 5	**92.9%**	88%	93%
Accuracy (validation)	Module 1	8.1%	5%	12.6%
	Module 2	15.7%	25.8%	28%
	Module 3	35.8%	51.1%	41.5%
	Module 4	55.9%	**67.2%**	**77%**
	Module 5	**67.1%**	54%	71%
Loss (training)	Module 1	15.52	3.25	7.4
	Module 2	5.07	3.6	3.11
	Module 3	2.41	0.25	2.39
	Module 4	0.11	0.1	**0.01**
	Module 5	**0.02**	**0.01**	0.02
Loss (validation)	Module 1	23.9	41	20.2
	Module 2	15.3	15.4	13.2
	Module 3	12.1	6.9	5.8
	Module 4	4.5	**2**	**2.1**
	Module 5	**3.9**	7.3	3.5

Depending on which task of recognition or retrieval we would like to perform, we propose different approaches.

6.1.1 For Gesture Recognition:

For gesture recognition, the model needs to predict a probability for each of the class labels. This probability is calculated by passing the fm_4 through two convolutional layers, followed by a fully connected and a softmax layer. The parameters of the newly added layers are trained by minimizing the cross entropy loss $\mathcal{L}_{top}$ and after that, the entire network is once more trained to adjust the weights to the new objective by minimizing the cross entropy loss $\mathcal{L}_{all}$. The newly proposed architecture is shown in Figure 6.6.

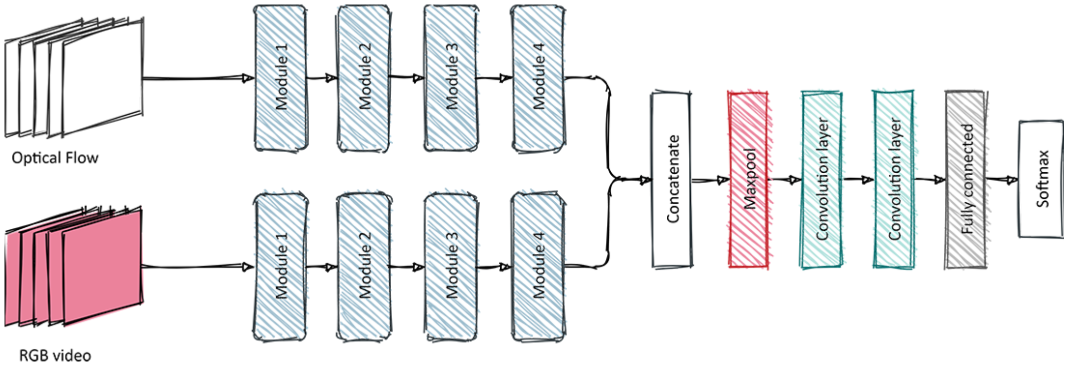


Figure 6.6. The network configuration for gesture recognition using I3D as base model.

At the inference time, a video which we would like to classify and label, will be fed to the network and after feature extraction, the output will be a text label, determining the class of the input gesture video.

6.1.2 For Gesture Similarity Learning and Retrieval

For Gesture Similarity Learning and Retrieval, we are interested in using a query video, search the collection and retrieve the similar gesture videos. Therefore, it is essential to train the network to learn the similarity metric between the dataset samples. It is worth mentioning that this type of learning-based similarity retrieval works perfectly when the collection of the videos are *static* or *semi-static*, meaning that the entire collection is known in advance or changes are limited. However, in *dynamic* collections depending on the new videos (if they are outliers or very different from the collection or number of changes are high), optimally the feature extraction model needs to be re-trained to fit the current collection. In such cases the more traditional approach of encoding the objects in a feature space would be more useful.

The objective of this gesture similarity retrieval is to compute the distance between the query object and each video in the collection and provide a ranked list of videos which are distance-wise similar to the query. For this purpose, we use the altered architecture introduced for the classification task with removing the softmax layer to obtain a 2048 dimension feature vector. This feature vector is used to train the network to learn the similarity metric between the dataset samples. We use two methods of similarity learning by Triplet and Contrastive networks.

To measure the similarity between the samples we use the euclidean distance. For Triplet network, a collection of triplets $\tau = (\boldsymbol{fv}_i, \boldsymbol{fv}_i^+, \boldsymbol{fv}_i^-), i = 1, ..., k$ are drawn, where $\boldsymbol{fv}_i, \boldsymbol{fv}_i^+, \boldsymbol{fv}_i^-$ are the feature vectors of the anchor, positive and negative samples, respectively. To encourage the network to learn diverse similarities, we introduce a margin μ. Therefore, the relation of similarity between two pairs is defined as:

$$D(f_\theta(\boldsymbol{fv}_i), f_\theta(\boldsymbol{fv}_i^+)) - D(f_\theta(\boldsymbol{fv}_i), f_\theta(\boldsymbol{fv}_i^-)) + \mu < 0 \tag{6.1}$$

where $\mu = 0.5$. The triplet loss is defined as following:

$$\mathcal{L}_\theta(\boldsymbol{fv}_i, \boldsymbol{fv}_i^+, \boldsymbol{fv}_i^-) = max(\boldsymbol{D}(f_\theta(\boldsymbol{fv}_i), f_\theta(\boldsymbol{fv}_i^+)) - D(f_\theta(\boldsymbol{fv}_i), f_\theta(\boldsymbol{fv}_i^+)) + \mu, 0) \tag{6.2}$$

In order to fulfill the object of the retrieval task, to ensure the similar samples with smaller distance are mapped to closer embeddings, we minimize the loss function in Equation 6.2:

$$\min_\theta \sum_{i=1}^{k} \mathcal{L}_\theta\left(\boldsymbol{fv}_i, \boldsymbol{fv}_i^+, \boldsymbol{fv}_i^-\right) \tag{6.3}$$

where k is the total number of triplets. To sample the triplets, we select the anchor class randomly, and positive and anchor samples share the same class label. To mine the negative

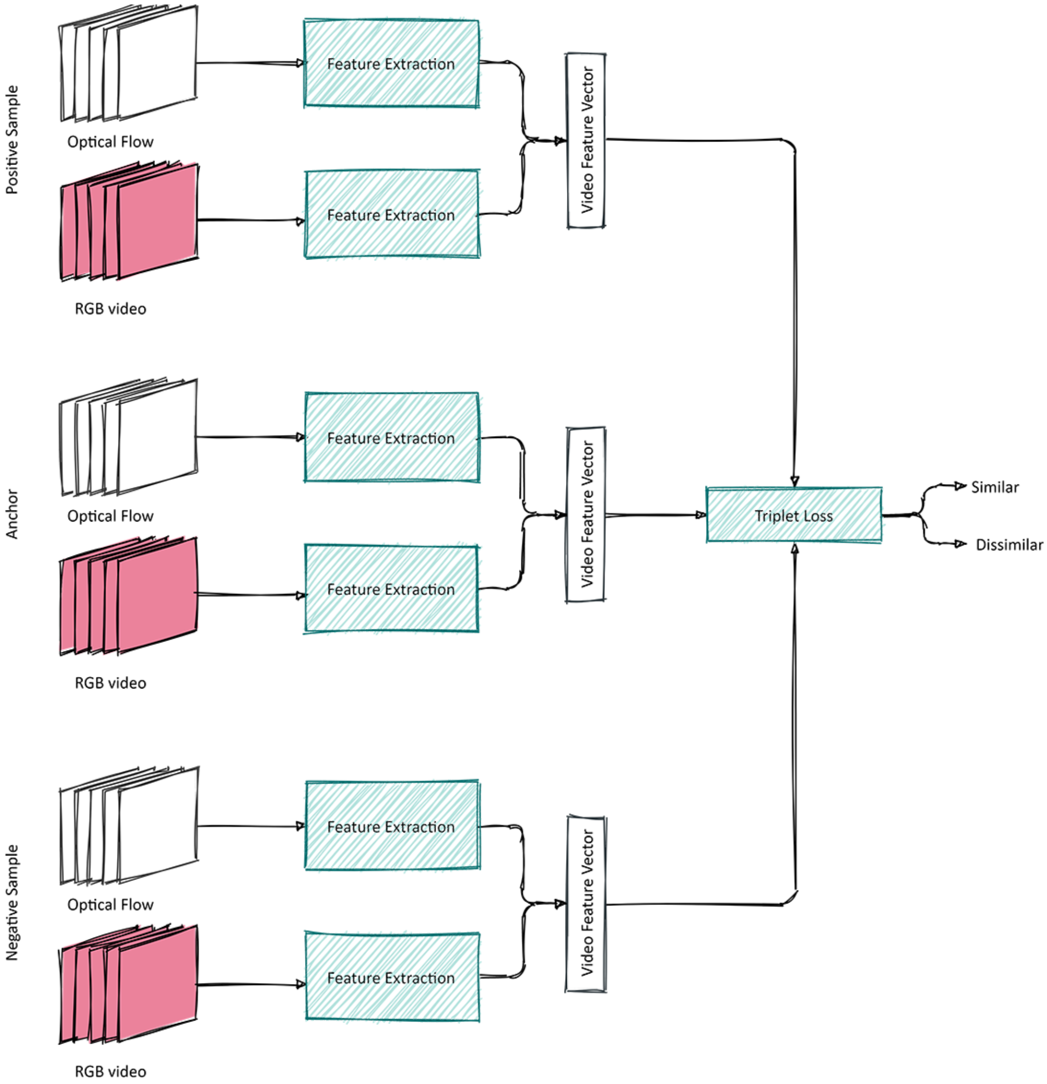


Figure 6.7. A configuration overview of the gesture retrieval with triplet similarity metric learning. In this model three samples as anchor, positive and negative are fed to the network.

samples, we consider all the classes except the anchor class. A schematic configuration of the similarity metric learning is shown in Figure 6.7.

For the contrastive network instead of three samples, we need similar and dissimilar pairs. According to the definition, given a pair of videos embedding $\boldsymbol{fv}_i, \boldsymbol{fv}_j$:

$$\mathcal{L}\left(f_\theta(\boldsymbol{fv}_i), f_\theta(\boldsymbol{fv}_j), s_{ij}\right) = \begin{cases} \frac{1}{2}\left\|f_\theta(\boldsymbol{fv}_i) - f_\theta(\boldsymbol{fv}_j)\right\|_2^2 & \text{if } s_{ij} = 1 \\ \frac{1}{2}\max\left(0, \mu - \left\|f_\theta(\boldsymbol{fv}_i) - f_\theta(\boldsymbol{fv}_j)\right\|_2^2\right) & \text{if } s_{ij} = 0 \end{cases} \tag{6.4}$$

where s_{ij} is the similarity label between the pair. The objective is to minimize this loss function as:

$$\mathcal{L}\left(\boldsymbol{fv}_i, \boldsymbol{fv}_j, s_{ij}\right) = s_{ij}\left\|\boldsymbol{D}(f_\theta(\boldsymbol{fv}_i) - f_\theta(\boldsymbol{fv}_j))\right\| + (1 - s_{ij})\max\left(0, \mu - \left\|\boldsymbol{D}(f_\theta(\boldsymbol{fv}_i) - f_\theta(\boldsymbol{fv}_j)\right\|\right) \tag{6.5}$$

The samples needed for the training are collected either from the same class (similar) or from different classes (dissimilar). A Schematic configuration of the network with contrastive similarity metric learning is shown in Figure 6.8.

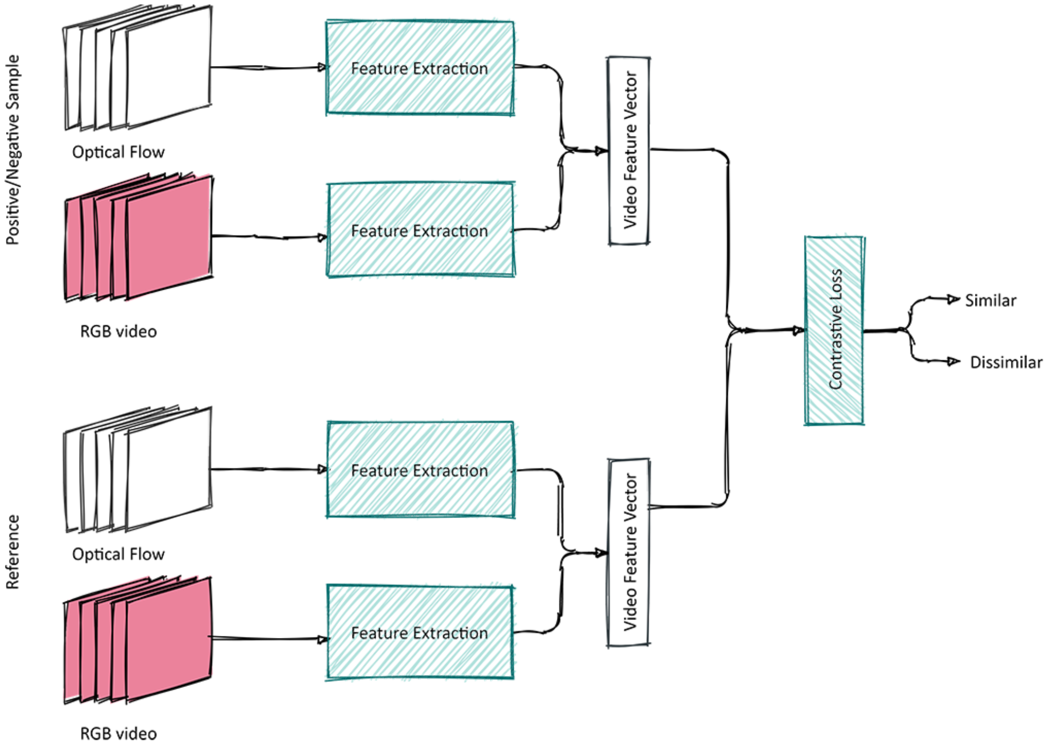


Figure 6.8. A configuration overview of the gesture retrieval with contrastive similarity metric learning. In this model two samples with either similar or dissimilar label are fed to the network.

At the inference time, a query video is input to the network and its spatio-temporal feature is extracted. Then, the Euclidean distance of this feature with the features of the collection which are pre-computed and stored in the database, is computed, and a ranked list of videos based on the lowest distance is retrieved. The experiments related to the similarity learning using the two loss functions is presented in Chapter 9.

6.2 Pose-Based Gesture Feature Extraction

The complexity of tracing gestural trajectories in multi-perspective scenarios requires a robust feature extraction module which can represent discriminative information about hand articulations. Although the optical flow and the RGB data can contribute to extracting such features, when the interactions between people get complicated, having additional pose modality could help in recognizing and following the hand motions. Additionally, the optical flow is a robust motion modeling algorithm when the brightness is consistent and there are no abrupt motions. In scenes where camera cuts exist, this sudden big displacement of pixels would break-down the optical flow ability.

Therefore, in addition to the previously introduced model with RGB and optical flow input, we propose a method to incorporate pose information extracted from RGB input to be used in scenes which are challenging for optical flow based methods. Our proposed model uses the pose keypoints together with RGB stream of data to create an attention map, and is inspired by RPAN [14] which is an end-to-end RNN with a pose-attention mechanism that learns to focus on active human joint parts. This is especially important in the recorded footage of news or talk shows, where the majority of actions in the scene are hand motions. Therefore, we believe that the representations from this model fit very well for retrieval tasks in complex settings.

RPAN originally have two streams of RGB and optical flow, where each of them are trained with the human activity recognition objective. The training involves extracting the keypoints from the RGB and optical flow streams and using attention mechanisms to the estimated joint locations. Due to the limitations of optical flow in scenes with abrupt movements which are common in talk show footage because of camera cuts, we don't follow the RPAN model architecture. We replace the optical flow stream with pose keypoints streams, obtained in the preprocessing steps, and together with the RGB input, the network learns the attention weights to focus on the active human joints. For the sake of completeness, an overview of the original RPAN method is illustrated in Figure 6.9. In the following we explain in detail our method and how it is different from the RPAN architecture. However, since additions to this method are mainly regarding the similarity learning, the detailed experiments will be presented in Chapter 9.

Instead of using a TSN network with RGB and optical flow streams, we extract the spatial features from the RGB data input using a ResNet convolutional network. The features are used to generate a convolutional cube with the size of $K_1 \times K_2 \times n$ based on aggregating n feature maps with the dimension of $K_1 \times K_2$. Therefore, for each frame t, the convolutional cube has the form [14]:

$$\boldsymbol{C}_t = \boldsymbol{C}_t(1), ..., \boldsymbol{C}_t(K_1 \times K_2) \tag{6.6}$$

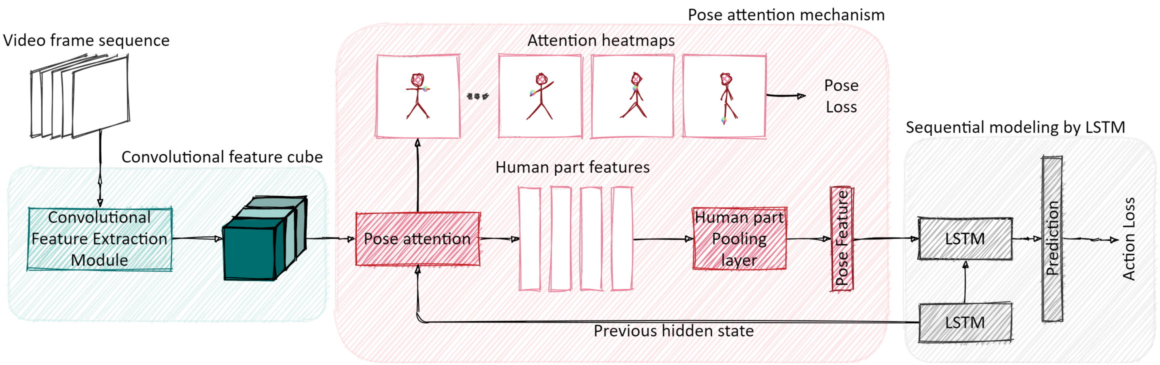


Figure 6.9. The architectural diagram of the RPAN method for action recognition. The method creates pose estimates as a bi-product together with action class probabilities.(the figure is redrawn from [14]).

which contains a feature vector at each k location $\boldsymbol{C}_t(k)$ where $k = 1, ..., K_1 \times K_2$.

The temporal dependencies between the video frames are modeled by LSTM units. To learn the dynamics of an activity and the fine-grained movements which comprise a certain gesture, it is important to include another level of supervision other than the gestural categories. Therefore, together with the RGB data, we input the joint keypoints extracted from a pose estimation method, which previously had been extracted in preprocessing methods. We follow the pose attention mechanism and the human part configuration from Du et al. [14]. The part configuration is the idea that a collection of joins usually involved in an activity together, and having the focus on these parts, would enable the model to learn the part-specific features. The configuration of the human body parts are shown in Figure 6.10.

Parts	Joints
Torso	1, 2, 3, 8, 9
Elbow	4, 5
Wrist	6, 7
Knee	10, 11
Ankle	12, 13

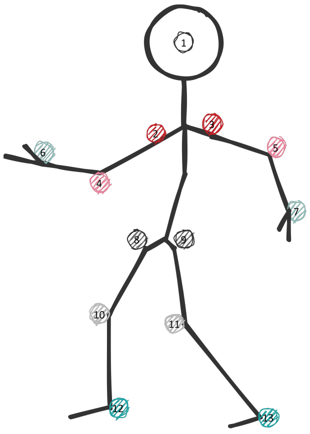


Figure 6.10. The assignments of the body joints to parts according to [14].

The pose attention mechanism is defined by an attention heatmap $\alpha_t^J(k)$ for each feature vector from Eq. 6.6 for each joint (J) which together with semantically relevant joints, form a body part structure (P):

$$\alpha_t^J(k) = \frac{\exp\{\boldsymbol{v}^J \tanh(\boldsymbol{A}_h^P \boldsymbol{h}_{t-1} + \boldsymbol{A}_c^P \boldsymbol{C}_t(k) + \boldsymbol{b}^P)\}}{\sum_k \exp\{\boldsymbol{v}^J \tanh(\boldsymbol{A}_h^P \boldsymbol{h}_{t-1} + \boldsymbol{A}_c^P \boldsymbol{C}_t(k) + \boldsymbol{b}^P)\}} \tag{6.7}$$

Here, $\boldsymbol{h}_{t-1}$ is the LSTM hidden state of the previous frame, $\boldsymbol{v}^J$, $\boldsymbol{A}_h^P$, $\boldsymbol{A}_c$, $\boldsymbol{b}^P$ are the attention parameters, all of which except $\boldsymbol{v}^J$, are shared between the joints ($J \in P$). Based on this attention heatmap, the human-part feature is extracted:

$$\boldsymbol{F}_t^P = \sum_{J \in P} \sum_k \alpha_t^J(k) \boldsymbol{C}_t(k) \tag{6.8}$$

After extracting all the human-part features from Equation 6.8, they are fused together by a pooling layer to generate pose-related features, $\boldsymbol{S}_t$. To capture the temporal dimension of the movements, the features are then fed to a LSTM and the ($\boldsymbol{h}_t$) is used to generate a prediction vector containing a probability for each class label at each frame t of video ($\hat{\boldsymbol{y}}_t$) in recognition task.

Training the network is done in end-to-end fashion and loss function is a cross entropy loss $\mathcal{L}_{\text{gesture}}$. In the original RPAN a pose-loss $\mathcal{L}_{\text{pose}}$ is also used to extract the joints in the absence of the joint keypoints. Since we have used this information as pre-computed to the network, we skip the $\mathcal{L}_{\text{pose}}$ minimization:

$$\mathcal{L}_{\text{gesture}} = \lambda_{\text{gesture}} \mathcal{L}_{\text{gesture}} + \lambda_\Theta \|\Theta\|_2 \tag{6.9}$$

where λ_{gesture} is loss coefficient for gesture and λ_Θ is a weight decay $\|\Theta\|_2$ is the L_2 regularization.

Even though we evaluate the system on a classification benchmark, the main goal of this method is to be used in retrieval tasks and to learn the similarity metric between the data points in the collection. Therefore, as in Section 6.1, we train the network for similarity metric learning using a triplet network. For this reason, we use the output of the LSTM unit after processing the entire video and map it to a one dimensional feature vector using two fully connected layers. An overview of the network with the newly added layers is shown in Figure 6.11. The similarity metric is an Euclidean distance and to train the network we minimize the triplet loss function λ_{triplet}.

For the retrieval process, all the features of the samples in the collection are extracted using the video data and the keypoints extracted using a pose estimation technique and stored in the database. The query type is the video as well, and undergoes the same feature extraction as above. Once the feature is extracted, the Euclidean distance between the query and the features in the database is computed and a ranked list of similar videos are retrieved.

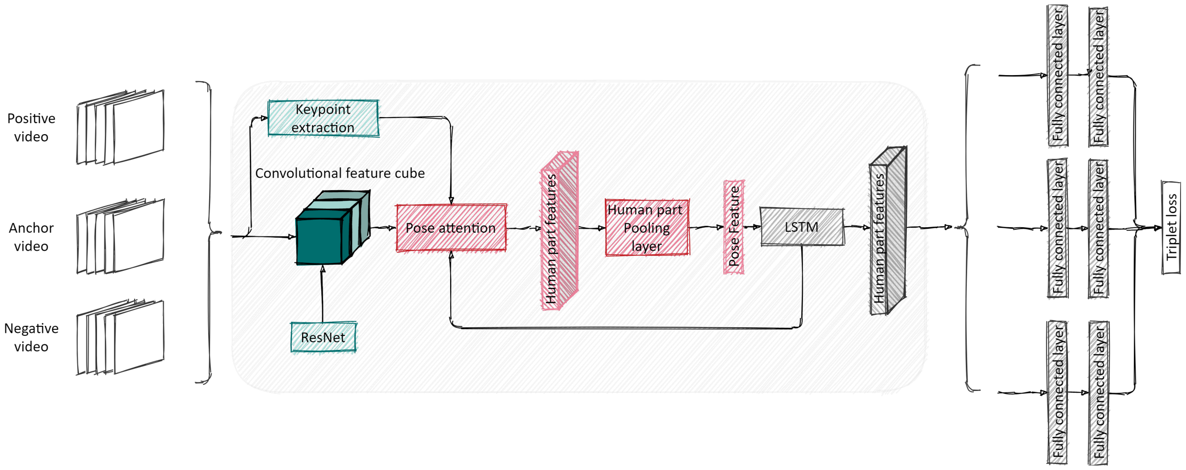


Figure 6.11. The proposed architecture diagram of the pose-based feature extraction and similarity learning with triplet loss.

