T.C. DOĞUŞ UNIVERSITY

INSTITUTE OF GRADUATE PROGRAMS

COMPUTER ENGINEERING

A New Approach Using Deep Learning Methodologies from Human Activity Recognition to Robot Grasping

Ph.D. Thesis by

Senem Tanberk

2014194003

Thesis Supervisor

Prof. Dr. Mitat Uysal

Thesis Co-Advisor

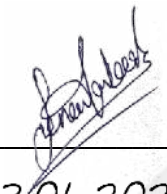
Dr. Dilek Bilgin Tükel

İstanbul, June 2020

DECLARATION

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university and institution. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references given.

Senem Tanberk

Signature : 

Date : 23.06.2020

ACKNOWLEDGEMENTS

This manuscript is the fruit of about three intense years, full of exciting and difficult discoveries regarding deep learning. It was an interesting and enjoyable journey, although challenging, which taught me a lot about deep learning.

Most importantly, I would like to thank to my thesis advisor Prof. Dr. Mitat Uysal for the continuous supports, incredible guidance, his patience, and motivation during this adventure. My deep gratitude also goes to my coadvisor, Dr. Dilek Bilgin Tükel, who spent her time to me and sharing all her valuable experience especially during conferences.

I would like to thank to Dr. Zeynep Hilal Kilimci for guidance to journal studies with her valuable insights. Furthermore, my sincere thanks also goes to the rest of my thesis committee: Prof. Dr. Selim Akyokuş and Dr. Aysun Güran, for their valuable comments and encouragement.

Finally, I would like to express my endless gratitude to my dear mother and father, and to my brother, who always support me throughout my life.

23.06.2020 Senem Tanberk

ABSTRACT

The research goal of this work is to develop an advanced deep learning model to analyse automatically human motion on videos and to present the prototype of a basic robotic imitation system that mimics the human movement supported by deep learning. For this purpose, we propose a hybrid deep model for human activity recognition, robotic simulator infrastructure for human imitation, and simple, intelligent, extensible and pluggable video analytic framework for integration.

First, we present a new hybrid deep learning model for human activity recognition in videos. We proposed a new 3-stream hybrid deep model with data fusion of 3D-CNNs fed by dense optical flow and LSTM fed by auxiliary information. We used SVM as a classifier. We generated 2 different datasets, namely the magnetic wall chess board video dataset (MCDS), and standard chess board video dataset (CDS). They consist of microvideos with duration of 5-6 second that contain meaningful movements by the chess player. We experimented hybrid deep model with two new datasets, the experimental results show remarkable performance compared to the state-of-the-art studies. The proposed hybrid deep model can be used in complex motion recognition tasks, thanks to its ability to fuse information with different characteristics such as motion, image, sound and text.

Second, we developed a robotic simulation infrastructure for chess-playing delta robot. We used V-REP virtual robot experiment platform by Coppelia Robotics. Generated robotic simulator can operate both standalone and by being controlled by an external system.

Finally, we designed a simple video analytic framework for human motion imitation system and used our hybrid deep learning model in this framework. We tested end to end system with a specific scenario in offline mode. In this way, we have developed an artificial intelligence supported human imitation system prototype to be used in the specific problem of imitating chess player by robot simulator. The robot simulator in the proposed system can imitate the chess player with motion primitive approach. As a result, we achieved an AI-powered, intelligent, extensible, and pluggable human imitation system prototype.

Keywords : Deep learning, optical flow, convolutional neural network, long short-term memory, human activity recognition, video classification, chess, delta robot, simulation, V-REP, video analytics framework, human imitation, motion primitive.

ÖZET

Bu çalışmanın araştırma amacı, videolardaki insan hareketlerini otomatik olarak analiz etmek için ileri bir derin öğrenme modeli geliştirmek ve derin öğrenme destekli temel bir robotik taklit sisteminin prototipini sunmaktır. Bu amaçla, insan aktivite tanıma için hibrit bir derin model, insan taklidi için robotik simülatör altyapısı ve entegrasyon için basit, akıllı, genişletilebilir ve pluggable bir video analitik çerçevesi öneriyoruz.