6.3 Representation Dimensionality Reduction

Reducing the dimension and learning to hash methods became increasingly popular in large-scale information retrieval tasks. Mapping the high dimensional data into a compact and binary code reduces the retrieval computational cost and increases the speed. Although it is not the focus of this thesis, for the sake of completeness we discuss the possibility of using this approach and examine the potential improvement in the speed and results of the gesture retrieval method.

For this purpose we convert one of the existing image feature quantization techniques to be used in a video retrieval task. Deep Triplet Quantization (DTQ) [188] is a method built upon a convolutional neural network to extract the representations from input data and map these features into low dimensional binary space $\boldsymbol{fv} \mapsto \boldsymbol{b} \in \{0,1\}^{\beta}$. Given a collection of videos, we use the feature extraction method introduced in Section 6.1 to obtain the video spatio-temporal representations $\boldsymbol{fv}$. Essentially, given a triplet of videos as $\boldsymbol{T}_i = \boldsymbol{s}_i^a, \boldsymbol{s}_i^p, \boldsymbol{s}_i^n$ representing the anchor, positive and negative video samples respectively, the hashing method aims to map each sample to the feature space with the loss function as:

$$\mathcal{L}_{\text{triplet}} = \sum_{i=1}^{k} \mathcal{L}_i = \sum_{i=1}^{k} \max\left(0, \mu - \|\boldsymbol{fv}_i^a - \boldsymbol{fv}_i^n\|_2^2 + \|\boldsymbol{fv}_i^a - \boldsymbol{fv}_i^p\|_2^2\right) \tag{6.10}$$

where μ is the similarity margin and k is the number pf samples.

The quantization part of the method is based on a set of M codebooks $\boldsymbol{\mathcal{C}} = [\boldsymbol{\mathcal{C}}_1, \ldots, \boldsymbol{\mathcal{C}}_M]$ where each codebook has L codewords as $\boldsymbol{\mathcal{C}}_m = [\boldsymbol{\mathcal{C}}_{m1}, \ldots, \boldsymbol{\mathcal{C}}_{mL}]$ where $\boldsymbol{\mathcal{C}}_{mL}$ is a d-dimensional cluster center codeword. The assignment vector $\boldsymbol{b}_i = [\boldsymbol{b}_{1i}; \ldots; \boldsymbol{b}_{Mi}]$ is an indicator which of the codewords is used to approximate the feature $\boldsymbol{fv}_i$. The quantization objective ensures that each triplet sample is assigned to one of the L codewords using K-means as:

$$Q = \sum_{i=1}^{k} \sum_{j \in \{a,p,n\}} \left\| \boldsymbol{fv}_i^j - \sum_{m=1}^{M} \boldsymbol{\mathcal{C}}_m \boldsymbol{b}_{mi}^j \right\|_2^2 \tag{6.11}$$

To control the redundancy of the codewords a weak orthogonality is enforced also as:

$$Q = \sum_{i=1}^{k} \sum_{j \in \{a,p,n\}} \left\| \boldsymbol{fv}_i^j - \sum_{m=1}^{M} \boldsymbol{\mathcal{C}}_m \boldsymbol{b}_{mi}^j \right\|_2^2 + \gamma \sum_{m=1}^{M} \sum_{m'=1}^{M} \left\| \boldsymbol{\mathcal{C}}_m^\top \boldsymbol{\mathcal{C}}_{m'} - \boldsymbol{I} \right\|_F^2 \tag{6.12}$$

where γ is the degree of orthogonality. Training of the binary representation learning is done by minimizing the triplet loss and the quantization loss as:

$$\min_{\boldsymbol{\Theta}, \boldsymbol{\mathcal{C}}, \boldsymbol{B}^j} \mathcal{L} + \lambda Q \tag{6.13}$$

where $\lambda > 0$ is a hyper-parameter controlling the triplet and quantization loss and Θ is the feature extraction weight. The learning procedure follows the alternating optimization paradigm by updating one variable iteratively while other variables are fixed. The I3D network parameters θ can be learned via standard back propagation. To learn the codebook $\boldsymbol{\mathcal{C}}$ gradient decent algorithm i used. To learn the binary codes $\boldsymbol{b}$, all the criteria of the binary codes should be fulfilled as:

$$\min_{\boldsymbol{b}_i^j} \|\boldsymbol{fv}_i^j - \sum_{m=1}^{M} \boldsymbol{\mathcal{C}}_m \boldsymbol{b}_{mi}^j\|^2 \qquad \text{s.t. } \|\boldsymbol{b}_{mi}^j\|_0 = 1, \quad \boldsymbol{b}_{mi}^j \in \{0,1\}^L \tag{6.14}$$

where the ℓ_0 ensures each sample is approximated only by one codeword. To optimize equation 6.14, the authors suggest using the Iterated Conditional Modes (ICM) approach that solves $\left\{\boldsymbol{b}_{mi}^j\right\}_{m=1}^{M}$ alternatively.

At the retrieval stage, instead of the common hamming distance, Asymmetric Quantizer Distance (AQD) is used which is based on the inner product similarity between the binary codes:

$$\mathrm{AQD}\left(\boldsymbol{q}, \boldsymbol{x}_n\right) = \boldsymbol{fv}_q^{\mathrm{T}} \left(\sum_{m=1}^{M} \boldsymbol{\mathcal{C}}_m \boldsymbol{b}_{mn} \right) \tag{6.15}$$

where q is the query, $\boldsymbol{fv}_q$ is the deep representation of the query and $\boldsymbol{x}_n$ is the database point.To compute the AQD between the query and all elements of the database, the inner product between the feature vector $\boldsymbol{fv}_q$ and the all codebooks are precomputed and stored in a lookup table.

Chapter 7

Implementation Details and Setup

After describing the methodology behind each retrieval approach, in this chapter we introduce all the technical details of the implementations of the methods. In this chapter you will find the details about the networks used in each module and the choice of hyper parameters in each method. The implementations are done using Tensorflow [3], Keras [4], Pytorch [5] and in very rare cases the caffe [6] library. All the implementations and training are done on servers with 1080, 2080 and 3080 GPUs.

7.1 Preprocessing and Data Preparation

As described in Section 5, the components of this module prepare the data for feature extraction and learning the representation of the input data. However, the input data needs to be prepared before feeding this module. One of these data preparations is extracting the pose keypoints which in addition to the preprocessing module, is used in the posed-based representation learning module.

The pose information needed for both preprocessing and the feature extraction is obtained using OpenPose [10] key-point extraction. Openpose offers to extract 25 or 18 keypoints from 2D input. According to the requirements of the feature extraction module we use the 18 keypoint format without the foot keypoints. Figure 7.1 shows the difference between the two formats. As can be seen, the 25 keypoints include the foot keypoints, which are not informative in our problem.

The keypoint extraction takes the video data as input with arbitrary length and size and processes them at 30 frames per second. To make sure the reconstruction of the video after the segmentation and at the retrieval is possible, we need to record the frames where there are no human instances available as well. This way there will be no temporal shift in the output and any mismatch in the frame numbers after the process.

7.1.1 Cross-angle Spatio-Temporal Segmentation

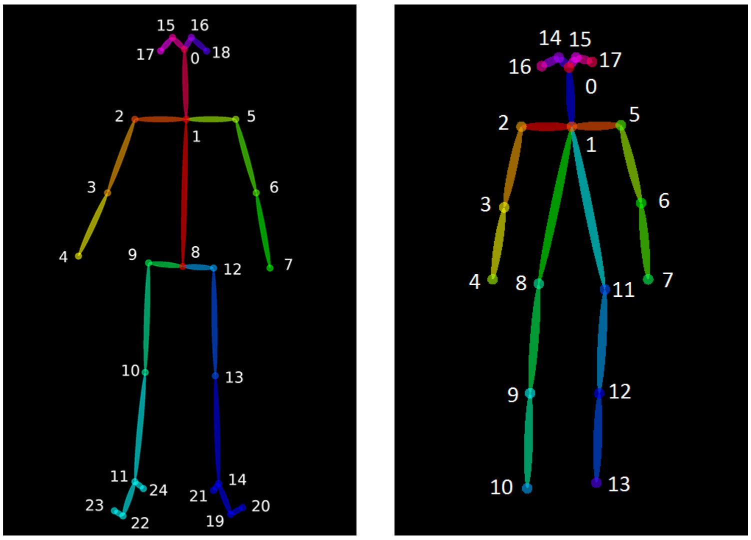
The spatial person segmentation is based on the official model released by the authors of Pose2Seg which is trained on the OCHumans dataset [8]. We use the video frames together

[3] https://www.tensorflow.org/

[4] https://keras.io/

[5] https://pytorch.org/

[6] https://caffe.berkeleyvision.org/



(a) (b)

Figure 7.1. The two keypoint configuration extracted by Openpose. According to the requirements of the feature extraction module, we use the 18 keypoints (b) format[7].

with the extracted key-points as the input. The output of the model is a saliency mask for each individual by removing the background iėreplacing the values of the pixels out of the mask by zero. The background removal especially causes the feature extraction network to be independent of the clutter and to extract the representations based on the actual motion of the individuals. The pose information input to the spatial segmentation model allows the segmentation of multiple persons even in occluded scenes.

The spatial person segmentation is based on the official model released by the authors of Pose2Seg which is trained on the OCHumans dataset [8]. We use the video frames together with the extracted key-points as the input. The output of the model is a saliency mask for each individual by removing the background i.e., replacing the values of the pixels out of the mask by zero. The background removal especially causes the feature extraction network to be independent of the clutter and to extract the representations based on the actual motion of the individuals. The pose information input to the spatial segmentation model allows the segmentation of multiple persons even in occluded scenes.

The input to the base network is the segmented masks of the spatial segmentation component, and the extracted features of the gallery are stored in the database. The features extracted from the query segment will be compared with the stored gallery feature entries in the database using the cosine similarity measure:

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} \tag{7.1}$$

Based on this measure, each query will get a similarity score which determines the person ID of that instance. To capture the continuous gesture articulation, a dictionary with the

[7] Images from: https://github.com/CMU-Perceptual-Computing-Lab/openpose

Table 7.1. The network architecture of the gesture classifier for temporal detection of hand gestures.

Type	Kernel size	Number of filters	Output shape	Parameters
convolution 3D	$3 \times 3 \times 3$	4	(14, 62, 62, 4)	328
maxpool 3D	$1 \times 2 \times 2$		(14, 31, 31, 4)	
convolution 3D	$3 \times 3 \times 3$	8	(12, 29, 29, 8)	872
maxpool 3D	$2 \times 2 \times 2$		(6, 14, 14, 8)	
convolution 3D	$3 \times 3 \times 3$	32	(4, 12, 12, 32)	6944
fully connected			2048	37750784
fully connected			1024	2098176
fully connected			27	27675

person ID, and the stack of frames which have this ID are constructed. The frames will be concatenated to the sequence as long as the same person ID is detected in consecutive frames. Once the frame does not contain the specific person ID, a new stack will be created with the new person ID.

Since the videos are long ($\approx$ 1 hour), the gallery instances of individuals who are identified in the beginning of the video, gets out-dated. To avoid missing the similar instances of the same person, due to the change of camera angle or pose, we update the gallery entries after each 10 consecutive frames. In other words, the reference for each individual is replaced with the same person after 10 frames. This will ensure that instances in the gallery are up-to-date and increase the accuracy of the identification component.

7.1.2 Temporal Gesture Segmentation

The binary temporal segmentation component is using 3DCNN to classify the window of frames and assign “gesture” or “no gesture” to them. The details of this network can be found in Table 7.1.

We train this classifier using cross entropy loss with Stochastic Gradient Descent (SGD) optimizer with momentum 0.9 on the training set of the Jester [4] dataset with 27 labels. We start the training with a learning rate of 0.01 and after each 10 epochs, we reduce it by a factor of 10. We train the network with 60 epochs and the sliding window has 16 frames and a stride of one.

After training, we assign 26 labels to the “gesture” category and the remaining one to the “no gesture” category. We set the threshold for the probability of the frames to be labeled by “no gesture” class $\tau = 0.4$.

7.2 Representation Learning and Retrieval

Each feature extraction method explained in Chapter 6 has different setup and parameters. In the following we describe the training process and the details about the network architectures of the methods.

7.2.1 RGB-based Gesture Representation Learning

The RGB-based method has a two stream network architecture which takes the RGB frames of the videos and the extracted optical flow as the inputs. The RGB input to this module is the output of the preprocessing component, providing a saliency mask over the individuals in the frame and based on the task (isolated or continuous gesture recognition) uses the temporal gesture detection.

7.2.1.1 Data Preparation

To extract the optical flow, we use the optical flow estimation algorithm from the OpenCV library based on TV-L1 and map the values to the interval $[0, 255]$ and store these images with 2 channels. Since the extraction is not real-time, we pre-compute the optical flow of the collection videos and store them beforehand.

7.2.1.2 Network Information

The modified two stream I3D architecture used for RGB feature extraction is described in Table 7.2. Initially we use the I3D network pre-trained on kinetics-400, which is an action recognition dataset for each stream. The weights are publicly available by authors of I3D. The two streams are then fused by concatenation and fed to two 3D convolutional layer and a fully connected layer.

For the gesture recognition task, the last fully connected layer is mapped to the classification layer with 249 neurons for training the recognition method. We train the newly added layers with minimizing cross entropy loss using Adam optimizer with momentum 0.9 for 10 epochs using the Chalearn Iso gesture dataset and afterwards, fine-tune the entire network and adjust the weights by minimizing the cross entropy loss. We use the Adam optimizer with 0.9 momentum and we start the optimization by learning rate 10^{-2}, and whenever the loss did not change for 5 epochs, we reduced the learning rate by a factor of 10. We continued training the network for another 50 epochs.

For the Gesture similarity Learning task the same procedure of training is adapted, with the difference that the loss function used is the triplet loss $\mathcal{L}_{\text{triplet}}$ and we use triplet inputs for two stream networks. The anchor is chosen from one of the 249 classes of ChalearnIso gesture dataset and the positive sample video is selected from the same class, while the negative sample is selected from one of the other 248 classes. We use the SGD with 0.9 momentum and the learning rate is set to 10^{-2} at the beginning of the training and decreased by a factor of 10 on plateau for 150 epoch.

At the retrieval stage, the features of the entire collection are extracted and stored in the database. The selected query video's feature is extracted with the same method and is used to search and retrieve the most similar videos in the collection. The retrieval is based on the Euclidean distance and a ranked list with the lowest distance first is obtained to be shown as the result.

7.2.2 Pose-based Gesture Representation Learning

The pose-based feature extraction requires the RGB and the keypoints input. The implementation of the network is based on a third party implementation of RPAN[8]. The backbone which is used to extract the spatial convolutional cubes from RGB input is based on ResNetv2-50. The input to the backbone has the shape $(224, 224, 3)$ and produces the feature cubes with the size of $(16, 7, 7, 2048)$. The entire network is once trained with classification objective to identify the class labels of the gestures in ChaleranIso dataset with minimizing the cross entropy loss $\mathcal{L}_{\text{gesture}}$ with the weight decay set to 5×10^{-4} as the regularization. The optimization is done using Adam optimizer with the momentum$= 0.9$ and the learning rate is 10^{-3}.

For the similarity metric learning using the triplet loss, we used all the samples as the anchor iteratively and selected all the valid triplets. The loss is the average of the hard and semi-hard triplets and is minimized using SGD with learning rate of 10^{-5} with the decrease factor of 10 on plateau for 200 epochs.

7.2.3 Dimensionality Reduction in Representation Learning for Gesture Retrieval

The binary hash learning is done via the same network architecture described in 7.2.1.2 and the output feature vector with the size 2048 is used for learning to hash method. We follow the instructions of [188], we use $L = 256$ codewords in each M codebook and the number of codebooks depends on the number of bits of the output B and is a hyper parameter calculated as $B = Mlog_2L$. Minimization of the triplet loss was done by SGD optimization with 0.9 momentum and learning 10^{-5} and using an exponential decay on plateau. The triplets are selected based on the class label similarity, i.e. the positive and anchor samples are chosen from the same class, and the negative sample is chosen from a different class than the anchor.

[8] The pretrained network is from third-party implementation: `http://cmlab.csie.ntu.edu.tw/~agethen/resnet_v2.npy`

Table 7.2. The network architectural details and number of parameters used for feature extraction. The added layers are shared between the two streams (optical flow and RGB), and the rest are identical with only the difference in the channel of the input (three for RGB and two for optical flow)

	type	**Kernel size**	**Number of filters**	**outputsize**	**params**
Module 1	convolution 3D	$7 \times 7 \times 7$	64	$(20, 112, 112, 64)$	66 048
	max pool 3D	$1 \times 3 \times 3$		$(20, 56, 56, 64)$	0
Module 2	convolution 3D	$1 \times 1 \times 1$	64	$(20, 56, 56, 64)$	4 288
	convolution 3D	$3 \times 3 \times 3$	192	$(20, 56, 56, 192)$	332 352
	max pool 3D	$1 \times 3 \times 3$		$(20, 28, 28, 192)$	0
Module 3	inception (3a)			$(20, 28, 28, 256)$	386 668
	inception (3b)			$(20, 28, 480)$	1 225 196
	max pool 3D	$3 \times 3 \times 3$		$(20, 14, 14, 480)$	0
Module 4	inception (4a)			$(10, 14, 14, 512)$	738 384
	inception (4b)			$(10, 14, 14, 512)$	905,112
	inception (4c)			$(10, 14, 14, 512)$	1,104,328
	inception (4d)			$(10, 14, 14, 528)$	1,357,376
	inception (4e)			$(10, 14, 14, 832)$	1,800,165
	max pool	$2 \times 2 \times 2$		$(5, 7, 7, 832)$	0
		Fusion of the two streams			
Added layers	concatenation			$(10, 7, 7, 832)$	0
	max pool	$2 \times 1 \times 1$		$(5, 7, 7, 832)$	0
	convolution 3D	$1 \times 1 \times 1$	256	$(5, 7, 7, 256)$	213 248
	convolution 3D	$3 \times 3 \times 3$	128	$(2, 3, 3, 128)$	884 864
	fully connected	$3 \times 3 \times 3$		(2048)	19 662 848

Part IV

Experimental Analysis

Chapter 8

Gesture Recognition Experiments

The methods explained in the previous chapters, can be used for gesture recognition as well. This task usually refers to the prediction of a label for the gesture input. The performance of a method in this task can be measured by some defined metrics due to the presence of labels and ground truth. Therefore, we performed comprehensive experimental analysis to evaluate the usability and performance in one of the large-scale gesture recognition datasets.

We split the experiments of gesture recognition task into two sub tasks:

- Isolated gesture recognition, which does not require temporal detection of gestures, since there is only one gesture per video present and
- continuous gesture recognition, where the final prediction and results are dependent on the temporal detection method.

In the following we present the results on the datasets related to each sub-task. To make the comparison of the different proposed methods easier, we use the following acronyms for the methods introduced in this thesis:

- *ROFI3D*: The RGB and optical flow-based two stream network
- *RKLSTM*: The RGB and keypoint-based LSTM network

In occasions where a preprocessing step is added, suffix “P” is added to the method, i.e. *ROFI3D-P* is the RGB and optical flow-based two stream network with preprocessing and *ROFI3D* is the same approach without the preprocessing.

In this chapter, we present the results of our analysis in two different types of gesture videos. The isolated gesture recognition, Section 8.1 consists of the introduction to the evaluation metrics used in this task as well as the evaluation results and the comparison with the state af the art in isolated gesture videos. Furthermore, the results of the conducted evaluations on continuous gesture videos with an introduction of the metrics used for these experiments are presented in Section 8.2.

8.1 Isolated Gesture Classification and Recognition

8.1.1 Evaluation metrics

One of the common metrics in any computer vision analysis task is the accuracy which refers to how close the prediction (output) of a method is to the true label of the input data. This metric is most useful in analyzing the data for which the annotations such as class labels are present. This metric is usually is measured as a percentage and is calculated as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{8.1}$$

The Chalearn isolated gesture recognition dataset [3], uses accuracy metric with the name *recognition rate* as:

$$r = \frac{1}{n}\sum_{i=1}^{n} \delta(P_l(i), G_l(i)) \tag{8.2}$$

where n is the number of samples, P_l is the predicted label and G_l is the ground truth. The $\delta(x_1, x_2)$ function is defined as:

$$\delta(x_1, x_2) = \begin{cases} 1 & \text{for } x_1 = x_2 \\ 0 & \text{otherwise} \end{cases} \tag{8.3}$$

8.1.2 Evaluation Results

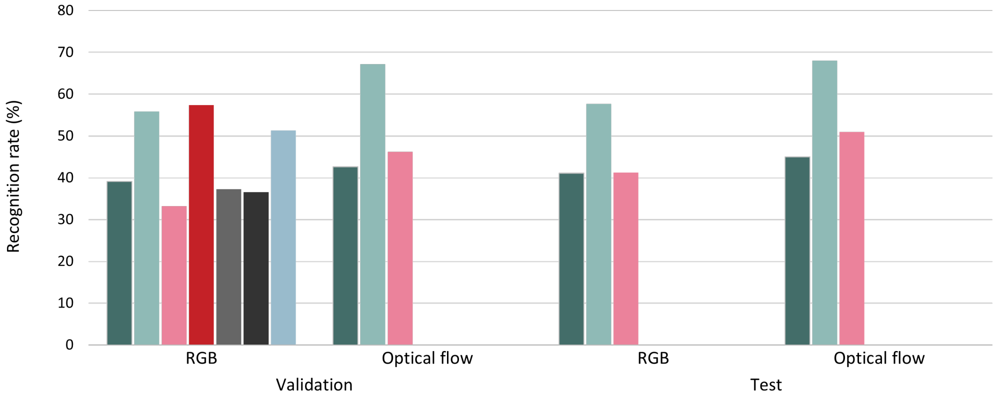
As the first experiment, we compare the results of the proposed recognition methods on the Chalearn Isolated gesture dataset [3]. Both of the methods and the variations are trained and validated on the Chalean Iso training and validation sets respectively and further tested on the test set. We have compared different setups of the proposed methods, with and without preprocessing (which in this case is the spatial segmentation) as well as against the state-of-the-art methods on the Chalearn Isolated gesture dataset. The results of the recognition rate on the validation and test set are shown in Table 8.1. As can be seen, our ROFI3D method with preprocessing has the highest recognition rate among our proposed methods and their variations. Additionally, with a closer look to the other state-of-the-art results and the input modalities used in the methods, we can see that our proposed method achieves the highest recognition rate among the methods independent of the depth modality.

In addition to the overall performance of our methods with multiple modalities, we broke down the ROFI3D performance to each individual modality, and compared the results with the best practices on the Chalearn Isolated dataset with the same modalities. The comparison of the results is illustrated in Figure 8.1. It is worth noting that most of the methods applied on the Chalearn gesture datasets use the provided depth data as well, which is not the case in our method. Except the method by Zhu et al. [190] using the RGB modality, ROFI3D-P outperforms all the other methods which reported their results per individual modalities on the Chalearn Isolated dataset.

As an experiment, we compared the recognition rate of proposed methods when using different numbers of frames when sampling the videos on the validation set of Chalearn Isolated gesture dataset, to find the best setup for the the training of the methods (see Figure 8.2). The results illustrated that with 40 frames per video, both variations of ROFI3D obtain the highest

Table 8.1. Comparison of the proposed methods in this thesis with the state-of-the-art on the Chalearn Isolated dataset [3] in gesture recognition task considering different modalities (RGB, Depth, Skeleton and Optical Flow).

Method	**Model**	**Modality**	**Fusion**	**Recognition rate(%)**	
				Validation	**Test**
ROFI3D-P (ours)	Two stream I3D	RGB + Optical Flow	Weight Fusion	**77.01**	**79.14**
ROFI3D (ours)	Two stream I3D	RGB + Optical Flow	Weight Fusion	69.72	71.5
RKLSTM-P (ours)	Pose-attention LSTM	RGB + Skeleton	-	64.38	70.45
RKLSTM (ours)	Pose-attention LSTM	RGB + Skeleton	-	61.26	67.66
AMRL [189]	VGG-16	Depth	score fusion	39.23	59.57
Zhu et. al. [190]	ConvLSTM, Res3D, Pyramid	RGB	score fusion	57.42	-
Li et. al. [191]	C3D	RGB	SVM	37.28	-
XDETVP [192]	Pyramidal C3D	RGB	Score Fusion	36.58	-
Zhang et. al. [122]	ConvLSTM, C3D	RGB	SVM	51.31	-
lostoy [193]	Masked C3D	RGB + Depth	score fusion	62.02	65.97
Duan et. al [194]	Two Stream CNN, C3D	RGB + Depth	Score Fusion	49.17	67.26
Zhu et. al. [190]	ConvLSTM, Res3D, Pyramid	RGB + Depth	score fusion	61.05	-
AMRL [195]	ConvLSTM, ResNet, C3D	RGB + Depth	score fusion	60.81	65.59
XDETVP [192]	Pyramidal C3D	RGB + Depth	Score Fusion	45.02	50.93
Li et. al. [191]	C3D	RGB + Depth	SVM	52.04	59.43
Baseline [3]	MFSK+BoVW	RGB + Depth	SVM	18.65	24.19
FOANET [196]	ResNet	RGB + Optical Flow	Sparse Fusion	71.41	75.41
Zhang et. al. [122]	ConvLSTM, C3D	RGB + Depth + Optical Flow	SVM	58.65	60.47
ASU [123]	ResC3D	RGB + Depth + Optical Flow	SVM	64.40	67.71
Li et. al. [191]	C3D	RGB + Depth + Optical Flow	SVM	54.50	60.93
Li et. al. [197]	ResC3D	RGB + Depth + Optical Flow	SVM	-	68.14
FOANET [196]	ResNet	RGB + Depth + Optical Flow	Sparse Fusion	**80.96**	**82.07**
Lin et. al. [198]	Skeleton LSTM, C3D	RGB + Depth + Skeleton	Weight Fusion	64.34	68.42
SYSU ISEE [199]	LSTM	RGB + Depth + Skeleton + Optical Flow	score fusion	59.70	67.02



	Validation		Test	
	RGB	Optical flow	RGB	Optical flow
■ ROFI3D	39.1	42.6	41.08	45
■ ROFI3D-P	55.9	67.2	57.7	68.04
■ Foanet	33.22	46.22	41.27	50.96
■ Zhu et. al.	57.42			
■ Li et. al.	37.28			
■ XDETVP	36.58			
■ Zhang et. al.	51.31			

Figure 8.1. The gesture recognition results obtained by each modality on validation and test set of Chalearn Isolated gesture dataset, compares with the methods which reported these values.

recognition rate. Due to the high computational cost of the RKLSTM for a higher number of frames per input video, we only considered 8 and 16 frames and the results showed the higher frame count would result in better accuracy on the validation set. While it is expected that with the higher number of frames the recognition performance improves, the added computational cost increases drastically for training and feature extraction at the inference stage.

Although the Chalearn Isolated gesture dataset is most relevant to our goal by having a large number of classes and samples, for the sake of comparison and completeness, we have also evaluated our proposed methods (this time only with preprocessing) on the Jester dataset [4]. The result of this experiment is shown in Table 8.2.

According to the methods which reported their results on the validation set of Jester dataset, both of our proposed methods have comparable results with state-of-the-art. However, the official reported results on the test set of this dataset [9] has obtained 97.37% accuracy. The lower accuracy by RKLSTM method could be the result of poor quality videos available in this dataset, specially with non-standard camera angles where sometimes parts of the torso and hands are not visible. Since the test labels are not publicly available, getting the results on the test set requires submission to the organizers, which has not been the top priority of our research to date.

[9] https://20bn.com/datasets/jester

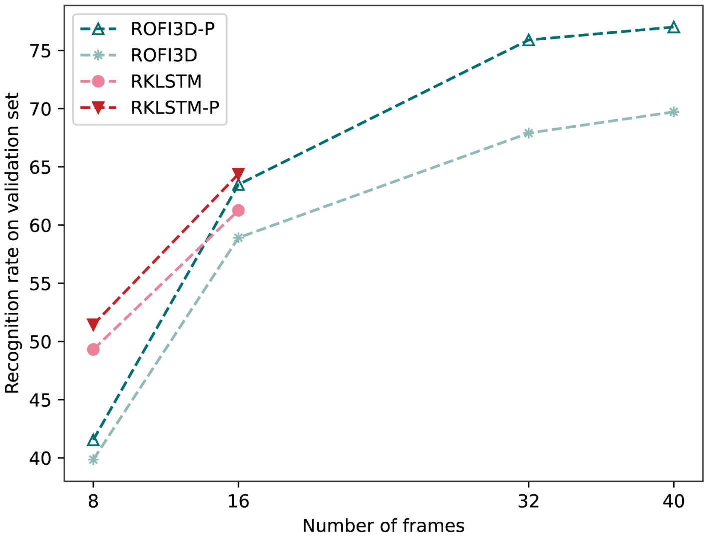


Figure 8.2. The effect of the number of frame samples per video in our proposed methods on the validation set of the Chalearn Iso gesture dataset.

Table 8.2. Comparison of the accuracy of our proposed methods with the state-of-the-art on the validation set of the Jester dataset.

Model	**Accuracy(%)**
C3D [79]	94.62%
Zhu et al. [122]	95.01%
Multiscale TRN [200]	94.78%
NUDT_PDL [201]	95.34%
Yu et al. [202]	95.77%
8-MFFs-3f1c [203]	**96.33%**
ROFI3D-P (ours)	**96.60%**
RKLSTM-P (ours)	96.38 %

8.2 Continuous Gesture Classification and Recognition

8.2.1 Evaluations Metrics

To evaluate the usability of our proposed methods and to compare them with the existing state-of-the-art, we introduce the evaluation metrics often used in the gesture recognition domain for the two subcategories of isolated and continuous gesture videos.

Jaccard Index: Unlike the isolated gesture datasets, where only one ground truth exists for each video, recognition of gestures in continuous gesture data requires a different metric for evaluation. The Chalearn continuous gesture dataset uses *Jaccard index* as an official metric to evaluate the methods. The mean Jaccard index $\overline{J_S}$ is calculated over all testing videos $S = \{s_1, ..., s_n\}$ as:

$$\overline{J_S} = \frac{1}{n}\sum_{j=1}^{n} J_{s_j} \tag{8.4}$$

where the Jaccard index J_s is is defined for each class label i as:

$$J_{s,i} = \frac{(G_{s,i} \cap P_{s,i})}{(G_{s,i} \cup P_{s,i})} \tag{8.5}$$

with $G_{s,i}$ being the ground truth and $P_{s,i}$ the prediction of sequence s of the label i.

Corrected Segmentation Rate (CSR): This metric is specifically designed to assess the usability of a temporal segmentation method in separating the gestures in one video. This metric is specifically designed to measure the correct time frame that the localization approach has predicted. The metric is based on the Intersection over Union (IoU) and is defined as following [204]:

$$E_{\mathrm{CSR}}(p, l, r) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} \psi\left(p_i, l_j, r\right)}{\max(n, m)} \tag{8.6}$$

where p is the predicted segmentation in start and end frame format, l is the ground truth and n and m are the number of segments in the prediction and in ground truth, respectively. $\psi(., ., r)$ is a function to measure the overlap of two segments using a predefined threshold r and is defined as:

$$\psi(a, b, r) = \begin{cases} 1, & \mathrm{IoU}(a, b) \geq r \\ 0, & \mathrm{IoU}(a, b) < r \end{cases} \tag{8.7}$$

where a and b are the segments predicted by the model and in the ground truth respectively. The IoU is defined as:

$$\mathrm{IoU}(a, b) = \frac{a \cap b}{a \cup b} = \frac{\max\left(0, \min\left(a_e, b_e\right) - \max\left(a_s, b_s\right)\right)}{\max\left(a_e, b_e\right) - \min\left(a_s, b_s\right)} \tag{8.8}$$

where a_s and b_s are the start and a_e and b_e are the end frames of the segments a and b. The threshold r decides how much misalignment is accepted between the prediction and the ground truth.

8.2.2 Evaluation Results

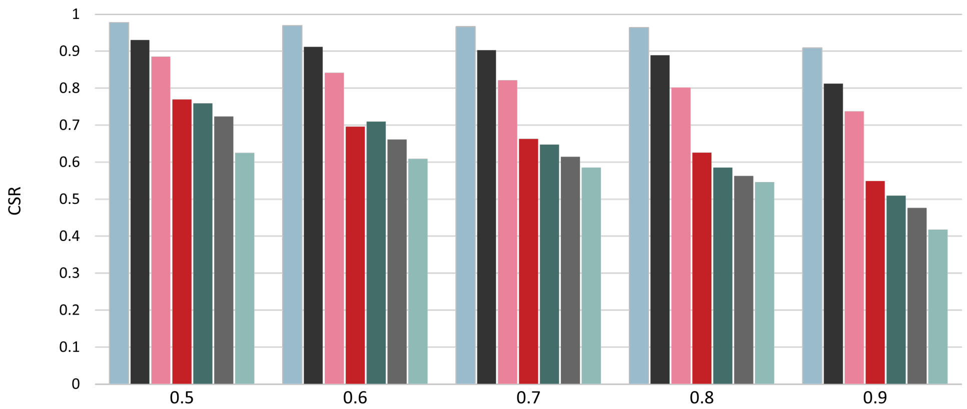
In addition to the recognition of the isolated gestures, we explored the performance of the two proposed methods of preprocessing to localize hand gestures temporally when they happen in one video. For this purpose, we used the Chalearn Continuous gesture dataset [3] for training and validation, and evaluated our method's performance on the test set of this dataset. The newly introduced CSR metric can measure how well the methods can predict the start and end of the gestures and temporally detect them, with less dependency on the performance of the recognition part of the process. This is important for methods such as ours, where the temporal localization is independent from the feature extraction and is performed in the preprocessing step. The results of our methods with QOM and binary classification method are presented in Table 8.3. The result of experiments in an isolated gesture dataset proved the positive effect of spatial segmentation in recognition tasks. Therefore, from here on we use ROFI3D-P and RKLSTM-P as the primary methods and will not consider the ROFI3D and RKLSTM without segmentation.

According to the results presented in Table 8.3, our methods using the temporal segmentation based on binary classification proved to be sub-optimal in practice. Our method based on QOM performs quite well in comparison to the binary classification method, and is able to segment the gestures temporally with higher confidence. However, in cases where the performer does not return to the "home position" (the hands' position in the first frame) between two adjacent gestures, this method fails to find the boundaries of the gestures. The current state-of-the-art for this dataset as measured by the Mean Jaccard Index (MJI) metric is achieved by FOANET [91], which uses the same fusion technique as in [196] isolated gesture recognition model. However, the current best method for temporal localization according to the CSR metric is obtained by Wan et. al. [204]. With focusing on the CSR measure, we can confidently relate the rather poor results of our method to the sub-optimal temporal segmentation module, which fails to correctly detect all boundaries.

We have additionally conducted another experiment with a focus on the different thresholds for the alignment of segments in CSR metric at different IoU. An overview of this experiment is shown in Figure 8.3. As expected, our QOM method performs better when the threshold is higher, and the performance for both methods is poor for lower thresholds. With a closer look, we can see the AMRL [205] is also using the QOM method on the depth modality, which is more sensitive on movements. However, the difference in performance at different IoU is imperceptible. The best performance in this comparison is achieved by the Bi-LSTM method from Wan et al. [204] is using a binary classification with the LSTM method on the skeletal data obtained from a Convolutional Pose Machine (CMP). Unfortunately, the CSR measure for the highest score in MJI is not available to make a more detailed comparison between this method and the attention mechanism with different modalities.

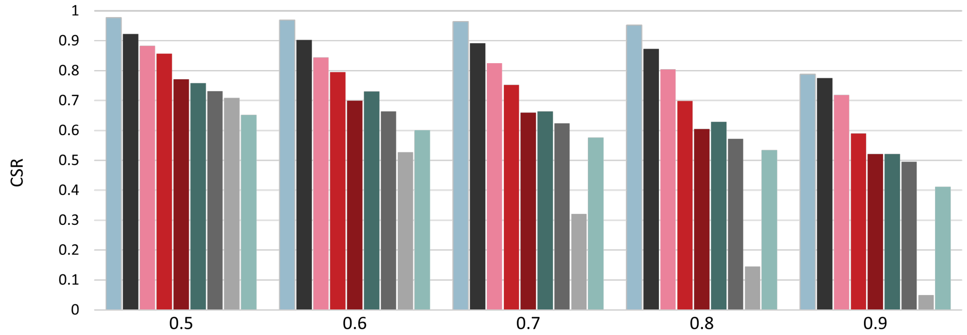
Table 8.3. Comparison of the proposed methods in this thesis with the state-of-the-art on the Chalearn Continuous dataset in gesture recognition task with different modalities (RGB, depth, skeleton and optical flow) using the MJI and CSR metrics with IoU = 0.7.

Method	**Model**	**Modality**	**Validation**		**Test**	
			MJI	**CSR**	**MJI**	**CSR**
ROFI3D-P-QOM (ours)	Two stream I3D	RGB + Optical Flow	**0.6108**	**0.6475**	**0.6104**	**0.6635**
RKLSTM-P-QOM (ours)	Pose-attention LSTM	RGB + Skeleton	0.5463	0.6475	0.5626	0.6635
ROFI3D-P-BC (ours)	Two stream I3D	RGB + Optical Flow	0.4621	0.5857	0.4537	0.5763
RKLSTM-P-BC (ours)	Pose-attention LSTM	RGB + Skeleton	0.4396	0.5857	0.4284	0.5763
AMRL [205]	VGG-16	Depth	0.2403	0.6636	0.2655	0.7520
Wan et. al. [204]	C3D, BiLSTM	Skeleton	0.6830	**0.9668**	0.7179	**0.9639**
TARDIS [206]	C3D	RGB	0.3430	-	0.3148	0.6603
Deepgesture [207]	ResNet, LSTM	RGB	0.3190	0.6159	0.3164	0.6241
PaFiFa [90]	C3D	RGB	0.3806	0.8213	0.3744	0.8254
ICR_NHCI [208]	C3D	RGB + Depth	0.5163	0.9034	0.6103	0.8917
Wang et. al. [195]	ConvLSTM, C3D	RGB + Depth	0.5958	0.6636	0.5950	0.7520
ICT_NHCI [209]	LSTM	RGB + Depth	0.2655	-	0.2869	0.3213
Baseline [3]	MFSK+BoVW	RGB + Depth	0.0918	-	0.1464	-
Hoang et. al. [210]	C3D, BiLSTM	RGB + Depth + Skeleton	-	-	0.5523	-
FOANET [91]	CNN, LSTM	RGB + Depth + Optical Flow	**0.7791**	-	**0.7740**	-
Zhu et. al. [211]	C3D,ConvLSTM	RGB + Depth + Optical Flow	0.5368	0.8553	0.7163	0.8776



	0.5	0.6	0.7	0.8	0.9
Wan et. al.	0.978	0.9699	0.9668	0.9638	0.9095
ICR_NHCI (2017)	0.931	0.9122	0.9034	0.889	0.813
PaFiFa	0.885	0.8421	0.8213	0.8024	0.7375
AMRL	0.77	0.696	0.663	0.626	0.549
ROFI3D-P-QOM	0.759	0.71	0.6475	0.586	0.51
Deepgesture	0.724	0.662	0.615	0.563	0.477
ROFI3D-P-BC	0.625	0.609	0.5857	0.546	0.418

(a) Validation



	0.5	0.6	0.7	0.8	0.9
Wan et. al.	0.9765	0.9686	0.9639	0.9522	0.7876
ICR_NHCI (2017)	0.923	0.9032	0.8917	0.873	0.775
PaFiFa	0.8833	0.8441	0.8254	0.8038	0.7187
AMRL	0.857	0.7954	0.752	0.699	0.59
TARDIS	0.771	0.7	0.66	0.605	0.521
ROFI3D-P-QOM	0.758	0.73	0.6635	0.629	0.521
Deepgesture	0.7313	0.664	0.6241	0.5722	0.4951
ICT_NHCI (2016)	0.709	0.527	0.321	0.145	0.049
ROFI3D-P-BC	0.652	0.601	0.5763	0.534	0.412

(b) Test

Figure 8.3. The comparison of temporal gesture localization methods according to the CSR metric at different IoU on the validation and test sets of the Chalearn Continuous gesture dataset.

Chapter 9

Gesture Retrieval Experiments

The gesture recognition evaluation performed in Chapter 8 showed that despite not using the depth modality, our methods have comparable performance with the state-of-the-art. In this chapter, we conduct experiments to evaluate the performance of the developed methods for *gesture retrieval* and analyze the results for large, real-world video collections. To have a reference, we perform both quantitative and qualitative analysis on labeled and not labeled datasets. In Section 9.1 we introduce the metrics we used to evaluate the methods in this chapter. Additionally, due to the absence of ground truth in the large dataset we use for our experiments, we will use statistical metrics to compare the results. Section 9.2 presents the experiments for evaluating the performance, both on labeled (Section 9.2.1) and unlabeled dataset (Section 9.2.2). At the end of this chapter in Section 9.3, we present our findings on the use of binary representations of the features for video retrieval.

9.1 Retrieval Evaluation Metrics

Throughout our experiments, we will use the metrics commonly used in the field of video retrieval. In case of user studies, we define scores, which are essentially the normalized rating given by the assessors.

Precision: One of the most commonly used metrics to measure the performance of a retrieval system is precision, which is a measure of result relevance.

Precision (P) is defined as the number of true positives T_p over the number of all the returned results:

$$P = \frac{T_p}{T_p + F_p} \tag{9.1}$$

where F_p is the false positives. Precision is usually calculated at k which indicates the proportion of relevant items on top-k results.

Beside precision, recall (R) is also very commonly used in the retrieval settings which is essentially defined as:

$$R = \frac{T_p}{T_p + F_n} \tag{9.2}$$

However for our experiments, we do not consider this metric as we cannot measure the return of false negative results. Higher precision is a sign of a more accurate retrieval method.

For some part of our experiments with the binary representations, to comply with the metrics most used in the field, we use mAP to measure the the quality of the performance at top-k results:

$$\text{mAP@k} = \frac{1}{k}\sum_{i}^{k} P@i \tag{9.3}$$

Discounted Cumulative Gain (DCG) In addition to precision@k, we compute the DCG as defined below:

$$dcg(s) = \sum_{i=1}^{N} \frac{2^{s_i} - 1}{log_2(i+1)} \tag{9.4}$$

where s is the list of scores corresponding to the retrieved results, using the aggregated scores as assigned by the assessors.

Fleiss' Kappa In addition to calculating the variance of the ratings, we computer the *Fleiss' Kappa* [212] (κ) to measure the inter-rater reliability which is defined as:

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \tag{9.5}$$

where $1 - \bar{P}_e$ is the degree of agreement above chance and $\bar{P} - \bar{P}_e$ is the degree of agreement between the assessors. This metric can give us an insight on the degree of agreement between the assessors and how the similarity is perceived by different people.

9.2 Gesture Retrieval Experiments and Results

According to the results of the recognition experiments with different modality, we have selected our methods using two modalities, due to their superior performance. Since there is no existing work on evaluating gesture retrieval systems to compare our results with, we have separated our evaluations into two main experiments:

- Gesture retrieval experiment using the Chalearn Isolated gesture dataset due to the presence of ground truth and largest number of classes, and the ability to perform more in depth quantitative experiments
- Gesture retrieval evaluations on an unlabeled subset of large-scale collection of real-world videos, NewsScape [6].

The subset of NewsScape dataset exhibits a large number of challenging scenes such as TV banners on the bottom of the scene, person-person and person-object occlusion and many camera movements and cuts. Since there is no reference available on this dataset which can be used as a ground truth to quantitatively measure the performance of our methods in real-world data, we perform a large user study and perform statistical as well as qualitative analysis on the results.

In the following, we explain the evaluation procedure, query formulation and results for each of the two evaluation setups.

9.2.1 Experiments on Chalearn Iso Gesture Dataset

To perform the evaluation, all of the methods extract the features of all the videos in the dataset during the *offline* part. These feature vectors are stored in a database and will be used in the *online* part of retrieval. The search for the gesture requires a query object which undergoes the same processing as the dataset videos, and its feature is used for the search. The Euclidean distance between the query's feature vector and the stored features in the database determines the results which according to this distance are closest and consequently most similar to the query. At the end, a ranked list of results are returned.

Query Video Selection For this experiment we used 9 query videos with the following characteristics for the evaluation:

- five videos from the test set of the Chalearn Iso dataset which are chosen according to the complexity and at the same time familiarity of the hand gestures for general public.
- four videos performed by two participants in a setting completely different from the dataset, to measure the ability of the method to generalize to out of the dataset samples. The participants were shown a video from the dataset and were asked to imitate the hand gesture. These four custom queries are shown in Figure 9.1.

Evaluation Setup Since the dataset is provided with the ground truth for the training and test videos, we use this opportunity to compute the retrieval metrics to measure the performance of the proposed methods on the retrieval task. The methods used for this part of the experiment are three variations of *ROFI3D* (*ROFI3D* trained with triplet loss without preprocessing, *ROFI3D-P* trained with triplet loss with preprocessing and *ROFI3D-CP* trained with contrastive loss with preprocessing) and *RKLSTM-P*. All methods are used to process all 9 query videos and 20 results per method from the training set are retrieved.

9.2.1.1 Evaluation Results

According to the labels of the retrieved results we can measure the precision@k with $k = 5, 10, 20$ to measure how many of the retrieved results shared the label with the query. The precision values for different methods are summarized in Table 9.1.