İlk olarak, videolardaki insan aktivitelerini tanıyacak yeni bir hibrit derin model sunuyoruz. Yoğun optik akış ve yardimci bilgilerin kaynastirildigi, 3D-CNN ve LSTM kombinasyonu ile gerçeklenen 3-akışlı yeni bir hibrit derin model oluşturduk. Hibrit derin modeldeki 3D-CNN'ler çoklu çerçeve ve yoğun optik akışla beslenir, LSTM ise yardımcı bilgilerle beslenir. Sınıflandırıcı olarak SVM kullandık. Manyetik duvar satranç tahtası video veri seti (MCDS) ve standart satranç tahtası video veri seti (CDS) olmak üzere 2 farklı veri seti oluşturduk. Bu veri setleri, satranç oyuncusu tarafından yapılan anlamlı hareketleri içeren 5-6 saniyelik mikrovideolardan oluşurlar. İki yeni veri seti ile hibrit derin modeli denedik, deneysel sonuçlar, teknoloji harikası diğer çalışmalara kıyasla dikkate değer bir performans gösterdi. Geliştirdiğimiz hibrit derin model, hareket ile görüntü, ses ve text gibi farklı özellikteki bilgileri aynı derin modelde kaynaştırabilme yeteneği sayesinde kompleks hareket tanımada kullanılabilir.

İkinci olarak, satranç oynayan delta robot için robotik bir simülasyon altyapısı geliştirdik. Coppelia Robotics tarafından geliştirilen V-REP sanal robot deney platformunu kullandık. Geliştirilen robot simülatörü hem bağımsız olarak hem de dış bir sistem tarafından kontrol edilerek çalışabilir.

Son olarak, insan hareket taklit sistemi için basit bir video analitik çerçevesi tasarladık ve hibrit derin öğrenme modelimizi bu çerçevede kullandık. Çevrimdışı modda belirli bir senaryo ile uçtan uca sistemi test ettik. Bu şekilde, satranç oyuncusunun robot simulatörü ile taklit edilmesi problemi özelinde kullanılacak bir yapay zeka destekli insan taklit sistemi prototipi geliştirmiş olduk. Önerilen sistemdeki robot simülatörü, satranç oyuncusunu hareket ilkel yaklaşımı ile taklit edebilir. Sonuç olarak, yapay zeka destekli, akıllı, genişletilebilir ve pluggable bir insan taklit sistem prototipi elde ettik.

Anahtar Kelimeler: Derin öğrenme, optik akış, evrişimli sinir ağı, uzun kısa süreli bellek, insan aktivite tanıma, video sınıflandırma, satranç, delta robot, simülasyon, V-REP, video analitik çerçevesi, insan takliti, hareket ilkeli.

CONTENTS

TABLE LIST

FIGURE LIST

ABBREVIATIONS

3D-CNN	**:**	3D Convolutional Neural Networks
AI	**:**	Artificial Intelligence
API	**:**	Application Programming Interface
CDS	**:**	Standard Chess Board Video Dataset
CNN	**:**	Convolutional Neural Networks
CNN-OF	**:**	Convolutional Neural Network Model Combined with Dense Optical Flow
DL	**:**	Deep Learning
FC	**:**	Fully Connected
GPU	**:**	Graphical Processing Unit
HAR	**:**	Human Activity Recognition
HDM	**:**	Flexible Hybrid Deep Model
HMI	**:**	Human Motion Imitation
HMM	**:**	Hidden Markov Model
LSTM	**:**	Long short-term Memory
MCDS	**:**	Magnetic Wall Chess Board Video Dataset
ML	**:**	Machine Learning
OF	**:**	Optical Flow
ReLU	**:**	Rectified Linear Units
RGB	**:**	Red Green Blue
RNN	**:**	Recurrent Neural Networks
SSD	**:**	Single Shot Detector
SVM	**:**	Support Vector Machine

1.INTRODUCTION

The automatic analysis and interpretation of human movements in videos is one of the main computer vision challenges. It has attracted the attention of researchers from late 1970s to the present. There are hundreds of potential applications for automatic video analysis subject : Activity recognition, smart home and building systems, security applications(anomaly detection in crowded places such as streets, malls, airports etc), analysis of customer behavior in shopping areas, assisted technologies(elderly support system, etc), clinical monitoring, fall detection(to identify falls and raise alarms), telepresence systems, augmented reality, robotic vision, robotic imitation and teleoperation systems, video annotation, video summarization and indexing, interactive game applications.

Video-based human action recognition is an active research area with many effective application areas such as healthcare, surveillance, entertainment, and military systems. The purpose of human activity recognition is to automatically analyze activities in a video (an image frame sequence). The primary purpose of this system is to classify videos that are segmented to include a single human activity by activity category. In general cases, there is continuous recognition of human activities in a long video sequences. These cases include determining the categories, start and end times of all activities in the video.