The results in Table 9.1 shows how many correct results per query have the identical label as the reference. It can be seen that the preprocessing has a visible positive effect on the retrieval of similar results and the methods trained to learn the similarity metric with triplet loss, can obtain higher precision in total. RKLSTM is clearly performing better than all

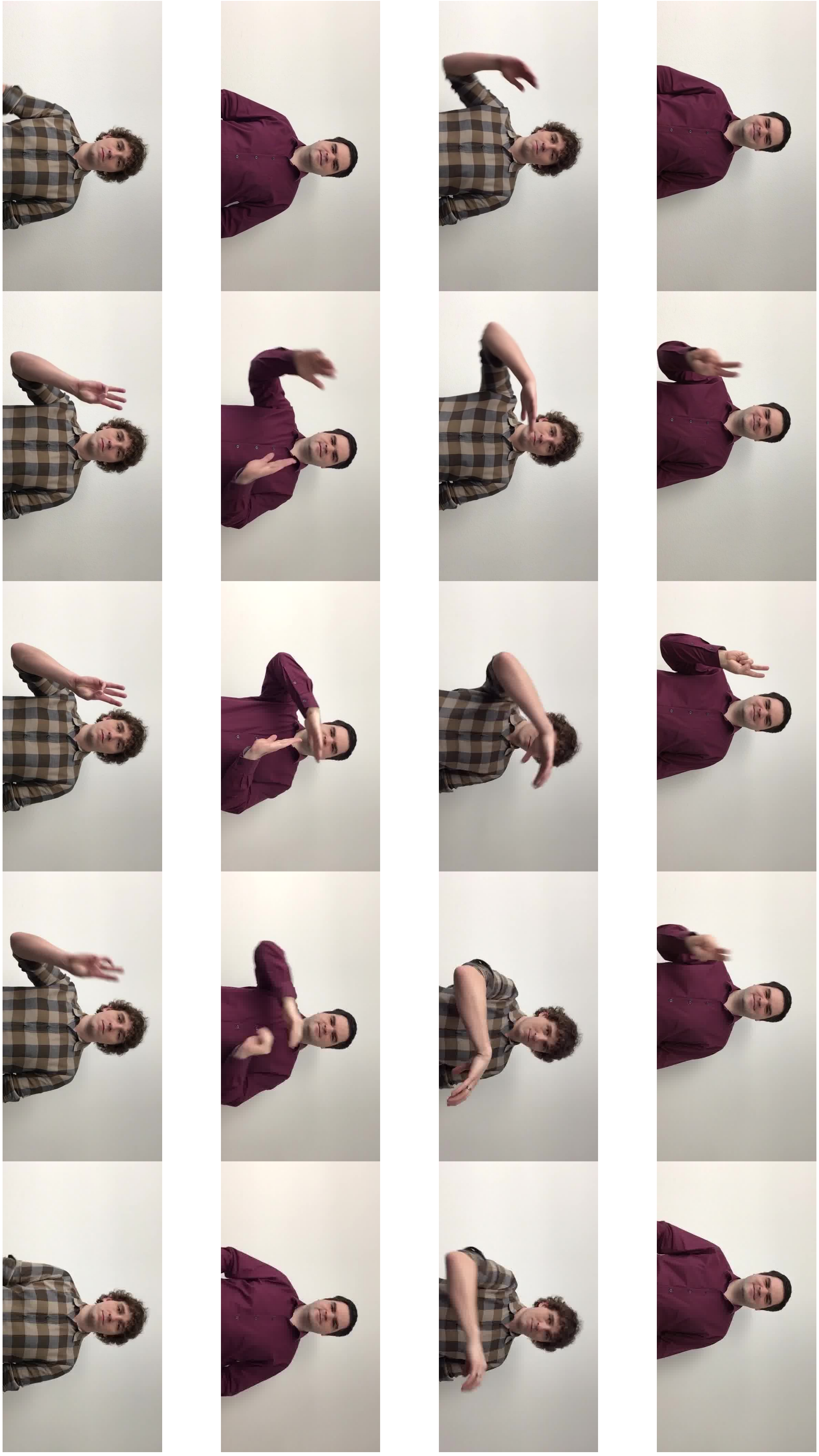


Figure 9.1. Sample frames from four custom query videos recorded by two participants used for evaluations.

Table 9.1. Maximum, mean and median precision at 5, 10 and 20 for the four proposed methods based on the ground truth from the dataset. The best result per row is printed in boldface.

model		**ROFI3D-P**	**ROFI3D**	**ROFI3D-CP**	**RKLSTM-P**
p@5	max	**0.8**	0.4	0.6	**0.8**
	mean	0.28	0.066	0.2	**0.22**
	median	**0.2**	0	**0.2**	**0.2**
p@10	max	**0.7**	0.2	0.3	0.6
	mean	0.177	0.044	0.144	**0.2**
	median	0.1	0	0.1	**0.2**
p@20	max	0.55	0.1	0.15	**0.6**
	mean	0.122	0.038	0.053	**0.194**
	median	0.05	0	0.05	**0.15**

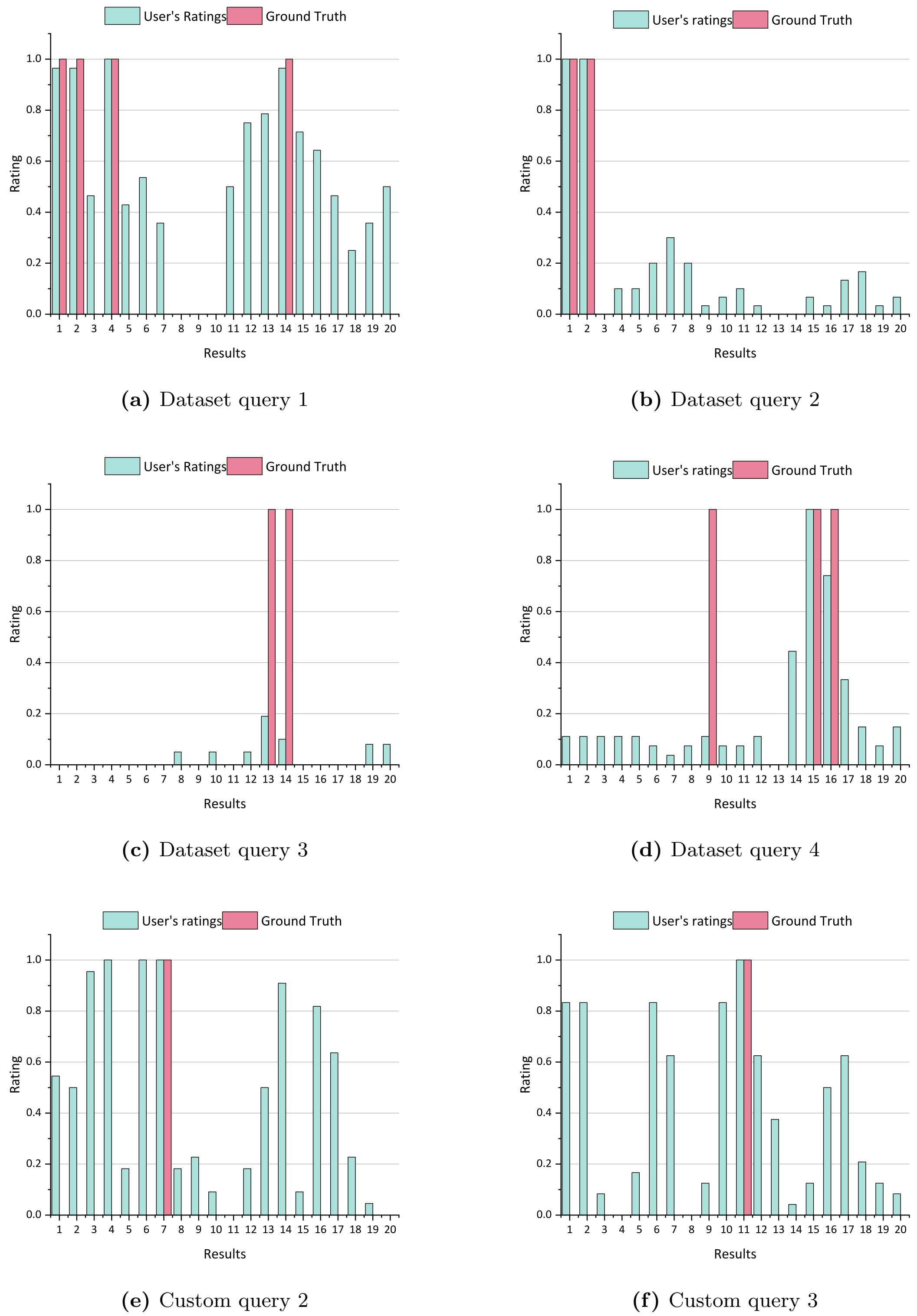
the variations of ROFI3D in retrieval and comparing with the result from the recognition experiments in the previous chapter, we can conclude that RKLSTM is more suitable for retrieval than recognition task.

When looking closely at the labels of the retrieved videos for the ROFI3D variations, we observed the repeated appearance of some labels other than the ground truth in the result list. Based on this observation we assume there is some sort of visual relationship between these videos and the query video.

In order to examine this hypothesis, we performed a user study with 10 individuals to assess the similarity of the results of the variations of ROFI3D. The participants were given a brief introduction in the evaluation survey and used their own personal devices to access the server. Each user was asked to rate all 9 queries each with 60 results for all three variations of ROFI3D. The assessors used four point Likert scale to assign a similarity score to the retrieved results (*'very good'* = 4, *'good'* = 3, *'ok'* = 2, *'bad'* = 1). The collected data was used to calculate DCG and precision, by defining the result as *relevant* when the normalized score was equal or greater than $\frac{2}{3}$. The result of this analysis is shown in Table 9.2.

It can be seen that while the ROFI3D-CP model has the highest single value for each measure according to the assessors' ratings, the ROFI3D-P model consistently outperforms the other two variations for all the mean and median measures. With a closer look, we can compare the results obtained from the data labels and the user's rating and compare them side by side. These observations are illustrated in Figure 9.2.

As can be seen in Figure 9.2, there are some result videos which despite their non-matching labels, have received high scores from the assessors and appeared similar to the query videos. This phenomenon is more frequent in custom queries, and can be the result of the difference in the appearance or quality of the videos and the gesture articulations by the performer. This observation is yet another proof of the ill-defined notion of visual similarity between hand gestures, specially when the user has no background in gesture's definitions and forms.



(a) Dataset query 1

(b) Dataset query 2

(c) Dataset query 3

(d) Dataset query 4

(e) Custom query 2

(f) Custom query 3

Figure 9.2. The side by side comparison of the assessors' score for four dataset and two custom queries compared with the ground truth labels from the dataset indicating the absolute relevance of the results.

Table 9.2. Maximum, mean and median dcg and precision at 5, 10 and 20 for the three presented network variants based on user study scores. The best result per row is printed in boldface.

model		**ROFI3D-P**	**ROFI3D**	**ROFI3D-CP**
dcg	max	4.25	2.69	**5.08**
	mean	**1.87**	1.25	1.62
	median	**2.38**	1.32	1.18
p@5	max	**0.8**	**0.8**	**0.8**
	mean	**0.33**	0.24	0.24
	median	**0.4**	0.2	0.2
p@10	max	0.5	0.4	**0.7**
	mean	**0.25**	0.16	0.15
	median	**0.3**	0.2	0.1
p@20	max	0.6	0.3	**0.7**
	mean	**0.23**	0.13	0.16
	median	**0.25**	0.1	0.15

Furthermore, it is valuable to see how well the proposed methods could retrieve similar results with the out-of-the-dataset query videos. This is specifically interesting when applying the methods for applications such as gesture annotations and search in large real-world collections, and using the *query by gesture* instead of long textual descriptions of the hand gestures, which is commonly used in linguistics. Table 9.3 summarizes the precision@k and DCG for different methods for different types of queries.

Table 9.3. Mean dcg and precision at 5, 10 and 20 for the three presented network variants with respect to if the queries were from the test set of [3] (*dataset*) or newly recorded (*custom*). The best result per row is printed in bold.

model		**ROFI3D-P**	**ROFI3D**	**ROFI3D-CP**	**RKLSTM-P**
dcg	dataset	1.85	1.75	**2.38**	-
	custom	**1.89**	0.61	0.67	-
p@5	dataset	**0.36**	**0.36**	**0.36**	0.3
	custom	0.3	0.1	0.1	**0.4**
p@10	dataset	0.22	**0.24**	**0.24**	**0.24**
	custom	**0.3**	0.08	0.05	**0.3**
p@20	dataset	0.21	0.16	**0.25**	**0.25**
	custom	0.25	0.09	0.06	**0.3**

It can be seen that there are large performance differences between the two query sources for the ROFI3D and ROFI3D-CP models, while the differences for the ROFI3D-P and RKLSTM-P model are considerably smaller. It can be also seen that the RKLSTM method outperforms all variations of ROFI3D in custom queries and is comparable with the other methods in dataset queries.

We also made a study on the user agreement on the scores given to query results. The κ values were between 0.09 to 0.38 with a mean of 0.26 which indicates a considerable level of disagreement in scoring the similarity of the results between the assessors. By taking a closer look at the variance between the scores given to each result for each query (illustrated in Figure 9.3) in comparison with the average of the scores for each query, we can see the level of disagreement differs between the results from different models, and in the method with the best performance according to our experiments (ROFI3D-P), this disagreement happens mostly for custom queries. This issue in most cases has a root in the difference between the direction or details of articulation of a gesture and the perception of similarity by assessors. Additionally, there are instances in the dataset where the subjects perform the gestures with fewer details of articulation than others, which to a novice assessor does not necessarily look the same.

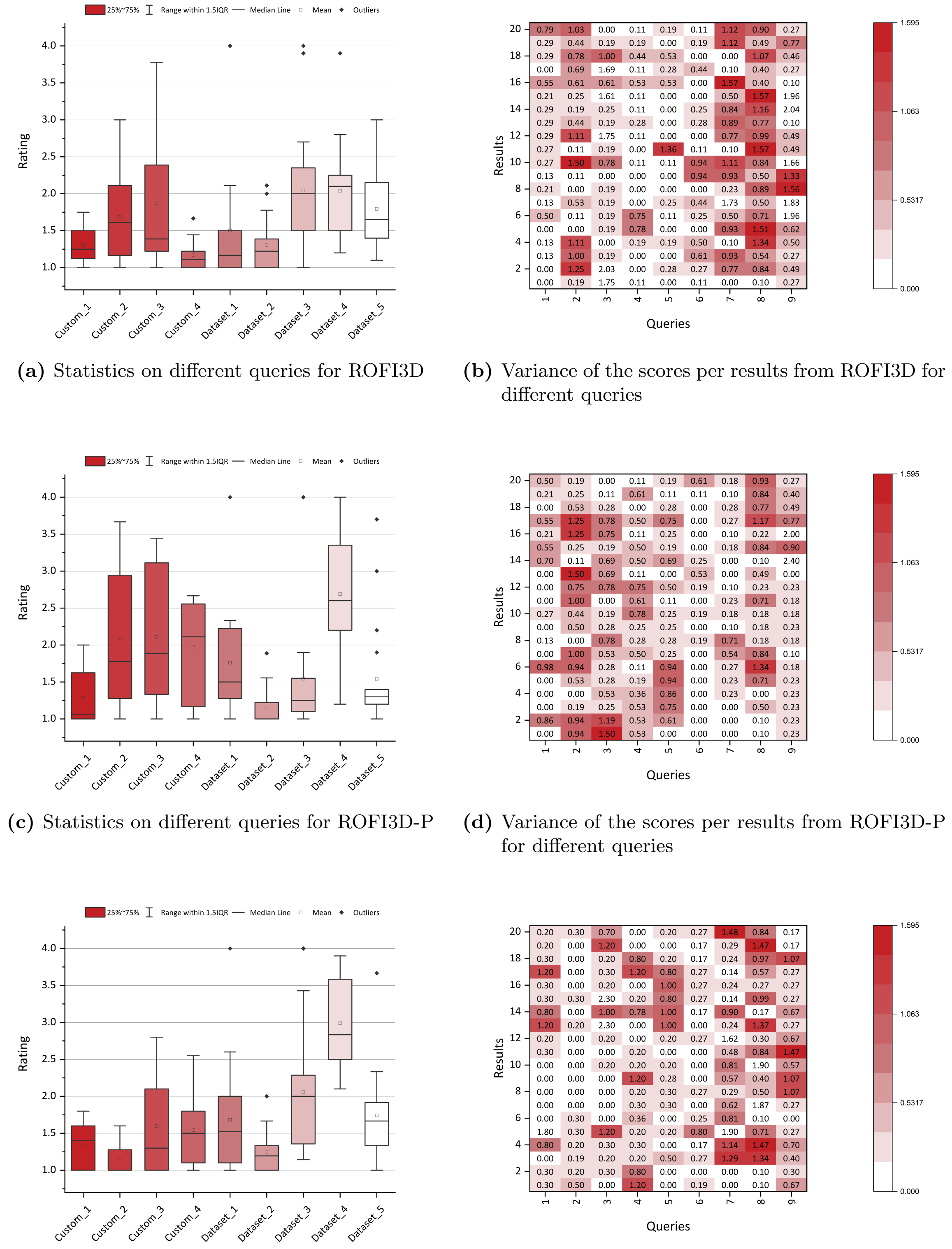
9.2.2 Experiment on NewsScape Dataset

The analysis of the results of our methods on a labeled dataset gave an insight on the usefulness of our methods in retrieving similar gestures in a controlled video collection. However, this analysis lacks an important aspect which we specifically aim at developing the RKLSTM method: the complex scenes with non-controlled environments and real-world challenges. The goal of this thesis is to expand the ability of computer vision tools to solve problems in the real-world, where not all the elements such as persons performing the gestures, the background, the style of gesticulation, camera position, etc. are pre-defined.

For this purpose, we used a subset of 259 videos from the NewsScape dataset, specifically from the *Ellen DeGeneres show*, which covers the entire year 2017 and is provided to us by Redhen lab. This is a specifically interesting dataset due to the various challenges present in the talk shows, where people use the hand and body gestures naturally. The dataset exhibits various sources of occlusions on the hands such as objects, persons and banners or subtitles on the scene. Due to the nature of the talk shows, usually there is more than one person in the scene, sometimes the audience is also shown.

According to the retrieval results obtained from the labeled dataset, we expect the RKLSTM method a better fit for this kind of data, specially because of the use of the skeleton keypoints instead of the optical flow. Since there are numerous scenes with camera and person movements, the ROFI3D method based on optical flow is not the best choice. Therefore, we only use RKLSTM for the evaluation, due to the large amount of processing time required for the entire subset of the dataset.

We ran the entire pipeline of RKLSTM with preprocessing steps including the spatial segmentation and cross angle person tracking to extract features of the 259 hours of videos in the *offline* phase. The preprocessing step produced 3 093 022 video clips based on the presence of persons in each camera shot and number of people present in the scene. The clips'



(a) Statistics on different queries for ROFI3D

(b) Variance of the scores per results from ROFI3D for different queries

(c) Statistics on different queries for ROFI3D-P

(d) Variance of the scores per results from ROFI3D-P for different queries

(e) Statistics on different queries for ROFI3D-CP

(f) Variance of the scores per results from ROFI3D-CP for different queries

Figure 9.3. The detailed analysis of variance and average score per query for three variations of ROFI3D.

lengths vary between fractions of a second and 74 seconds with the average of 1.45 seconds. The very short clips are the artifact of the mis-re-identification during the preprocessing step and the very long ones usually are the solo presence of the host talking without any camera movement. For the evaluations, we removed the ultra short video clips whose lengths were shorter than 2 seconds (60 frames). After this filtering, 1 501 037 video clips with an average length of 3.29 seconds were available.

To the best of the author's knowledge, to date, this is by far the largest gesture video collection and study made in the field of gesture retrieval. The extracted features from these clips were stored in the database and were used for the retrieval during an *online* phase.

The retrieval was performed by processing the query videos similarly as the dataset collection videos and their features were extracted by RKLSTM method. The feature extraction and preprocessing (including the keypoint extraction) for the entire collection took 1030 hours on our in-house servers and the query processing time for a 3 seconds video takes approximately 50 seconds.

Query Video Selection Similarly to the previous experiment on Chalearn Iso dataset, we select queries from different sources to assess the usability of the retrieval method. This time we select 10 diverse queries where:

- seven videos are taken from the dataset, with four of them representing co-speech gestures, two of them gestures such as clapping and waving and one query represented the sitting pose of the performer.
- three videos were performed by the author in the room setting which has a vast difference to the videos from the dataset. This is specifically used to measure the ability of the method to generalize to different samples. The three queries aimed to re-create co-speech gestures that occur while talking. These three custom queries are shown in Figure 9.4.

Evaluation Setup The *Ellen DeGeneres show* is not coming with any labeled data, therefore we performed a user study to analyze the quality of the results by the perceived similarity by different people. In addition to the assessors without linguistic background, we asked linguistic and cognitive science experts to participate in the survey separately. We ran the survey on two different servers for *linguists* and *non-linguists* assessors with a slight difference in formulating the similarity. More specifically, the experts were asked to rate the *formal* similarity of the gestures and the non-experts in linguistics were asked to rate the *visual* similarity between the gestures. Although both refer to the same concept, the vocabulary was adjusted to decrease the disparity of results due to misunderstanding.

The retrieval metrics were calculated by considering the result as *relevant* when the normalized score was equal or greater than $\frac{2}{3}$. In total, 76 people participated in our survey, of which 14 were linguists and 62 with non-linguistic background. The survey was designed to assign a random query based on the lowest number of results to each participant, and on average each query result collected 30 scores.

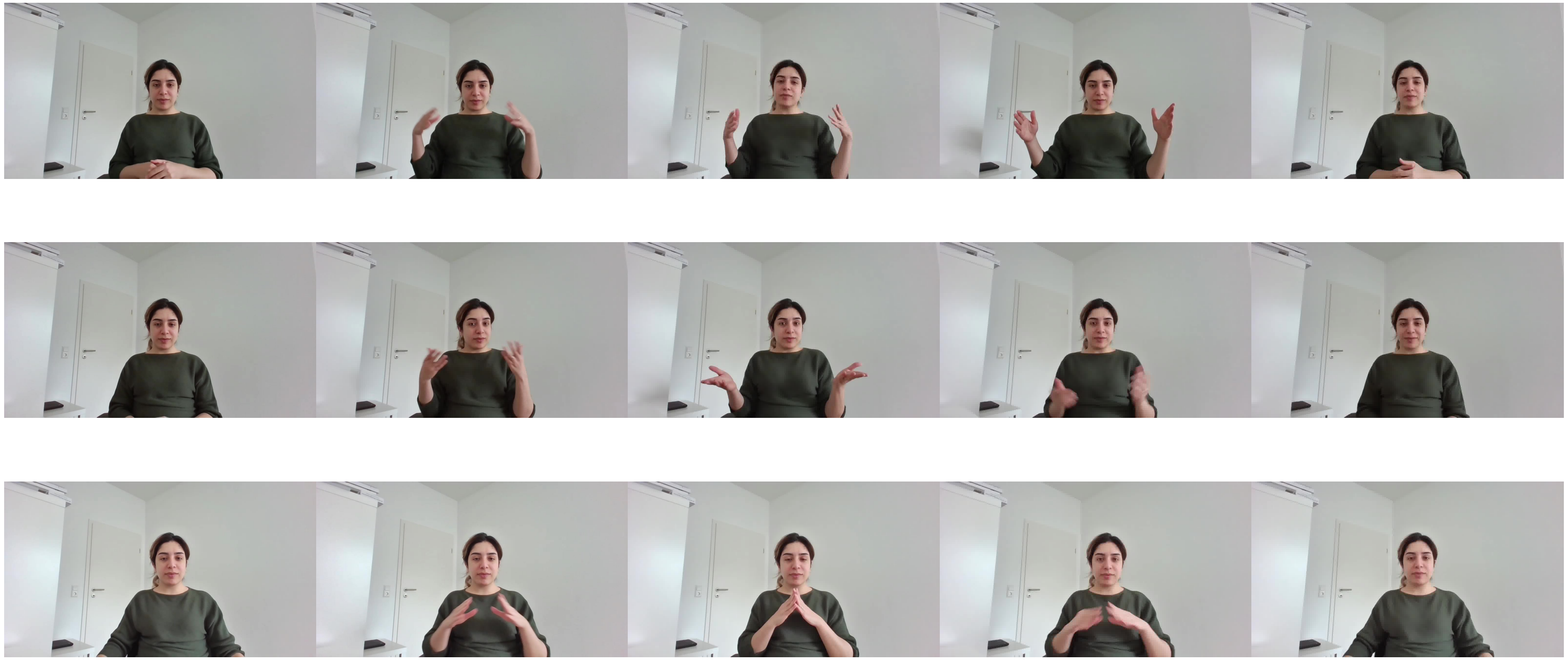
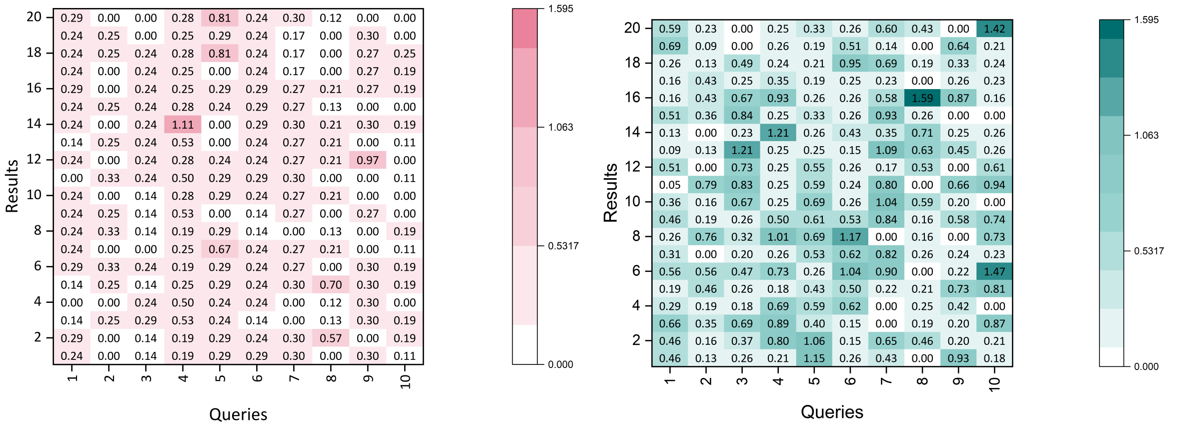


Figure 9.4. Sample frames from three custom query videos recorded by the author used for evaluations.



(a) Linguists

(b) Non-linguists

Figure 9.5. The heatmap showing the variance of the scores per query rated by linguists and non-linguists assessors (ratings are between 1-4).

9.2.2.1 Evaluation Results

Following the procedure of the previous user study, we made some statistical analysis to have an overview of the scores given to the results. Table 9.4 summarizes these analyses.

With a first glance at Table 9.4, we notice the high value for the DCG, especially from linguistics participants. This value indicates the better ranked results according to our method, which appeared more similar to the query according to the linguist assessors. Although not as impressive, the non-linguistic assessors also found the first results more similar.

The precision of the method according to the scores is higher than in the previous experiment. This is more notable for one of the queries where all the top five results were rated uniformly very similar by the assessors. Additionally a stable precision at different numbers of results (P@5, P@10 and P@20) can be due to a large number of videos in the database. To find a point where the precision starts to degrade and the diversity among the retrieved results increases, further experiments with a higher number of results (possibly 50 or 100) can be beneficial. However, with a user study setup, it is unlikely to expect assessors to rate 100 results of one query.

Comparing the statistics we gathered over the results, interestingly, we observed a high number of average inter-raters agreement over all the scores, with $\kappa = 0.67$. This value is higher for linguists, with $\kappa = 0.81$. We can observe this also in the variances of the scores assigned by different assessors to the results (see Figure 9.5). The heatmap indicates that the linguist assessors' scores have more consistency, and the variations between the scores are lower. On the other hand, from the heatmap it can be observed that the non-linguist assessors do not share the same consistency, which could be the result of a lack of deeper knowledge about certain co-speech gestures.

Similarly, we can observe the statistics of the scores per query, and take a closer look at the results (see Figure 9.6). According to the results, query 8 was rated the worst both according to the linguists and non-linguists. Figure 9.7 summarized the score per results for both linguistics and non-linguistics. Detailed observation of this kind for all the other queries can be found in Appendix A. This observation leads us to perform the analysis of precision with separated queries, to see how well the method can generalize. The results of this analysis can be seen in Table 9.5. We can clearly see that the custom queries have lower precision than the queries which are selected from the collection. We expected this to some extent due to the considerable difference between the custom query and the collection videos. Since the method also relies on the visual cues of the videos, change of colors and angle of the camera can be a reason for the difference of the performance.

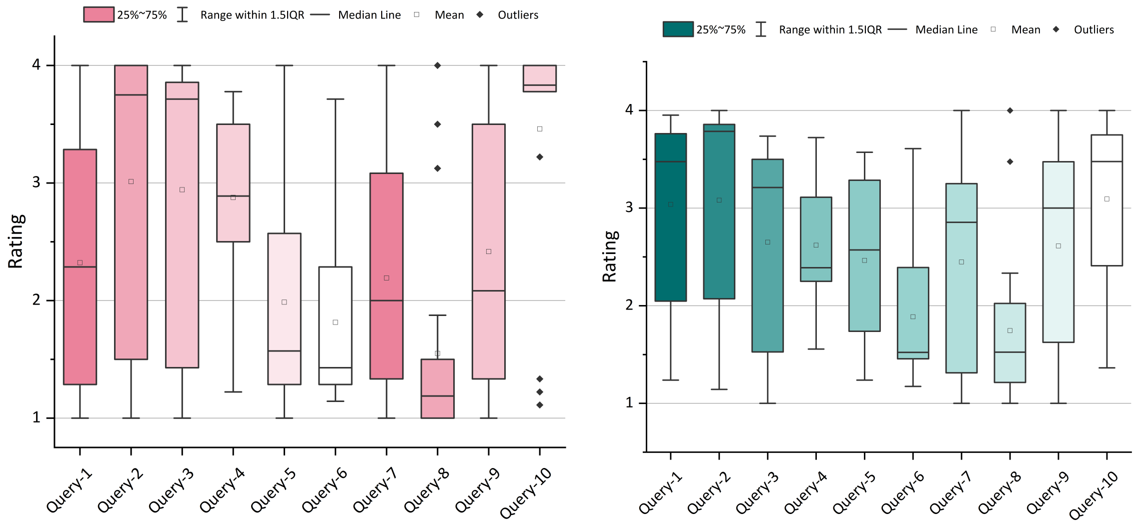
A more detailed statistical analysis on the queries and results in addition to sample frames from different queries than the ones used in the evaluations are presented in Appendix A. While studying these results we found a more uniform scoring pattern between linguists, specially on the co-speech gestures (queries 1, 4, 5, 6). However, this pattern was more visible in queries which represented more familiar gestures and actions (queries 2, 3, 10).

9.3 Dimensionality Reduction Experiments

For the sake of completeness, we have tested the ROFI3D method with binary representations and tested it with the state-of-the-arts in the video retrieval tasks when possible. To measure the performance of the method, we used mAP where higher value indicates a better retrieval model. Additionally, to have more insight into the efficiency of the binary representation for retrieval tasks, we measured the time and quality of retrieval on a gesture dataset.

Table 9.4. Maximum, mean and median dcg and precision at 5, 10 and 20 for RKLSTM based on user study split between linguists, non-linguists and total scores. The best result per column for each category is printed in boldface.

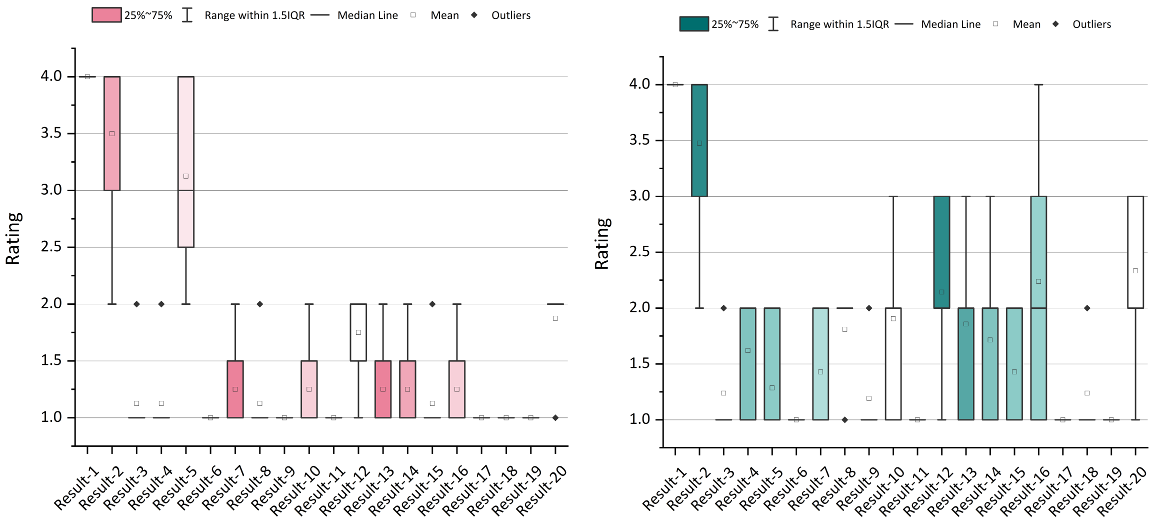
	Participants	**P@5**	**P@10**	**P@20**	**dcg**
mean	overal	0.38	0.4	0.395	5.85
	Linguists	0.4	0.41	0.41	**6.55**
	Non linguists	**0.44**	**0.47**	**0.44**	5.86
median	overal	0.4	0.35	0.375	5.64
	Linguists	0.4	0.35	0.4	5.58
	Non linguists	0.4	**0.6**	**0.5**	**5.70**
max	overal	**1**	0.9	0.8	8.71
	Linguists	**1**	**1**	**0.85**	**12.79**
	Non linguists	0.8	0.8	0.7	8.33



(a) Linguist

(b) Non-linguist

Figure 9.6. The descriptive statistical analysis of the mean score given by linguist and non-linguist assessors for all the queries.



(a) Linguist

(b) Non-linguist

Figure 9.7. The detail scores given to custom query 8 separately by linguist and non-linguist assessors. The linguists consistently rated this query results with low scores, while there is disparity of the ratings between non-linguist assessors.

Since there are no hashing results reported on gesture specific datasets, we used the human activity dataset, JHMDB [213] for measuring the performance of the short binary codes generated by ROFI3D-DTQ and compare them with state-of-the-art. In order to be comparable

Table 9.5. Mean precision at 5, 10 and 20 according to the overall scores with respect to if the queries were from the video collection of NewsScape (*dataset*) or newly recorded (*custom*).

	dataset			custom		
	P@5	**P@10**	**P@20**	**P@5**	**P@10**	**P@20**
Overall	0.4	0.5	0.5	0.35	0.25	0.23

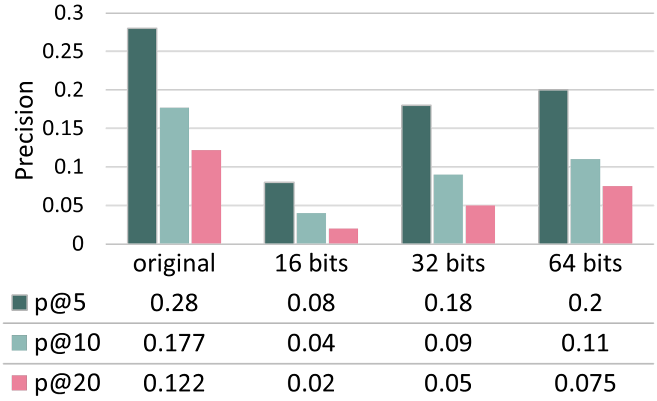
Table 9.6. Comparison of different video hashing methods on JHMDB datasets according to mAP% with different number of bits.

Model	**16 bits**	**32 bits**	**64 bits**
DSH [216]	5.28	6.37	6.76
ITQ-CNN [214]	13.25	14.57	15.10
DVH-CNN [168]	35.19	37.43	37.95
SPDTH [169]	38.66	43.88	46.47
HetConv-MK-BiDLSTM [215]	41.02	45.56	48.27
ROFI3D-DTQ	39.08	41.34	46.21

with the reported results on this dataset, we use the 16, 32 and 64 code bit lengths. Table 9.6 shows the performance of retrieval by percentage of mAP for different code lengths. The methods selected for the comparison use CNN features and different dimensionality reduction approaches; DVH [168] and ITQ [214] use PCA to generate different code length and SPDTH [169] produces hash codes by preserving the temporal dependency of the frames. It is natural to expect the methods with the ability of encoding the temporal dimension to have higher retrieval accuracy. The best results are obtained by HetConv-MK-BiDLSTM [215] and our two stream method based on 3D kernel convolutions is the runner up in 16, 32 and 64 bit codes.

One of the main reasons we have included the dimensionality reduction method in this thesis, is to explore the possibility of enhancing the retrieval speed by preserving the quality of the results on the Chalearn Isolated gesture dataset. We have replicated the same retrieval experiment as in Section 9.2.1, this time with binary codes and different lengths. The method (ROFI3D-DTQ) was trained on the Chalearn dataset and the queries were selected from the test set. We used the same queries as before and used the label of the results to measure the mean precision at $k = 5, 10, 20$ for all the queries. The results for different code lengths are displayed in Figure 9.8. The code length selection in this experiment follows the idea of compactness. According to [217], the length of a *compact* code is not largely greater than $\log_2(L)$ where L is the number of classes in the dataset. Therefore our binary codes cannot be smaller than 5 bits, and 16, 32 and 64 are relatively compact according to this number.

By comparing the retrieval performance of the original feature with the binary codes with different lengths, it can be seen that the information loss with the short binary codes are



	original	16 bits	32 bits	64 bits
■ p@5	0.28	0.08	0.18	0.2
■ p@10	0.177	0.04	0.09	0.11
■ p@20	0.122	0.02	0.05	0.075

Figure 9.8. The precision at 5, 10 and 20 for different code length in comparison to the original length of the vector on Chalearn Isolated gesture dataset.

huge especially with the 16 bit codes. This can be explained by the rich temporal information present in the original embedding which is essential for gesture recognition and retrieval. We believe that during the quantization of the gestures, the information loss increases, specifically due to the large number of classes available in the dataset, and the high intra-class dependency which is an inherent characteristic of the gesture dataset. Therefore, encoding the discriminative spatio temporal information of video clips requires longer code lengths. Additionally, looking at the rate at which the precision at k for different code lengths is improved, we do not expect 128 bit code to surpass the results from the original length code.

When comparing the speed of the retrieval, we observed the large amount of the retrieval time, both on the online and offline part, is taken by the preprocessing and the feature extraction prior to the search. To be more specific, encoding the entire Chalearn Isolated training set without the quantization method took roughly 30 hours and the retrieval of 20 results, iėthe comparison of features and ranking the results, were done in less than 10 seconds. Using the quantization method, the feature extraction of the collection remained almost unchanged, however, the retrieval time needed to fetch 20 results for a query dropped to two seconds for code length of 32. According to these data, we believe that the decrease of retrieval accuracy does not justify the speed improvement.

Chapter 10

Discussion

The results of this thesis can open different lines of discussions on the usefulness of methods based on different modalities of the data, and the importance of paying more attention to the creation of datasets to further enhance the computer vision methods and tools in the areas of gesture recognition and retrieval. This thesis opens a door to unexplored fields of gesture search which have very domain specific challenges and need to be addressed with specific methods. In the following, we discuss the results from the evaluations and observations we made throughout this thesis.

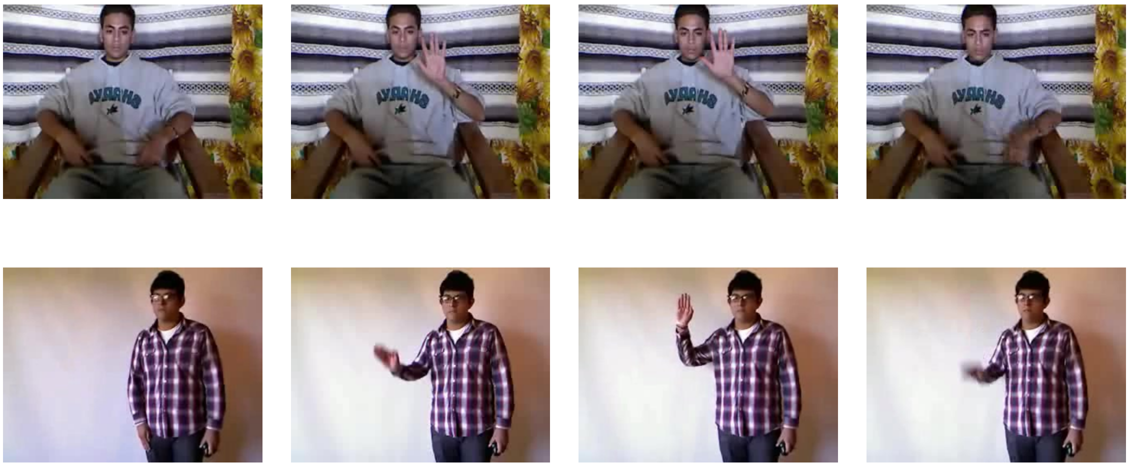


Figure 10.1. Sample frames from two videos in the evaluation from Chalearn Iso dataset. The two videos have dissimilar label (top: RefereeWrestlingSignals2 and bottom: ChineseNumbers/wu) despite their visual similarity.

10.1 On the Perception of Gesture Similarity

One of the most observed phenomena during the evaluation of the retrieval result by the assessors on both labeled and real-world dataset are the unclear boundaries between dissimilar and similar gestures. Our experiment with the labeled dataset showed that there are instances where the assessors at large agree that the query video is *similar* to the results, while the label of the video indicates differently. Figure 10.1 is an example of such a query and results

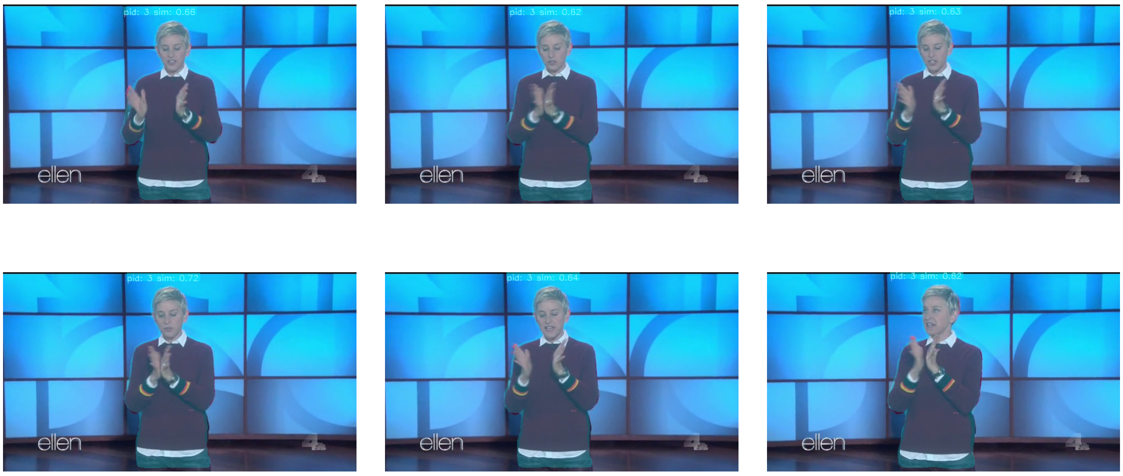


Figure 10.2. Sample frames from the clapping query from the NewsScape dataset with the different function as of the intended.

from the Chalearn Iso dataset. In this instance, the result does not share the same label as the query, however, from a trajectory point of view, they look very similar. Taking a look at the Figure10.1 and exploring the samples from the dataset, we noticed the difference between the two classes containing these videos lies within the flexed fingers, which are not always demonstrated perfectly either.

Generally, flipped trajectories of gestures are considered to have different meaning and therefore are assessed as dissimilar. However, occasionally such gestures have the same label in the dataset. Moreover, the gesture articulation varies from person to person and sometimes these differences make similar gestures look dissimilar, even though they are from the same category.

In linguistics, where the hand gestures are defined with different components such as form and function, there is not one single notion of similarity which is generally applicable. For example, one of the selected hand gestures in the evaluation is showing the host clapping with only the palms of the hand (Figure 10.2). The retrieved results to this query video are very similar to the action of clapping which involves bringing up the hands and the back and forth trajectory of arms in the horizontal axis. However when listening to the speech of the host, she is explicitly referring to this type of clapping (with palms only) as not fulfilling the objective of clapping which is making noise. Therefore, linguistically, the retrieved results which are the correct form of clapping do not have the similar *function*, therefore are not entirely “similar”. Since our proposed method in this thesis is entirely independent of speech, the results cannot reflect such functional similarity.

The results obtained from the evaluations showed that the notion of similarity is an ill-defined concept, especially in the hand gesture context when a clear definition can not be given to describe that motion. There have been detailed analyses on the notion of similarity in [49], which summarizes how ill-defined this notion is on images. This problem gets more severe when talking about gestures, where the spoken words, the body posture and the facial expressions can impact how similar one gesture would seem to another. Therefore, when using off-the-shelf methods for feature extraction, it is important to take into account the fact that the results could potentially not be as perfect as shown on curated and controlled datasets.

10.2 On Different Modalities of Data for Gesture Retrieval

It is worth mentioning that the feature extraction module of both of our proposed methods are using only RGB frames as input, and are entirely independent of the depth modality which is available in the training dataset, and from which most of the methods available in this field benefit from. According to the results from the state-of-the-art methods on the gesture recognition datasets, the best results are obtained with the depth, optical flow and RGB modalities together. The depth data is the projection of the 3D space into 2D space and can help in identifying the gestures, specially those which have movement in the axis perpendicular to the camera. Our experiments show that despite the absence of this modality, our methods obtain satisfactory results, especially on the real-world video data collection. We believe that part of this performance is due to the preprocessing methods used to remove the background clutter and deal with the occlusion and multi person scenarios. Additionally, using the skeletal data extracted from human instances in the scenes improves the retrieval results specially on the NewsScape dataset with free-form gestures.

The absence of the depth modality in TV footage as the biggest source of gesture videos for multi-modal studies in different fields, can be a motivation for the computer vision community to develop larger datasets with sufficient annotations for the training of neural networks.

10.3 On Dimensionality Reduction and Performance

Our limited experiments on using binary representations for retrieval showed that despite the semi-comparable results for action video data, the short binary codes do not contain sufficient information of the gesture instances. Additionally, most of the computational cost of the retrieval in our method is the preprocessing and the feature extraction, which remains unchanged after the quantization of features. Therefore, unless the quantization method can have comparable results to the non-binary feature-based retrieval, we do not a gain from quantization of the features.

10.4 On Dataset Imbalance

The Chalearn gesture dataset, which is currently the largest annotated hand gesture video collection and is used for the training of our models, is greatly imbalanced regarding the number of samples per class. Statistical analysis on the Chalearn Iso dataset (Figure 10.3) shows an imbalance in number of samples per class which ultimately can bias the feature extractor to learn more of a certain class [218]. Although not used in this thesis, data augmentation can be used to balance the number of samples per class to alleviate this problem [219].