Human activities are categorized as gestures, atomic actions, human-to-object or human-to-human interactions, group actions, behaviors and events, depending on their complexity (Vrigkas, Nikou & Kakadiaris, 2015). In addition, a bottom-up approach is followed by many researchers to recognize human activity. The main stages of these systems are as follows: feature extraction, activity learning and classification, activity recognition and segmentation (Cheng, Wan, Saudagar et al., 2015; Agahian, 2018). Although there have been many studies on this subject, it is still an active research area with challenging research issues such as activity analysis for assisted living in smart homes (Agahian, 2018).

Chess is one of the most popular board game in the world. Chess has always been a challenge for computer hardware designers and software developers. It is an attractive topic for research on human-machine interaction. It is necessary to produce solutions in

areas such as image processing, coordination and strategy establishment to solve the chess problem.

Robotic applications try to solve the problem of learning mapping, also called policy, between world and actions. Human motion imitation and teleoperation are examples of this policies. In recent years, human motion imitation by robots is an active research field. Robots need to imitate human movements to increase their autonomous behaviors. Imitation and teleoperation in articial intelligence are great interest for many reasons on domains such as assisted technologies, robot-assisted surgery, space systems, remote robot controller, search and rescue robot, and interactive game technology.

AI-powered video analytics is capable of automatic video analysis. It can integrate video classification for human activity recognition and human motion imitation solutions in a single framework. It can be used for video segmentation, indexing, retrieval, smart compression, and content analysis also.

In this thesis, human activity recognition task will be achieved using 3D-CNNs, LSTM, and optic flow. We aim to classify human activities in chess for Forward1, Forward2, DiagonalLeft and DiagonalRight via SVM. Robotic chess simulator infrastructure will be developed by V-REP. Finally, deep learning video analytics framework will be proposed for human motion imitation, and then robotic chess simulator will show an example of the task by imitating the human.

1.1 Contribution of the Thesis

The contribution of this thesis can be grouped in three main categories, a hybrid deep model for human activity recognition, V-REP robotic chess simulator infrastructure and deep learning video analytics framework for human motion imitation.

First, for the hybrid deep model, two new datasets which contain microvideos with duration of 5-6 seconds are collected. Videos consist of meaningful chess moves by the chess player. Then a new flexible multi-stream hybrid deep model is introduced. It is a novel combination of 3D-CNNs fed by both multiple frames and optical flow and long LSTM fed by auxiliary information over video frames. We used data fusion approach with concatenate to combine multi-streams. The hybrid deep model is used for the purpose of human activity recognition. The proposed hybrid deep model can be used to recognize complex human activity tasks, because it can fuse information with different

characteristics such as motion, image, sound and text. Furthermore, it can combine single camera videos with images from a second camera or data from various sensors. We introduced a new hybrid deep model that can be used for the task of human activity recognition in the video analytics framework.

Second, we introduced a robotic chess simulator infrastructure which can operate both standalone and by being controlled by an external system.

Finally, we presented an early prototype of deep learning video analytics framework for human motion imitation. This is the integration point of hybrid deep model and robotic simulator. In this framework, video is analysed with deep learning approaches and then human activity recognition task is completed. The robotic chess simulator takes commands via service API from video analytics framework. It can imitate the chess player with motion primitive approach. We showed an early prototype of system end to end for AI-based video analytics framework from human activity recognition to human motion imitation.

1.2 Thesis Impact and Potential Application Domains

Video processing and human activity recognition is the most challenging tasks in computer vision area. Proposed multi-stream hybrid deep model and DL video analytics framework can be easily applied to the following artificial intelligence fields :

- Smart surveillance.
- Autonomous driving.
- Robotic : Robot vision and robot control.
- Smart home systems.
- Healthcare.
- Video content analysis, summarization, indexing and retrieval, adding automatic captioning to videos.
- Video conferencing.
- Sport analysis and exercise training.
- Human-computer interaction.
- Character animation.
- Augmented reality.

Sample applications are below :

- Production control systems.
- Monitoring elderly, disabled, or injured people, and children.
- Unmanned aerial, ground and underwater vehicles, autonomous cars and vehicles
- Military applications.
- Traffic monitoring and management, parking management.
- Crowd management in airport, stadiums, malls and hospitals, detecting movement in unwanted or prohibited directions.
- Football Analytics : First step is to detect objects such as person, vehicles, faces, football field, animals in video. Second step is to find out the detected objects actions and activities and check rule violation.