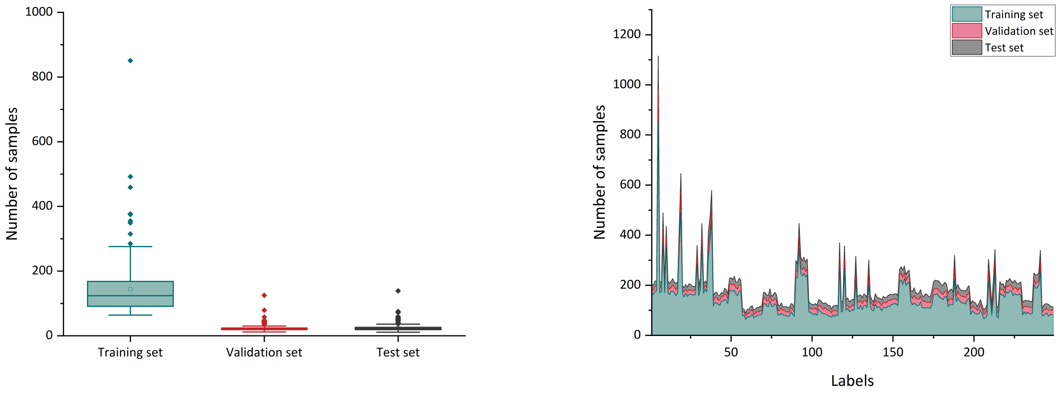


Figure 10.3. The statistics on the number of samples per class in each training, validation and test set of Chalearn Iso gesture dataset.

10.5 On Preprocessing and Network Architecture

We have used different preprocessing steps on input data prior to feeding them to the feature extraction module, to specifically overcome the challenges existing in the real-world data.

10.5.1 Effect of Segmentation

When taking a look at the results presented in Tables 9.2 and 9.3, we clearly see the methods which are using the preprocessing have superior performance in the retrieval of hand gestures. This effect is especially more visible for the custom queries which causes the method not to consider the background of the individual performing the gesture and to generalize to the real-world samples. This is more important in the real-world video collection, where the background clutter and the occlusion would degrade the result of recognition and retrieval.

10.5.2 Effect of Temporal Localization of Gestures

Our experiments suggest that our proposed localization methods for the hand gestures are not sufficiently reliable to be used for real-world data. Especially due to the fluid trajectories of the hand during the conversation, they do not always return to the resting position which is essential for our best performed temporal localization method. This topic requires more in-depth analysis of the end-to-end architectures which can locate the hand gestures during the feature extraction and do not perform this in a preprocessing step.

10.5.3 Effect of Pretraining and Network Architecture

Both of our models, as explained in Chapter 6, are trained on the Chalearn Iso gesture dataset. However, we have used the pretrained model of I3D on kinetics-400 which is an action recognition dataset prior to adding new layers. According to our experiments, the pretraining for this architecture is essential due to the use of 3D convolutions, which are generally difficult to train. However, the RKLSTM network is only trained on gesture dataset. We believe that using a pretraining for this method would also benefit the feature extraction. However, we skipped this comparison due to the shortage of time and would recommend to include pretraining of the network on larger activity datasets.

10.6 Impact on Linguistic Studies

The results gathered from linguists who participated in the evaluation survey of RKLSTM have shown that the retrieved videos meet the formal similarity expectations of the experts in the field. However, our method in its current state cannot be extended to include the function of the gestures due to the lack of speech modality input. Despite the lack of function similarity retrieval in our proposed method, our current pipeline can be used as a preliminary gesture suggestion for analysis to reduce the manual annotation effort. Additionally, with some modifications, we can group the type of simple hand movements which are attractive for linguist studies, such as *arms up* or *hands wide open* to narrow down the search for specific gestures.

RKLSTM currently does not support fingers keypoints and the search for hand gestures with fine-grained movements, such as *air quotes* of the fingers is not very successful. Further studies to measure the usability of the finger keypoints in retrieving such gestures are required.

10.7 Beyond Gesture Similarity Retrieval

We have observed that our proposed method could detect the similarity for queries representing some actions, such as dancing, hugging, and even sitting. We have explored this functionality and have listed some of these results in the Appendix A. These results also include static poses and actions such as the way someone sits or stands. Currently, most of the video retrieval systems work with the textual queries or image input. Extending the query type of these systems by video and possibly performing the action could potentially be useful in video retrieval systems. Such an integration would allow a deeper analysis of the performance of our method in more diverse situations.

Part V

Conclusion and Future Work

Chapter 11

Conclusion

This thesis opens a new door in gesture analysis in computer vision by introducing the hand gesture retrieval for communication gestures. The underlying motivation of this work was the absence of a search medium specifically designed to overcome the challenges existing in hand gestures. Although different works studied the recognition of the hand gestures in curated, controlled environments, the real-world media collection has far more complications which cannot be addressed with their methods. After reviewing the literature in the field, we proposed a preprocessing followed by a feature extraction pipeline which to a large extent can meet the expectations of a retrieval system.

The different components proposed in this thesis are domain specific approaches to overcome the challenges especially existing in real-world scenarios of human interactions. Our preprocessing module consists of a spatial segmentation module followed by a person identification and tracking module. These two steps jointly detect each human even when parts of the body are occluded and isolate them from the environment to reduce the impact of background clutter. Additionally, we have explored two different solutions for the challenging topic of gesture temporal localization.

Next, as the primary goal of this thesis, we have proposed a similarity learning approach to encode gesture instances coming from the preprocessing module. As the source of the largest chunk of video data is TV footage, the methods presented in this thesis are independent of depth modality and sensor signals. We developed two different methods: One of them uses two streams of inputs with RGB and optical flow data and extracts the spatio-temporal features of gesture videos using the 3D convolutional network. The second one uses LSTM units to model the temporal dependencies between the gesture video frames and uses a skeletal attention mechanism to extract features of the RGB videos. As a side experiment, we tried the binary representation learning to reduce the dimensionality of spatio-temporal features extracted from the two streams method.

We have conducted comprehensive experiments, evaluations and ablation studies on the usability of our proposed methods for gesture retrieval and recognition. The results obtained from the evaluation of our two streams method on the annotated gesture recognition datasets showed that the absence of depth modality in our pipeline hinders our method to achieve state-of-the-art accuracy. However, the difference is negligible and our method has the highest accuracy in comparison with the state-of-the-art methods not using depth modalities.

Due to the newness of the gesture retrieval task and absence of the state-of-the-art, we evaluated our method through a user study survey with participants partly from the linguistics field. The performance of our pose-based LSTM method, according to our assessors, is very good on the majority of the presented queries. However, custom ('out of the dataset') queries do not have similarly good results. Additionally, the user study showed that the notion of similarity between hand gestures is ill-defined and the disparity among the ratings by the assessors is due to the different perception each person has. Our experiments on binary representations on the spatio-temporal features showed that the gain in speed of retrieval comes at the cost of losing performance which, at the current state, does not justify the use of it.

The contributions of this thesis are initial steps to extend the computer vision tools in the field of gesture retrieval for communication gestures. Currently the methods existing in the gesture recognition domain are dependent on curated datasets, which does not exhibit the real-world challenges. The proposed methods help researchers in generating large scale annotated datasets with real-world challenges.

Chapter 12

Applications and Future Work

This final chapter will introduce the applications of gesture recognition and retrieval (Section 12.1) and suggest possible paths to follow, in order to improve the existing work and overcome the existing challenges in the field (Section 12.2).

12.1 Applications

Hand gesture recognition and retrieval have a great potential not only in modern applications but also in helping the existing computer vision tasks to solve real-world problems.

12.1.1 Linguistics

The focus of this thesis was to propose methods which are suitable for *in the wild* gestures and which can reliably detect the visual and formal similarity among them. However, the function of the gestures contain broader topics and can have different categories depending on the research interest of the linguistic community. One of the possible applications of this thesis is the implementation of the proposed method in the backend of the gesture annotation tools used by linguistic researchers. These tools often take the list of the media contents manually and display it in order to be annotated. The integration of the methods developed in this thesis within such a tool, would result in a sophisticated gesture annotation framework for linguistic studies. More specifically, the media contents selected for annotation could be pre-filtered to increase the amount of positive samples and increase the speed of annotation. This application can be a first step towards the generation of a labeled collection of gestures used to further improve the state-of-the-art for *in the wild* gesture recognition and retrieval.

12.1.2 Human-Machine Interactions

One of the traditional applications of hand gesture recognition is in the field of human-machine interaction. There is a lot of research going on in the field of gestural commands for controlling robots [220], robotic assistance interactions [221] and entertainment [222]. Over the last one and half years with the spread of *Covid-19* and the ongoing pandemic, the hygiene of touch screen devices available in public places has become a topic of concern and touchless devices have gained more attention than ever. Hand gestures are considered a natural interface to interact with machines and the intuitiveness of using the hands makes the gesture controls a popular choice for the general public. The RGB based gesture recognition systems with the ability to recognize and track a person in a multi-person environment, can be a feasible and low cost solution for a large number of businesses.

12.1.3 Sign Language Interpreters

The advances of technology in many areas in the past decade have helped many people with disabilities to be more included in society. However, there are still areas which the latest technology is missing for people with disabilities. Video Relay Service (VRS) is a form of communication used by people with hearing disability to make a connection through phone, instead of text messaging. Such services connect the user to an assistant who translates the sign language made by the user into speech and communicates with the other side of the call. Such services could benefit from gesture recognition methods to automate the procedure and reduce costs which are often paid by the government or the state.

12.1.4 Integration into Video Search Systems

With the tremendous amount of media content across multiple devices, nowadays the need of a multi modal search system can be felt more than ever. *vitrivr* is a video retrieval system which benefits from diverse modalities and query types to search within a large collection of videos. Since a large amount of the videos contain humans, activity and specially gesture retrieval methods could be used as a new type of query to search the collection for similar movements. The integration of the methods of this thesis to systems such as *vitrivr*, will enable a deeper in-action evaluation of the pipeline through challenges such as the Video Browser Showdown (VBS)[10].

12.2 Future Work

This thesis is opening an under-explored path in the video retrieval area with focusing on communication hand gestures. However, the methods developed in this thesis can be further enhanced to address the existing challenges in the domain. In the following we will outline our suggestions for the future work in this area.

12.2.1 Content-Aware Gesture Retrieval

Co-speech gestures, as obvious by their names, have a strong tie with the spoken words. To search for them in video collections, analyzing both modalities –vision and speech– is important. Some of these gestures come with an indicator phrase which might be slightly different in wording. For example "*from the beginning to the end*" or "*from the start to finish*" comes usually with a hand gesture moving along the horizontal axis. This information already can be used to narrow down the results. Additionally, extending the existing vision-based method to a multi-modal pipeline to learn the similarity between the gestures by co-embedding the vision and speech (either audio or text) can bridge the existing semantic gap in the co-speech gesture retrieval.

[10] https://videobrowsershowdown.org/

12.2.2 Multi-View Gesture Recognition and Retrieval

The existing footage from talk shows often exhibit drastic camera movements and change of angles when recording the speakers. This viewpoint change causes confusion in the process of projecting the hand gestures into a 2D plane and then extracting features. This problem is not severe if the large amount of articulation is from one view, but in cases where the hand gesture is recorded from two or more points of views, the representation of gestures by the retrieval system will not be completely aligned with the reality of the gesture. One possible solution is to use the view independent feature embedding method [223] which can be trained by 2D projections of 3D poses, multi-view images and frames.

12.2.3 Temporal Gesture Localization in The Wild

According to our observations, the proposed temporal localization of gestures does not yield good results on real-world gestures. The limitation is the basic assumption in the QOM that the gesture duration is finished when the hands come back to the reference location. However, we can observe, even in our gesturing behavior, that not always the hands return to the starting location, and often the next gesture starts from where the previous one stopped. The idea of using binary localization for detecting gestures temporally also does not work in the real-world media data, as the “no gesture” label does not include all the absence of gesture instances in the datasets. Therefore, more thoughts should be put in designing a detection algorithm to further narrow down the results. One suggestion would be to use the attention mechanism in the temporal dimension in the LSTM model. This would attempt to score the existence of a gesture at each timestamp t and the final feature representation will aggregate these scores into a feature map.

12.2.4 In-depth Gesture Similarity Research

One step beyond this thesis which can open possibilities of using machine vision in interdisciplinary fields, is the study of the correlation of the hand gestures with facial expressions and gestural behaviors. During the study of the evaluation results, one hypothesis was made which suggests the potential existence of a pattern in gesture articulation of individuals. According to our hypothesis, this behavioral pattern would show itself by the appearance of the same person performing the query in the retrieved result set. This potentially could be of an interest to cognitive scientists to analyze this personal style of gesticulation. However, such a study would require more specific data, from each individual in different situations to extract such a style pattern which enables the inference of such correlation.

Appendix A

Supplementary Figures

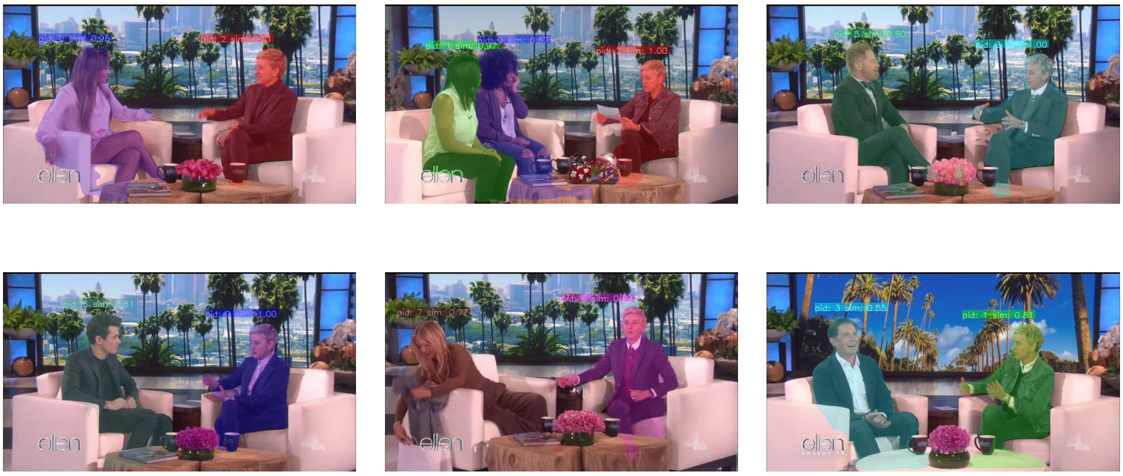


Figure A.1. key-frames from the query with the sitting pose of Ellen (top-left) and the results retrieved from RKLSTM method. The results demonstrated have a continuous sitting pose within the selected temporal boundry.

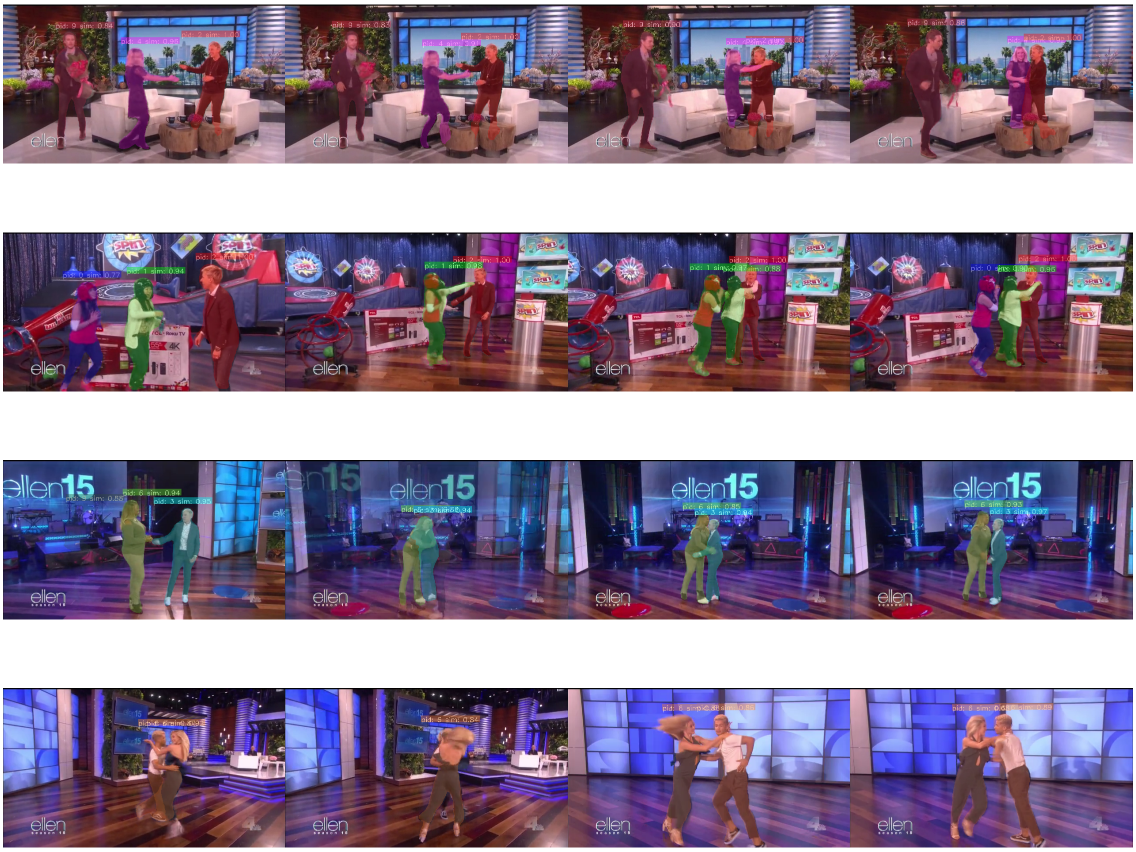


Figure A.2. Sample frames from the query two persons hugging (top) and three top results retrieved from RKLSTM method. The last result illustrates two people dancing, which might not reflect the hugging action, however, the movements are quite similar to the ones in a hug.

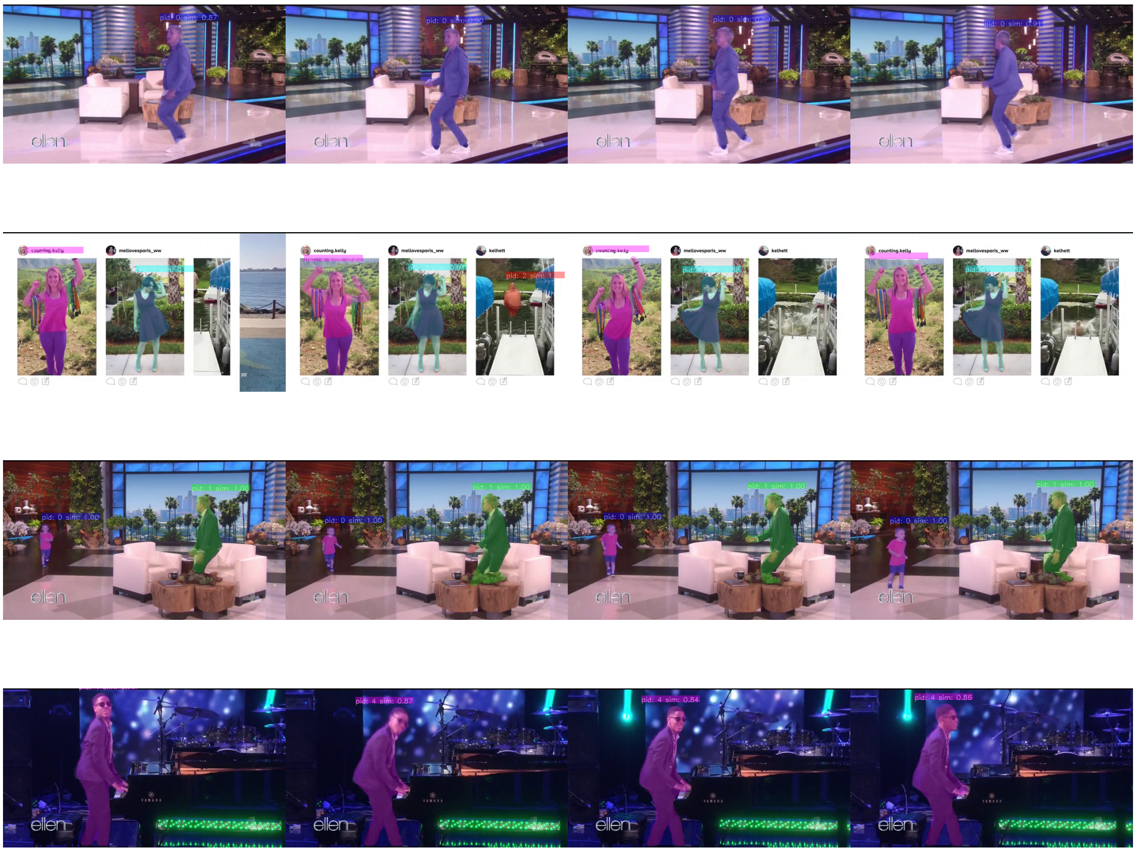
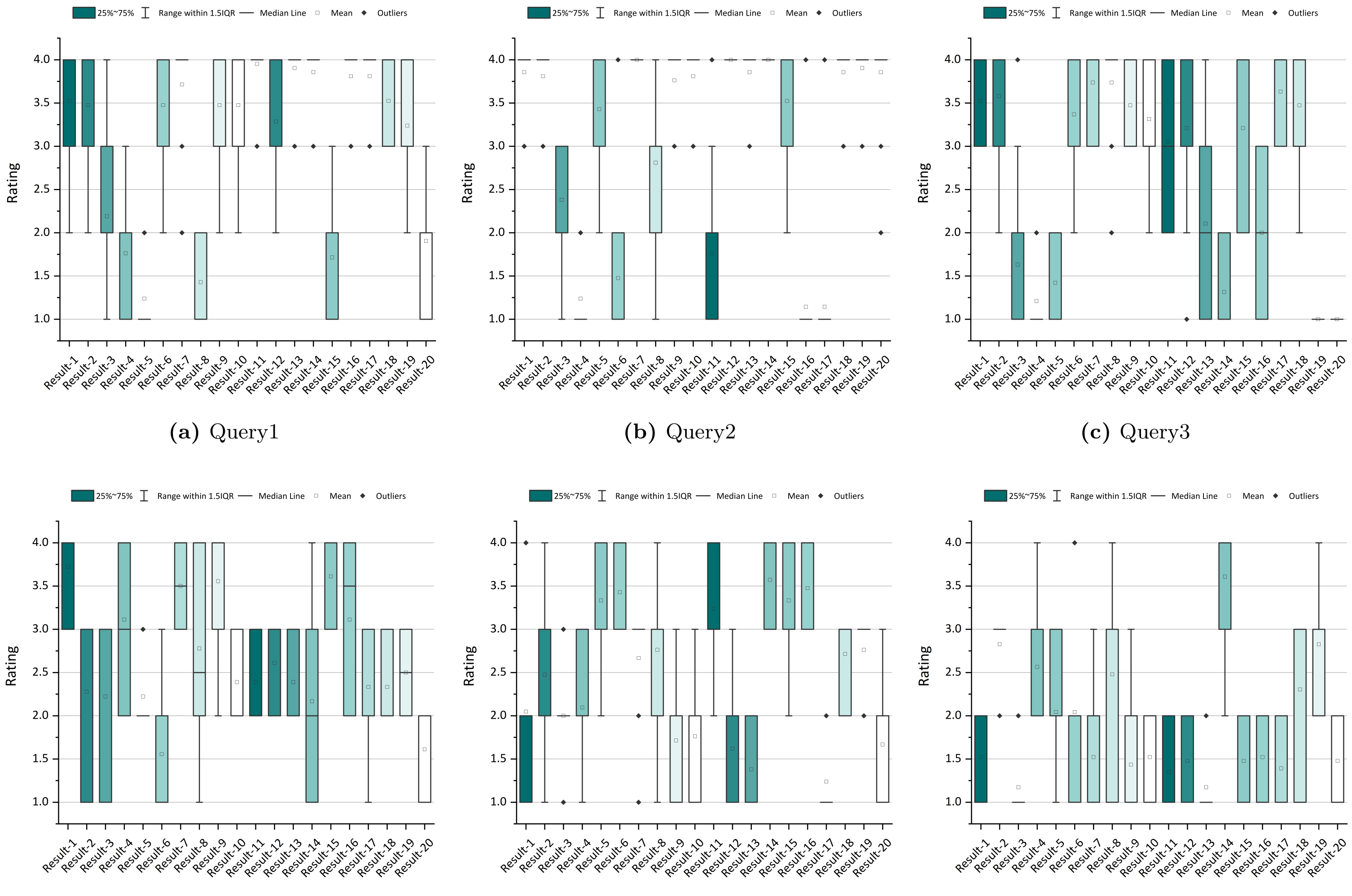


Figure A.3. Sample frames from the query showing a person dancing (top) and three top results retrieved from RKLSTM method.



(a) Query1

(b) Query2

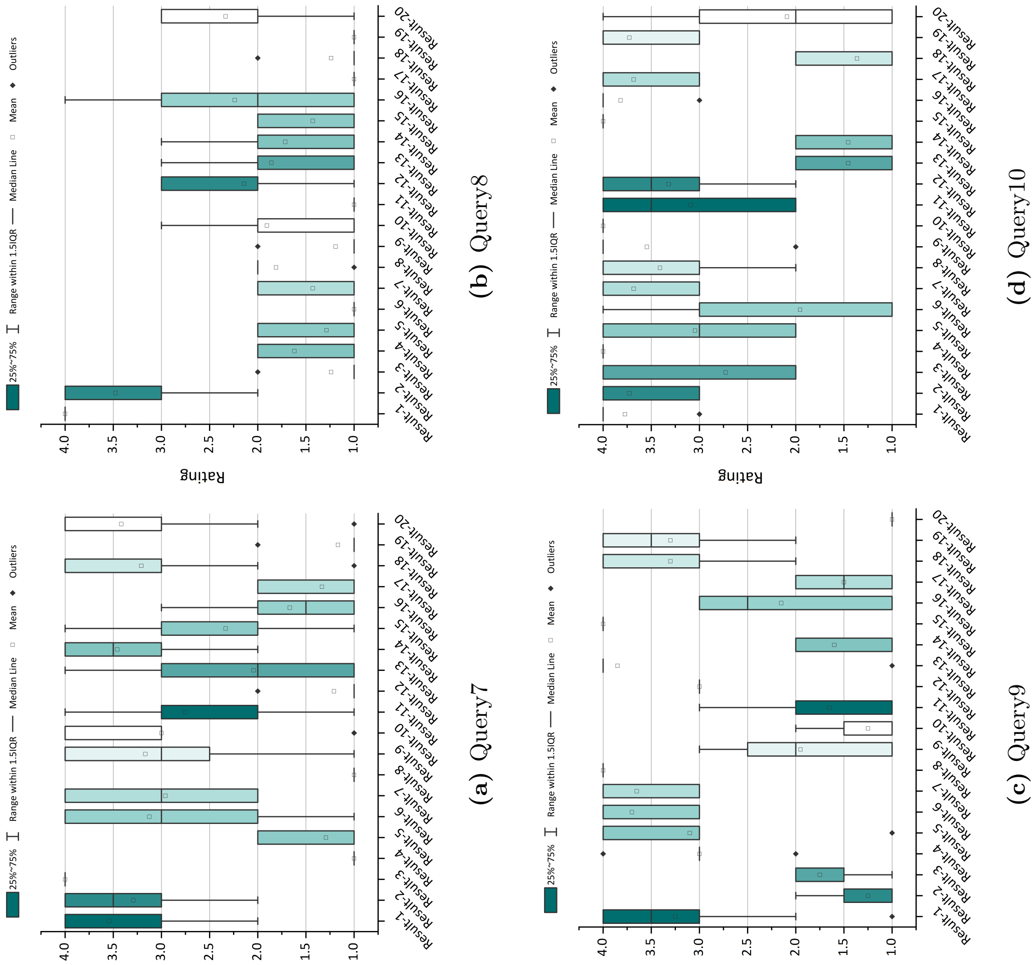
(c) Query3

(d) Query4

(e) Query5

(f) Query6

Figure A.4. Detailed statistics on scores given by non-linguists for each of the queries in evaluations on Newscape dataset.



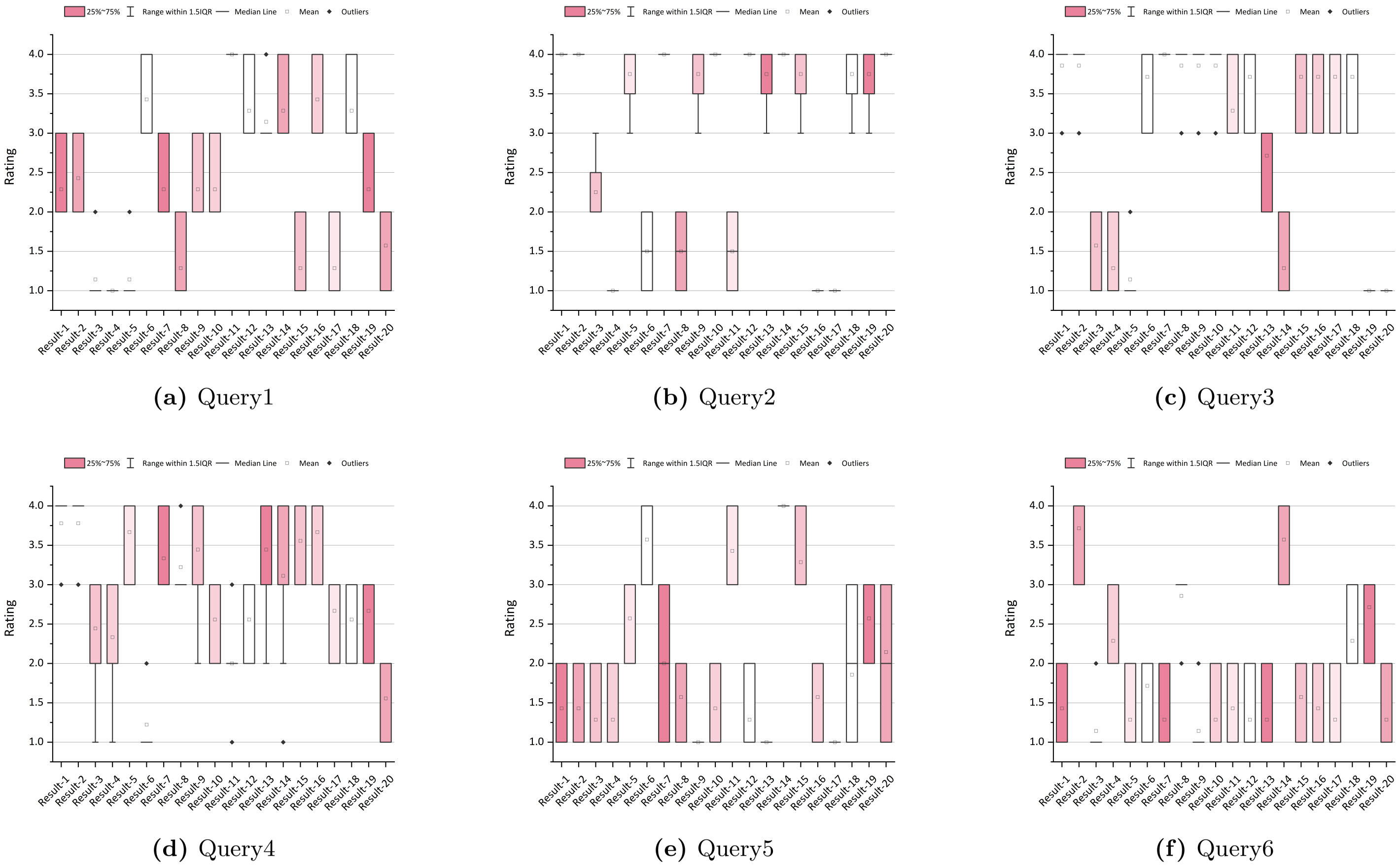
(a) Query7

(b) Query8

(c) Query9

(d) Query10

Figure A.5. Detailed statistics on scores given by non-linguists for each of the queries in evaluations on Newscape dataset.



(a) Query1

(b) Query2

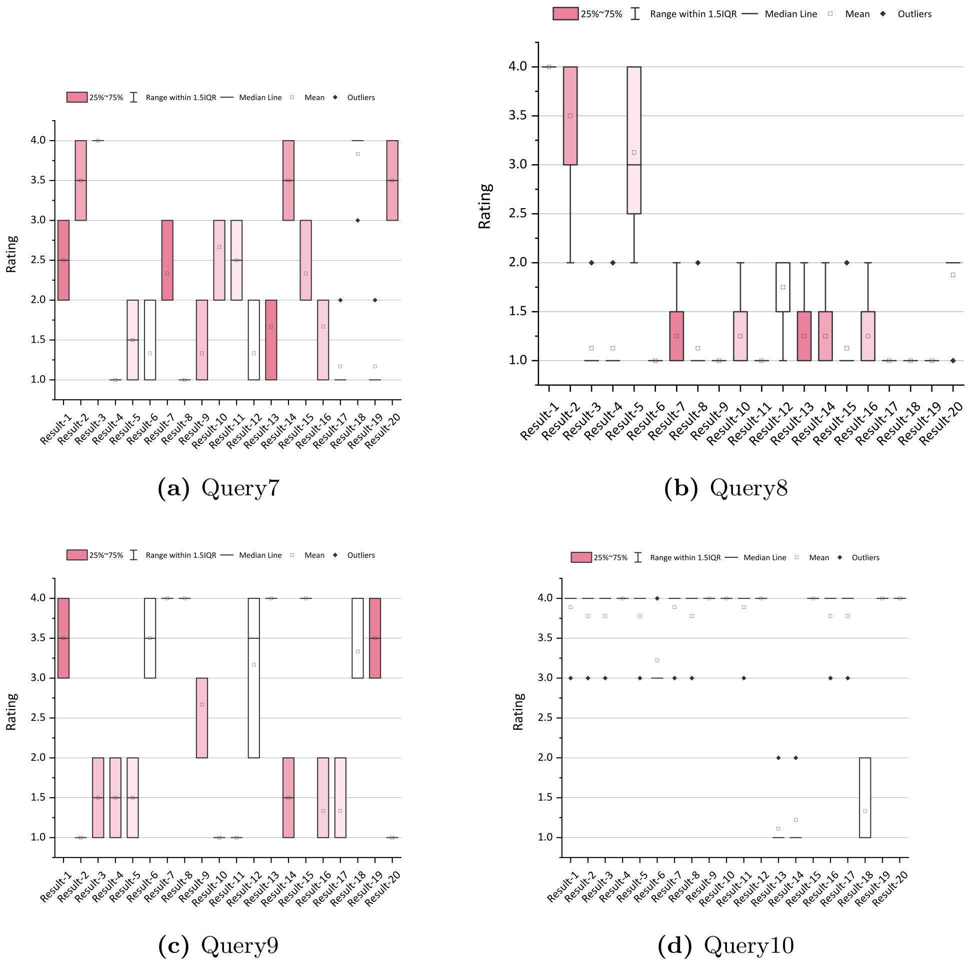
(c) Query3

(d) Query4

(e) Query5

(f) Query6

Figure A.6. Detailed statistics on scores given by linguists for each of the queries in evaluations on Newscape dataset.



(a) Query7

(b) Query8

(c) Query9

(d) Query10

Figure A.7. Detailed statistics on scores given by linguists for each of the queries in evaluations on Newscape dataset.

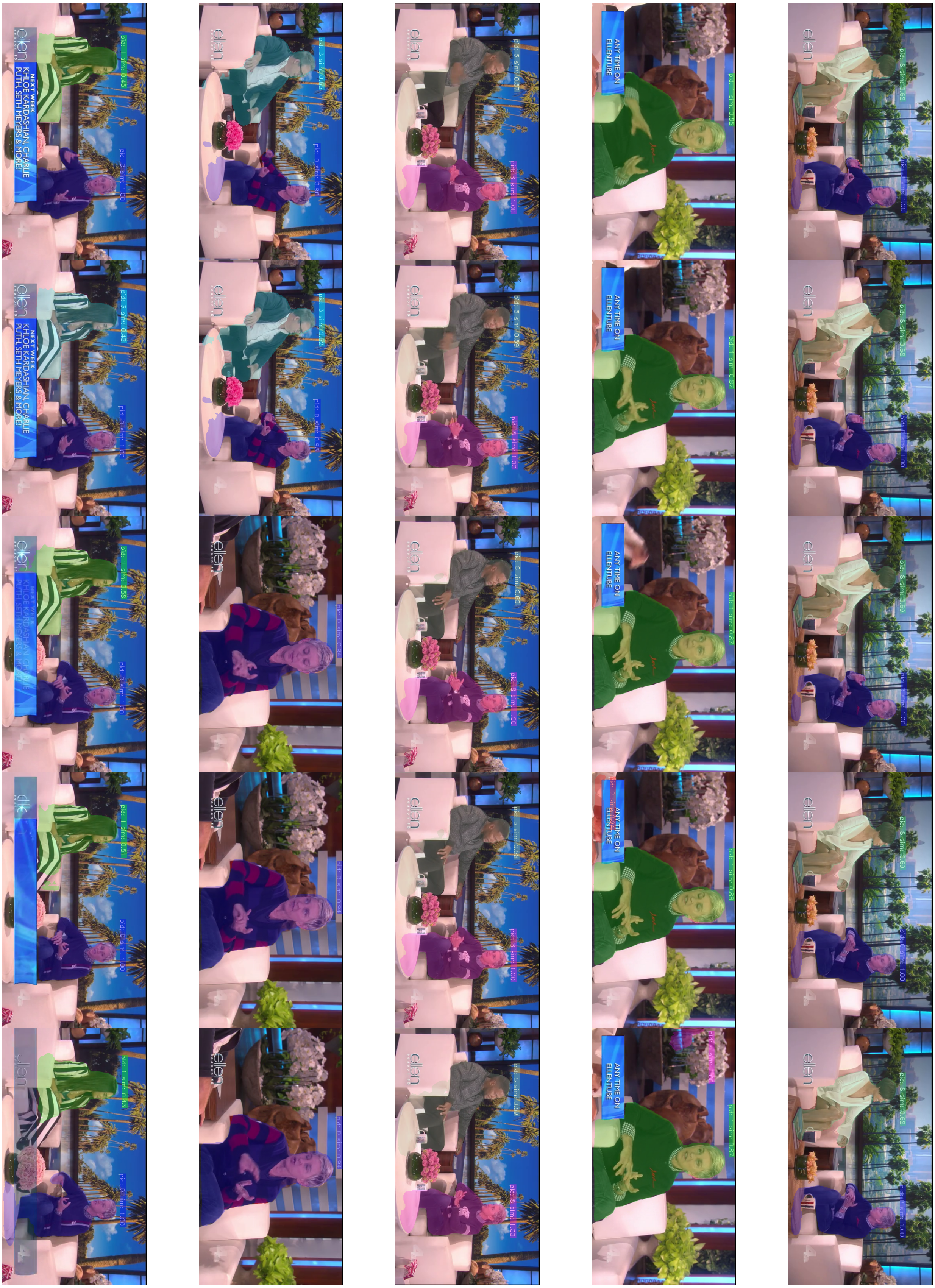


Figure A.8. Sample frames from the query with the Ellen counting figures while numerating verbally (top) and the results retrieved from RKLSTM method.

Bibliography

[1] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks”, in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[2] A. Arnab, S. Zheng, S. Jayasumana, B. Romera-Paredes, M. Larsson, A. Kirillov, B. Savchynskyy, C. Rother, F. Kahl, and P. H. Torr, “Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction”, *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 37–52, 2018.

[3] 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*, 2016, pp. 56–64.

[4] J. Materzynska, G. Berger, I. Bax, and R. Memisevic, “The jester dataset: A large-scale video dataset of human gestures”, in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[5] H. Jhuang, H. Garrote, E. Poggio, T. Serre, and T. Hmdb, “A large video database for human motion recognition”, in *Proc. of IEEE International Conference on Computer Vision*, vol. 4, no. 5, 2011, p. 6.

[6] J. Joo, F. F. Steen, and M. Turner, “Red hen lab: Dataset and tools for multimodal human communication research”, *KI-Künstliche Intelligenz*, vol. 31, no. 4, pp. 357–361, 2017.

[7] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles, “Activitynet: A large-scale video benchmark for human activity understanding”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970.

[8] S.-H. Zhang, R. Li, X. Dong, P. Rosin, Z. Cai, X. Han, D. Yang, H. Huang, and S.-M. Hu, “Pose2seg: Detection free human instance segmentation”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 889–898.

[9] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyra-

mid networks for object detection", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[10] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields", in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017.* IEEE Computer Society, 2017, pp. 1302–1310.

[11] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3415–3424.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[13] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset", in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[14] W. Du, Y. Wang, and Y. Qiao, "RPAN: An end-to-end recurrent pose-attention network for action recognition in videos", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3725–3734.

[15] N. Gupta, "Effective body language in organizations", *IUP Journal of Soft Skills*, vol. 7, no. 1, p. 35, 2013.

[16] K. Hogan, *Can't get through: eight barriers to communication.* Pelican Publishing, 2003.

[17] C. Pagán Cánovas, J. Valenzuela, D. Alcaraz Carrión, I. Olza, and M. Ramscar, "Quantifying the speech-gesture relation with massive multimodal datasets: Informativity in time expressions", *Plos one*, vol. 15, no. 6, p. e0233892, 2020.

[18] A. S. Dick, S. Goldin-Meadow, U. Hasson, J. I. Skipper, and S. L. Small, "Co-speech gestures influence neural activity in brain regions associated with processing semantic information", *Human brain mapping*, vol. 30, no. 11, pp. 3509–3526, 2009.

[19] A. Cienki and I. Mittelberg, "Creativity in the forms and functions of spontaneous gestures with speech", *The Agile Mind: A Multidisciplinary Study of a Multifaceted Phenomenon. Berlin, Germany: De Gruyter Mouton*, pp. 231–252, 2013.

[20] S. Goldin-Meadow and D. Brentari, "Gesture, sign, and language: The coming of age of sign language and gesture studies", *Behavioral and Brain Sciences*, vol. 40, 2017.

[21] M. R. Key, *The relationship of verbal and nonverbal communication.* Walter de Gruyter,

1980, no. 25.

[22] A. Kendon, *Gesture: Visible action as utterance.* Cambridge University Press, 2004.

[23] D. McNeill, *Hand and mind: What gestures reveal about thought.* University of Chicago press, 1992.

[24] Y. C. Wu and S. Coulson, "How iconic gestures enhance communication: An erp study", *Brain and language*, vol. 101, no. 3, pp. 234–245, 2007.

[25] R. M. Krauss, "Why do we gesture when we speak?" *Current directions in psychological science*, vol. 7, no. 2, pp. 54–54, 1998.

[26] A. Kendon, T. A. Sebeok, and J. Umiker-Sebeok, *Nonverbal communication, interaction, and gesture: selections from Semiotica.* Walter de Gruyter, 2010, vol. 41.

[27] C. D. Mortensen, *Communication theory.* Routledge, 2017.

[28] R. Krauss, Y. Chen, and R. Gottesman, "Lexical gestures and lexical access: a process model. teoksessa d. mcneill (toim.) language and gesture, 261-283", 2001.

[29] P. Ekman and W. V. Friesen, "The repertoire of nonverbal behavior: Categories, origins, usage, and coding", *Nonverbal communication, interaction, and gesture*, pp. 57–106, 1969.

[30] D. McNeill, "Gesture: a psycholinguistic approach", *The encyclopedia of language and linguistics*, pp. 58–66, 2006.

[31] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[32] M. Minsky and S. A. Papert, *Perceptrons: An introduction to computational geometry.* MIT press, 2017.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors", *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[35] S. Ruder, "An overview of gradient descent optimization algorithms", *arXiv preprint arXiv:1609.04747*, 2016.

[36] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[37] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition", *Neural*

computation, vol. 1, no. 4, pp. 541–551, 1989.

[38] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio", *arXiv preprint arXiv:1609.03499*, 2016.

[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[40] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm networks", in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 4. IEEE, 2005, pp. 2047–2052.

[41] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks", *Advances in neural information processing systems*, vol. 27, pp. 3104–3112, 2014.

[42] H. Larochelle and G. E. Hinton, "Learning to combine foveal glimpses with a third-order boltzmann machine", in *Advances in neural information processing systems*, 2010, pp. 1243–1251.

[43] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention", *Advances in neural information processing systems*, vol. 27, pp. 2204–2212, 2014.

[44] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: unsupervised learning using temporal order verification", in *European Conference on Computer Vision.* Springer, 2016, pp. 527–544.

[45] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate", *arXiv preprint arXiv:1409.0473*, 2014.

[46] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[47] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[48] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning.* MIT press Cambridge, 2016, vol. 1, no. 2.

[49] L. Rossetto, "Multi-modal video retrieval", Ph.D. dissertation, University_of_Basel, 2018.

[50] J.-J. Wen and S. Wang, "Seeker: Keyword-based information retrieval over relational

databases." *Ruan Jian Xue Bao(J. Softw.)*, vol. 16, no. 7, pp. 1270–1281, 2005.

[51] S. Bai and S. An, "A survey on automatic image caption generation", *Neurocomputing*, vol. 311, pp. 291–304, 2018.

[52] L. Rossetto, I. Giangreco, C. Tanase, and H. Schuldt, "vitrivr: A flexible retrieval stack supporting multiple query modes for searching in multimedia collections", in *Proceedings of the 2016 ACM on Multimedia Conference.* ACM, 2016, pp. 1183–1186.

[53] L. Rossetto, I. Giangreco, S. Heller, C. Tănase, H. Schuldt, S. Dupont, O. Seddati, M. Sezgin, O. C. Altıok, and Y. Sahillioğlu, "Imotion–searching for video sequences using multi-shot sketch queries", in *International Conference on Multimedia Modeling.* Springer, 2016, pp. 377–382.

[54] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Paper recommender systems: a literature survey", *International Journal on Digital Libraries*, vol. 17, no. 4, pp. 305–338, 2016.

[55] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition", in *ICML deep learning workshop*, vol. 2. Lille, 2015.

[56] C. Silberer and M. Lapata, "Learning grounded meaning representations with autoencoders", in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 721–732.

[57] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping", in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[58] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering", in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015.* IEEE Computer Society, 2015, pp. 815–823.

[59] R. Ohlander, K. Price, and D. R. Reddy, "Picture segmentation using a recursive region splitting method", *Computer graphics and image processing*, vol. 8, no. 3, pp. 313–333, 1978.

[60] T. Lindeberg and M.-X. Li, "Segmentation and classification of edges using minimum description length approximation and complementary junction cues", *Computer Vision and Image Understanding*, vol. 67, no. 1, pp. 88–98, 1997.

[61] S. Guberman, V. V. Maximov, and A. Pashintsev, "Gestalt and image understanding", *Gestalt Theory*, vol. 34, no. 2, p. 143, 2012.

[62] 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*, 2014, pp. 580–587.

[63] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition", *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[64] R. Girshick, "Fast R-CNN", in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[65] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár, "A multipath network for object detection", *arXiv preprint arXiv:1604.02135*, 2016.

[66] P. O. O Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates", *Advances in neural information processing systems*, vol. 28, pp. 1990–1998, 2015.

[67] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks", *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[68] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN", in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[69] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[70] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation", *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[71] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", in *International Conference on Medical image computing and computer-assisted intervention.* Springer, 2015, pp. 234–241.

[72] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4929–4937.

[73] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, "Towards accurate multi-person pose estimation in the wild", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4903–4911.

[74] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017,

pp. 2334–2343.

[75] S. Tripathi, M. Collins, M. Brown, and S. Belongie, "Pose2instance: Harnessing keypoints for person instance segmentation", *arXiv preprint arXiv:1704.01152*, 2017.

[76] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy, "Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 269–286.