1.3 Thesis Organization

Thesis consists of eight chapters and each chapter contains valuable information regarding this thesis. The organization of this thesis is briefly organized as follows :

- Chapter 2 gives a literature review regarding CNN, LSTM, SVM, human activity recognition, human motion imitation and video analytics framework. It provides an overview of existing state-of-the-art studies for the present work.
- Chapter 3 presents research methodologies regarding human activity recognition : 3D convolutional neural network, long short-term memory, optical flow, and SVM classifier respectively.
- Chapter 4 provides an overview for the research methodology regarding three approaches of human motion imitation.
- Chapter 5 is dedicated to the problem of video classification containing chess videos to address human activity recognition and introduces a new flexible hybrid deep model combining 3D-CNNs and LSTM.
- Chapter 6 describes our robot simulator infrastructure implemented by V-REP.
- Chapter 7 illustrates an early prototype of deep learning video analytics framework for human motion imitation. It uses motion primitive-based approach for human imitation.
- Chapter 8 is the final chapter. In this chapter, there are conclusions reached at this thesis and recommendations for future work chapter.

2. RELATED WORK

This section provides a brief summary of deep learning models related to human activity recognition, optical flow, video classification, and smart video surveillance in the literature.

The traditional approach for video classification (Wang, Ullah, Klaser et al., 2009; Liu, Luo & Shah, 2009; Sivic & Zisserman, 2003; Niebles, Chen & Fei-Fei, 2010) has three phases: First, visual features in videos are extracted densely (Wang, Kläser, Schmid et al., 2011), or sparsely (Laptev, 2005; Dollár, Rabaud, Cottrell G. et al., 2005). Next, fixed-sized video-level description is obtained. Lastly, a classifier like SVM is used to get video classes. In (Karpathy, Toderici, Shetty et al., 2014), the authors are inspired by CNNs (LeCun, Bottou, Bengio et al., 1998) to replace all three phases with a single neural network. They model videos with convolutional neural networks in a very similar way to model images of CNNs for video classification. CNNs are directly applied to the multiple frames. Experiments are carried out on the UCF-101 action recognition dataset. They apply video based deep learning techniques with CNNs to analyze videos containing human motions and observe significant performance improvements compared to the baseline model.

Simonyan and Zisserman (Simonyan & Zisserman, 2014) propose a deep video classification model to incorporate spatial and temporal information based on ConvNets. Two-stream CNNs are employed in the proposed approach. One stream takes RGB image frames as input while the other stream takes stacked optical flow by computing. Standard action recognition datasets, namely UCF-101 and HMDB-51, are evaluated and tested to show the contributions of their approach. Despite limited training data, they demonstrate that the proposed deep model achieves remarkable classification performance. RNNs are also considered for video-based HAR. In (Arif, Wang, Hassan and et al., 2019), Arif et al. combine 3D-CNN and LSTM for human activity recognition task effectively. They introduce a 3D-Conv-based model and its iterative training method to integrate distinctive information in a video into motion maps. They use the LSTM encoder/decoder to obtain video level representations to help recognize complex frame-to-frame hidden sequential patterns for video classification. They test the proposed model with benchmark datasets and achieve a promising performance. In (Sun, Wang & Yeh, 2017), video classification and captioning is implemented with 3D-CNN and LSTM networks in one architecture.

The authors explore methods and architectures to understand how to categorize and caption videos, automatically. Experimental results demonstrate that the proposed multi-task architecture exhibits significant advantages for complex video captioning models. With this architecture, the image captioning dataset can be trained in the first stage, and then pre-trained weights can be used during the video captioning model training phase. The proposed architecture gives better results than end-to-end training on video captioning datasets and saves training time.

In (Qian, Mao, Xiang et al., 2010), a system framework is presented to recognize human activities in videos by an SVM multi-class classifier with a binary tree architecture. Hierarchical classification is introduced and multiple SVMs are aggregated to recognize multiple actions. Each SVM in the multi-class classifier is trained, separately. The proposed multi-class SVM model is applied to both a home-brewed activity dataset and Schüldt's public dataset. Experimental results demonstrate that the usage of the proposed architecture yields exceptional identification performance. In (Wang, Kläser, Schmid et al., 2011), dense optical flow trajectories are employed for action recognition tasks. Wang et. al. propose an approach to describe videos by dense trajectories inspired by the success of dense sampling in image classification. They introduce a novel descriptor that is robust to camera motion based on motion boundary histograms. The proposed descriptor model outperforms state-of-the-art descriptors. The KTH, YouTube, Hollywood2, and UCF sports datasets are employed to demonstrate the efficiency of the proposed model in the experiments. The works reported in (Laptev & Lindeberg, 2003) and (Dollár, Rabaud, Cottrell G. et al., 2005) are also examples of optical flow based approaches. In (Latah, 2017), 3D CNNs and SVM are used to recognize human action in videos. First, 3D CNNs are used to extract spatial and temporal features from consecutive video frames. The SVM approach is then used in order to classify videos. The proposed architecture is trained on the KTH action recognition dataset. Experimental results show that the proposed architecture achieves significant classification performance.