[77] D. Zhou and Q. He, "Poseg: Pose-aware refinement network for human instance segmentation", *IEEE Access*, vol. 8, pp. 15 007–15 016, 2020.

[78] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage CNNs", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1049–1058.

[79] 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*, 2015, pp. 4489–4497.

[80] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5734–5743.

[81] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia, "Turn tap: Temporal unit regression network for temporal action proposals", in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3628–3636.

[82] Y. Xiong, Y. Zhao, L. Wang, D. Lin, and X. Tang, "A pursuit of temporal accuracy in general activity detection", *arXiv preprint arXiv:1703.02716*, 2017.

[83] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition", in *European conference on computer vision.* Springer, 2016, pp. 20–36.

[84] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. Carlos Niebles, "Sst: Single-stream temporal action proposals", in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2911–2920.

[85] J. Gao, K. Chen, and R. Nevatia, "Ctap: Complementary temporal action proposal generation", in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 68–83.

[86] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, "Bsn: Boundary sensitive network for

temporal action proposal generation", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[87] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, "Bmn: Boundary-matching network for temporal action proposal generation", in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3889–3898.

[88] Z. Liu, X. Chai, Z. Liu, and X. Chen, "Continuous gesture recognition with hand-oriented spatiotemporal feature", in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[89] H. Wang, P. Wang, Z. Song, and W. Li, "Large-scale multimodal gesture recognition using heterogeneous networks", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3129–3137.

[90] N. Cihan Camgoz, S. Hadfield, and R. Bowden, "Particle filter based probabilistic forced alignment for continuous gesture recognition", in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3079–3085.

[91] P. Narayana, J. R. Beveridge, and B. A. Draper, "Continuous gesture recognition through selective temporal fusion", in *2019 International Joint Conference on Neural Networks (IJCNN).* IEEE, 2019, pp. 1–8.

[92] Y. Iwai, H. Shimizu, and M. Yachida, "Real-time context-based gesture recognition using hmm and automaton", in *Proceedings International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems. In Conjunction with ICCV'99 (Cat. No. PR00378).* IEEE, 1999, pp. 127–134.

[93] B. Stenger, "Template-based hand pose recognition using multiple cues", in *Asian conference on computer vision.* Springer, 2006, pp. 551–560.

[94] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering", in *Proceedings of fifth IEEE international conference on automatic face gesture recognition.* IEEE, 2002, pp. 423–428.

[95] N. D. Binh, E. Shuichi, and T. Ejima, "Real-time hand tracking and gesture recognition system", *Proc. GVIP*, pp. 19–21, 2005.

[96] H. Wang and C. Schmid, "Action recognition with improved trajectories", in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558.

[97] V. N. J. J. Clark *et al.*, "Automated visual surveillance using hidden markov models", in *International Conference on Vision Interface*, 2002, pp. 88–93.

[98] S. Marcel, O. Bernier, J.-E. Viallet, and D. Collobert, "Hand gesture recognition using

input-output hidden markov models", in *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580).* IEEE, 2000, pp. 456–461.

[99] W.-L. Lu and J. J. Little, "Tracking and recognizing actions at a distance", in *Proceedings of the ECCV Workshop on Computer Vision Based Analysis in Sport Environments (CVBASE'06), Graz, Austria*, vol. 1, 2006.

[100] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes", *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007.

[101] S. Gutta, H. Huang, F. Imam, and H. Wechsler, "Face and hand gesture recognition using hybrid classifiers", in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition.* IEEE, 1996, pp. 164–169.

[102] Z. Hang and R. Qiuqi, "Visual gesture recognition with color segmentation and support vector machines", in *Proceedings 7th International Conference on Signal Processing, 2004. Proceedings. ICSP'04. 2004.*, vol. 2. IEEE, 2004, pp. 1443–1446.

[103] Q. Chen, "Real-time vision-based hand tracking and gesture recognition", Ph.D. dissertation, University of Ottawa (Canada), 2008.

[104] G. Hua and Y. Wu, "Multi-scale visual tracking by sequential belief propagation", in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1. IEEE, 2004, pp. I–I.

[105] A. El-Sawah, N. D. Georganas, and E. M. Petriu, "A prototype for 3-d hand tracking and posture estimation", *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 8, pp. 1627–1636, 2008.

[106] W. Zhao, Y.-G. Jiang, and C.-W. Ngo, "Keyframe retrieval by keypoints: Can point-to-point matching help?" in *International Conference on Image and Video Retrieval.* Springer, 2006, pp. 72–81.

[107] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval", in *Proceedings of the 6th ACM international conference on Image and video retrieval*, 2007, pp. 494–501.

[108] A. Gilbert, J. Illingworth, and R. Bowden, "Fast realistic multi-action recognition using mined dense spatio-temporal features", in *2009 IEEE 12th international conference on computer vision.* IEEE, 2009, pp. 925–931.

[109] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies", in *2008 IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 2008, pp. 1–8.

[110] M. Marszalek, I. Laptev, and C. Schmid, “Actions in context”, in *2009 IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 2009, pp. 2929–2936.

[111] N. Ikizler and P. Duygulu, “Human action recognition using distribution of oriented rectangular patches”, in *Workshop on Human Motion.* Springer, 2007, pp. 271–284.

[112] Y. Wang, P. Sabzmeydani, and G. Mori, “Semi-latent dirichlet allocation: A hierarchical model for human action recognition”, in *Workshop on Human Motion.* Springer, 2007, pp. 240–254.

[113] S.-F. Wong, T.-K. Kim, and R. Cipolla, “Learning motion categories using both semantic and structural information”, in *2007 IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 2007, pp. 1–6.

[114] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks”, in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[115] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[116] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning”, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[117] O. Patsadu, C. Nukoolkit, and B. Watanapa, “Human gesture recognition using kinect camera”, in *2012 ninth international conference on computer science and software engineering (JCSSE).* IEEE, 2012, pp. 28–32.

[118] H. Hikawa and Y. Araga, “Study on gesture recognition system using posture classifier and jordan recurrent neural network”, in *The 2011 International Joint Conference on Neural Networks.* IEEE, 2011, pp. 405–412.

[119] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, “Advances in optimizing recurrent networks”, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing.* IEEE, 2013, pp. 8624–8628.

[120] G. Lefebvre, S. Berlemont, F. Mamalet, and C. Garcia, “Blstm-rnn based 3d gesture classification”, in *International conference on artificial neural networks.* Springer, 2013, pp. 381–388.

[121] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, “Hand gesture recognition with 3d convolutional neural networks”, in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 1–7.

[122] L. Zhang, G. Zhu, P. Shen, J. Song, S. Afaq Shah, and M. Bennamoun, “Learning spatiotemporal features using 3DCNN and convolutional lstm for gesture recognition”, in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 3120–3128.

[123] Q. Miao, Y. Li, W. Ouyang, Z. Ma, X. Xu, W. Shi, and X. Cao, “Multimodal gesture recognition based on the resc3d network”, in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3047–3055.

[124] M. A. Goodale, A. D. Milner *et al.*, “Separate visual pathways for perception and action”, 1992.

[125] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, *arXiv preprint arXiv:1409.1556*, 2014.

[126] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutional two-stream network fusion for video action recognition”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1933–1941.

[127] C. Feichtenhofer, A. Pinz, and R. P. Wildes, “Temporal residual networks for dynamic scene recognition”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4728–4737.

[128] ——, “Spatiotemporal multiplier networks for video action recognition”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4768–4777.

[129] H. Wang, P. Wang, Z. Song, and W. Li, “Large-scale multimodal gesture recognition using heterogeneous networks”, in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3129–3137.

[130] J. Wan, S. Escalera, G. Anbarjafari, H. Jair Escalante, X. Baró, I. Guyon, M. Madadi, J. Allik, J. Gorbova, C. Lin *et al.*, “Results and analysis of chalearn lap multi-modal isolated and continuous gesture recognition, and real versus fake expressed emotions challenges”, in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3189–3197.

[131] S. Sharma, R. Kiros, and R. Salakhutdinov, “Action recognition using visual attention”, *CoRR*, vol. abs/1511.04119, 2015.

[132] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. M. Snoek, “Videolstm convolves, attends and flows for action recognition”, *Comput. Vis. Image Underst.*, vol. 166, pp. 41–50, 2018.

[133] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial transformer networks”, in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal,*

Quebec, Canada, 2015, pp. 2017–2025.

[134] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need", *arXiv preprint arXiv:1706.03762*, 2017.

[135] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network", in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019.* Computer Vision Foundation / IEEE, 2019, pp. 244–253.

[136] C. Cao, Y. Zhang, C. Zhang, and H. Lu, "Body joint guided 3-d deep convolutional descriptors for action recognition", *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 1095–1108, 2018.

[137] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, "Moddrop: adaptive multi-modal gesture recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1692–1706, 2015.

[138] H. Wang and L. Wang, "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 499–508.

[139] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1110–1118.

[140] P. Wang, W. Li, C. Li, and Y. Hou, "Action recognition based on joint trajectory maps with convolutional neural networks", *Knowledge-Based Systems*, vol. 158, pp. 43–53, 2018.

[141] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention lstm networks for 3d action recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1647–1656.

[142] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data", in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[143] Q. Ke, S. An, M. Bennamoun, F. Sohel, and F. Boussaid, "Skeletonnet: Mining deep part features for 3-d action recognition", *IEEE signal processing letters*, vol. 24, no. 6, pp. 731–735, 2017.

[144] J. Li, X. Xie, Q. Pan, Y. Cao, Z. Zhao, and G. Shi, "Sgm-net: Skeleton-guided multimodal network for action recognition", *Pattern Recognition*, p. 107356, 2020.

[145] K. Gavrilyuk, R. Sanford, M. Javan, and C. G. Snoek, "Actor-transformers for group

activity recognition", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 839–848.

[146] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 797–819, 2011.

[147] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey", *Speech Communication*, vol. 56, pp. 85–100, 2014.

[148] L. Rossetto, M. Amiri Parian, I. Giangreco, R. Gasser, S. Heller, and H. Schuldt, "Deep learning-based concept detection in vitrivr", in *25th International Conference on MultiMedia Modeling.* Springer, 2019.

[149] L. Rossetto, I. Giangreco, H. Schuldt, S. Dupont, O. Seddati, M. Sezgin, and Y. Sahillioğlu, "Imotion—a content-based video retrieval engine", in *International Conference on Multimedia Modeling.* Springer, 2015, pp. 255–260.

[150] L. Rossetto, M. Amiri Parian, R. Gasser, I. Giangreco, S. Heller, and H. Schuldt, "Deep learning-based concept detection in vitrivr", in *MultiMedia Modeling*, I. Kompatsiaris, B. Huet, V. Mezaris, C. Gurrin, W.-H. Cheng, and S. Vrochidis, Eds. Cham: Springer International Publishing, 2019, pp. 616–621.

[151] N. Spolaôr, H. D. Lee, W. S. R. Takaki, L. A. Ensina, C. S. R. Coy, and F. C. Wu, "A systematic review on content-based video retrieval", *Engineering Applications of Artificial Intelligence*, vol. 90, p. 103557, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197620300488

[152] U. Gargi, R. Kasturi, and S. H. Strayer, "Performance characterization of video-shot-change detection methods", *IEEE transactions on circuits and systems for video technology*, vol. 10, no. 1, pp. 1–13, 2000.

[153] J. E. Jackson, *A user's guide to principal components.* John Wiley & Sons, 2005, vol. 587.

[154] H. Liu and H. Motoda, *Computational methods of feature selection.* CRC Press, 2007.

[155] B. Stein and S. M. zu Eissen, "Fingerprint-based similarity search and its applications", *Universität Weimar*, 2007.

[156] J. Wang, W. Liu, S. Kumar, and S.-F. Chang, "Learning to hash for indexing big data—a survey", *Proceedings of the IEEE*, vol. 104, no. 1, pp. 34–57, 2015.

[157] J. L. Bentley, "Multidimensional binary search trees used for associative searching", *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[158] A. Likas, N. Vlassis, and J. J. Verbeek, “The global k-means clustering algorithm”, *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[159] H. V. Jagadish, B. C. Ooi, K.-L. Tan, C. Yu, and R. Zhang, “idistance: An adaptive b+-tree based indexing method for nearest neighbor search”, *ACM Transactions on Database Systems (TODS)*, vol. 30, no. 2, pp. 364–397, 2005.

[160] A. Gionis, P. Indyk, R. Motwani *et al.*, “Similarity search in high dimensions via hashing”, in *VLDB*, vol. 99, no. 6, 1999, pp. 518–529.

[161] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, “Locality-sensitive hashing scheme based on p-stable distributions”, in *Proceedings of the twentieth annual symposium on Computational geometry.* ACM, 2004, pp. 253–262.

[162] M. S. Charikar, “Similarity estimation techniques from rounding algorithms”, in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing.* ACM, 2002, pp. 380–388.

[163] R. Panigrahy, “Entropy based nearest neighbor search in high dimensions”, *arXiv preprint cs/0510019*, 2005.

[164] V. Erin Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, “Deep hashing for compact binary codes learning”, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2475–2483.

[165] Z. Cao, M. Long, J. Wang, and P. S. Yu, “Hashnet: Deep learning to hash by continuation”, in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5608–5617.

[166] Y. Cao, M. Long, J. Wang, Q. Yang, and P. S. Yu, “Deep visual-semantic hashing for cross-modal retrieval”, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 2016, pp. 1445–1454.

[167] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, “Multiple feature hashing for real-time large scale near-duplicate video retrieval”, in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 423–432.

[168] V. E. Liong, J. Lu, Y.-P. Tan, and J. Zhou, “Deep video hashing”, *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1209–1219, 2016.

[169] L. Shen, R. Hong, H. Zhang, X. Tian, and M. Wang, “Video retrieval with similarity-preserving deep temporal hashing”, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 4, pp. 1–16, 2019.

[170] J. Song, H. Zhang, X. Li, L. Gao, M. Wang, and R. Hong, “Self-supervised video hashing with hierarchical binary auto-encoder”, *IEEE Transactions on Image Processing*,

vol. 27, no. 7, pp. 3210–3221, 2018.

[171] G. Wu, L. Liu, Y. Guo, G. Ding, J. Han, J. Shen, and L. Shao, “Unsupervised deep video hashing with balanced rotation”. IJCAI, 2017.

[172] G. Wu, J. Han, Y. Guo, L. Liu, G. Ding, Q. Ni, and L. Shao, “Unsupervised deep video hashing via balanced code for large-scale video retrieval”, *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1993–2007, 2018.

[173] Y. Wang, X. Nie, Y. Shi, X. Zhou, and Y. Yin, “Attention-based video hashing for large-scale video retrieval”, *IEEE Transactions on Cognitive and Developmental Systems*, 2019.

[174] A. Stefan, H. Wang, and V. Athitsos, “Towards automated large vocabulary gesture search”, in *Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments*, 2009, pp. 1–8.

[175] A. Corradini, “Dynamic time warping for off-line recognition of a small gesture vocabulary”, in *Proceedings IEEE ICCV workshop on recognition, analysis, and tracking of faces and gestures in real-time systems.* IEEE, 2001, pp. 82–89.

[176] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, “Simultaneous localization and recognition of dynamic hand gestures”, in *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION’05)-Volume 1*, vol. 2. IEEE, 2005, pp. 254–260.

[177] S. Yousefi and H. Li, “3d hand gesture analysis through a real-time gesture search engine”, *International Journal of Advanced Robotic Systems*, vol. 12, no. 6, p. 67, 2015.

[178] V. Ferrari, M. J. Marín-Jiménez, and A. Zisserman, “Pose search: Retrieving people using their pose”, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR, USA*, 2009.

[179] M. Eichner, M. Marin, A. Zisserman, and V. Ferrari. 2d human pose estimation and search in tv shows and movies. [Online]. Available: https://www.robots.ox.ac.uk/~vgg/research/pose_estimation/index.html

[180] C. Zhang, “Dynamic gesture retrieval: searching videos by human pose sequence”, 2020.

[181] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context”, in *European conference on computer vision.* Springer, 2014, pp. 740–755.

[182] O. Köpüklü, A. Gunduz, N. Kose, and G. Rigoll, “Real-time hand gesture detection and classification using convolutional neural networks”, *arXiv preprint arXiv:1901.10323*, 2019.

[183] F. Jiang, S. Zhang, S. Wu, Y. Gao, and D. Zhao, “Multi-layered gesture recognition

with kinect." *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 227–254, 2015.

[184] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", in *2009 IEEE conference on computer vision and pattern recognition.* Ieee, 2009, pp. 248–255.

[185] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset", *arXiv preprint arXiv:1705.06950*, 2017.

[186] C. Ma, J. Huang, X. Yang, and M. Yang, "Robust visual tracking via hierarchical convolutional features", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2709–2723, Nov 2019.

[187] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow", in *Joint pattern recognition symposium.* Springer, 2007, pp. 214–223.

[188] B. Liu, Y. Cao, M. Long, J. Wang, and J. Wang, "Deep triplet quantization", in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 755–763.

[189] P. Wang, W. Li, S. Liu, Z. Gao, C. Tang, and P. Ogunbona, "Large-scale isolated gesture recognition using convolutional neural networks", in *2016 23rd International Conference on Pattern Recognition (ICPR).* IEEE, 2016, pp. 7–12.

[190] G. Zhu, L. Zhang, L. Yang, L. Mei, S. A. A. Shah, M. Bennamoun, and P. Shen, "Redundancy and attention in convolutional lstm for gesture recognition", *IEEE transactions on neural networks and learning systems*, vol. 31, no. 4, pp. 1323–1335, 2019.

[191] Y. Li, Q. Miao, K. Tian, Y. Fan, X. Xu, R. Li, and J. Song, "Large-scale gesture recognition with a fusion of rgb-d data based on saliency theory and c3d model", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2956–2964, 2017.

[192] G. Zhu, L. Zhang, L. Mei, J. Shao, J. Song, and P. Shen, "Large-scale isolated gesture recognition using pyramidal 3d convolutional networks", in *2016 23rd International Conference on Pattern Recognition (ICPR).* IEEE, 2016, pp. 19–24.

[193] Y. Li, "Simple Trick: Masked C3D for Isolated Sign Language Recognition", http://chalearnlap.cvc.uab.es/media/results/None/iccv17_isogd_lostoy_3rd.pdf.

[194] J. Duan, J. Wan, S. Zhou, X. Guo, and S. Z. Li, "A unified framework for multi-modal isolated gesture recognition", *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1s, pp. 1–16, 2018.

[195] H. Wang, P. Wang, Z. Song, and W. Li, "Large-scale multimodal gesture segmentation

and recognition based on convolutional neural networks", in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 3138–3146.

[196] P. Narayana, R. Beveridge, and B. A. Draper, "Gesture recognition: Focus on the hands", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5235–5244.

[197] Y. Li, Q. Miao, X. Qi, Z. Ma, and W. Ouyang, "A spatiotemporal attention-based resc3d model for large-scale gesture recognition", *Machine Vision and Applications*, vol. 30, no. 5, pp. 875–888, 2019.

[198] C. Lin, J. Wan, Y. Liang, and S. Z. Li, "Large-scale isolated gesture recognition using a refined fused model based on masked res-C3D network and skeleton lstm", in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 52–58.

[199] W. Li, J.-F. Hu, B. Li, and W.-S. Zheng, "Learning both dynamic and static action context for gesture recognition", http://chalearnlap.cvc.uab.es/media/results/None/iccv17_isogd_SYSU%20ISEE_2nd.pdf.

[200] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal relational reasoning in videos", in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 803–818.

[201] K. Yang, R. Li, P. Qiao, Q. Wang, D. Li, and Y. Dou, "Temporal pyramid relation network for video-based gesture recognition", in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 3104–3108.

[202] B. Yu, Z. Luo, H. Wu, and S. Li, "Hand gesture recognition based on attentive feature fusion", *Concurrency and Computation: Practice and Experience*, vol. 32, no. 22, p. e5910, 2020.

[203] O. Kopuklu, N. Kose, and G. Rigoll, "Motion fused frames: Data level fusion strategy for hand gesture recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2103–2111.

[204] J. Wan, C. Lin, L. Wen, Y. Li, Q. Miao, S. Escalera, G. Anbarjafari, I. Guyon, G. Guo, and S. Z. Li, "Chalearn looking at people: Isogd and congd large-scale rgb-d gesture recognition", *IEEE Transactions on Cybernetics*, 2020.

[205] P. Wang, W. Li, S. Liu, Y. Zhang, Z. Gao, and P. Ogunbona, "Large-scale continuous gesture recognition using convolutional neural networks", in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 13–18.

[206] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden, "Using convolutional 3d neural networks for user-independent continuous gesture recognition", in *2016 23rd Interna-*

tional Conference on Pattern Recognition (ICPR). IEEE, 2016, pp. 49–54.

[207] L. Pigou, M. Van Herreweghe, and J. Dambre, "Gesture and sign language recognition with temporal residual networks", in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 3086–3093.

[208] 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*, 2017, pp. 3056–3064.

[209] X. Chai, Z. Liu, F. Yin, Z. Liu, and X. Chen, "Two streams recurrent neural networks for large-scale continuous gesture recognition", in *2016 23rd International Conference on Pattern Recognition (ICPR).* IEEE, 2016, pp. 31–36.

[210] N. N. Hoang, G.-S. Lee, S.-H. Kim, and H.-J. Yang, "Continuous hand gesture spotting and classification using 3d finger joints information", in *2019 IEEE International Conference on Image Processing (ICIP).* IEEE, 2019, pp. 539–543.

[211] G. Zhu, L. Zhang, P. Shen, J. Song, S. A. A. Shah, and M. Bennamoun, "Continuous gesture segmentation and recognition using 3DCNN and convolutional lstm", *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 1011–1021, 2018.

[212] J. L. Fleiss, "Measuring nominal scale agreement among many raters." *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.

[213] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition", in *International Conf. on Computer Vision (ICCV)*, Dec. 2013, pp. 3192–3199.

[214] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval", *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2916–2929, 2012.

[215] R. Anuranji and H. Srimathi, "A supervised deep convolutional based bidirectional long short term memory video hashing for large scale video retrieval applications", *Digital Signal Processing*, vol. 102, p. 102729, 2020.

[216] Z. Jin, C. Li, Y. Lin, and D. Cai, "Density sensitive hashing", *IEEE transactions on cybernetics*, vol. 44, no. 8, pp. 1362–1371, 2013.

[217] X. Liu, X. Nie, Q. Zhou, X. Xi, L. Zhu, and Y. Yin, "Supervised short-length hashing." in *IJCAI*, 2019, pp. 3031–3037.

[218] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions", *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, Nov 2016.

[219] Z. Zhang, S. Wei, Y. Song, and Y. Zhang, "Gesture recognition using enhanced depth motion map and static pose map", in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017).* IEEE, 2017, pp. 238–244.

[220] N. Kawarazaki, I. Hoya, K. Nishihara, and T. Yoshidome, "7 cooperative welfare robot system using hand gesture instructions", in *Advances in Rehabilitation Robotics.* Springer, 2004, pp. 143–153.

[221] O. Rogalla, M. Ehrenmann, R. Zollner, R. Becher, and R. Dillmann, "Using gesture and speech control for commanding a robot assistant", in *Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication.* IEEE, 2002, pp. 454–459.

[222] Y.-T. Chen and K.-T. Tseng, "Developing a multiple-angle hand gesture recognition system for human machine interactions", in *IECON 2007-33rd Annual Conference of the IEEE Industrial Electronics Society.* IEEE, 2007, pp. 489–492.

[223] J. J. Sun, J. Zhao, L.-C. Chen, F. Schroff, H. Adam, and T. Liu, "View-invariant probabilistic embedding for human pose", in *European Conference on Computer Vision.* Springer, 2020, pp. 53–70.