Depending on the structure of the deep learning network, the main representative works can be summarized as in (Zhang, Zhang, Zhong et al., 2019) as methods based on two-stream convolutional networks, those based on 3D convolutional networks , and those based on long short-term memory (LSTM) . In a two-stream convolutional network,

the optical flow information is calculated from the image sequence. The image and optical flow sequence are respectively used as the input to the two convolutional neural networks (CNNs) during the model training process. Fusion occurs in the last classification layer of the network. The inputs of the two-stream network are a single-frame image and a multi optical flow frame image stack, and the network applies 2D image convolution. In contrast, a 3D convolution network regards the video as a 3D space-time structure, and uses a 3D convolution method to learn human action features**.**

In (Liu, Zhang & Tian, 2016) study, Liu et al. proposed a straightforward action recognition framework containing three components: $3D^2CNN$, joint vector calculation, and classifiers fusion. $3D^2CNN$ in the proposal framework learns high-level features from depth video. Second stream in the framework, joint feature vector is calculated according to joint skeleton. Then, data fusion is implemented with SVM classification results from $3D^2CNN$ and joint vector to recognize action. In (Liu, Shahroudy, Xu & Wang, 2016), Liu et al. extended RNN based 3D action recognition to spatio-temporal domain. They designed a new ST-LSTM network to analyse the 3D location of individual joints in video images. Wang et al. in the (Wang, Gao, Song & Shen, 2017) study introduced a general framework for video action recognition. They presented end-to-end pipeline for saliency-aware 3-D CNN with LSTM, next a time series pooling layer, and then softmax layer for activity prediction on videos. In (Wang, Gao, Wang, Sun et al., 2018) study, Wang et al. proposed end-to-end framework via fusion of two stream 3D convnet for human action recognition. Their model takes both spatial and motion features. Their model includes a combination of 3D convnet and LSTM to represent video. Their approach improved other CNN-based video analysis algorithms. In (Chuankun, Hou, Wang & Li, 2018), Chuankun et al. presented a new method for action recognition using multiple deep networks. They used LSTM and CNN for temporal information and spatial information, multiviews are generated from a skeleton sequence. Multiply-score fusion is applied for recognition task. Keçeli in (Keçeli, 2018) proposed an action recognition method for single or dyadic actions. First, he generated an isosurface from the depth sequence, and extracted images from different angles of it. Second, he used CNN to extract high-level features from depth video. And then, he used random forest algorithm for classification. In (Keçeli, Kaya & Can, 2018), Keçeli et al. developed a framework for action recognition to be used in both single-person actions and dyadic interactions. 2D representations in pre-trained CNNs

and 3D volume representations in 3D-CNN of videos are combined and then ranked by using Relieff algorithm to choose strong features. Then, SVM classifier is used fot classification. Wang et al. (Wang, Wang, Song & Li, 2017) proposed a method for gesture recognition with heterogeneous networks including Convolutional Neural Networks, 3D convolutional neural network, and Convolutional LSTM Networks. This heterogeneous networks can learn effectively spatiotemporal features. In (Song, Yuyu, Gao, Li et al., 2017) study, Song et al. presented a multimodal stochastic RNN framework including S-LSTM and M-LSTM for video captioning. Song et al. in (Gao, Li, Song & Shen, 2019) introduced a novel hLSTMat encoder-decoder framework with hierarchical LSTMs for visual captioning such as image and video captioning.

The number of applications and systems for smart video surveillance has developed rapidly in recent years. Here, we examine smart video surveillance frameworks for academic research areas and industrial application areas. Hu et al. (Hu, Xue, Mei et al., 2017) presented a smart Video and image analysis framework consisting of four components: a set of evaluation tools, criteria, datasets and an operation system. In reference (Ostheimer, Lemay, Ghazal et al., 2006), distributed real-time video processing is developed, it is used for objects and event detection, but it is limited to few applications. Hossain (Hossain, 2014) introduced a cloud-based framework for the surveillance system, it consists of segmentation, motion detection, tracking activity and recognition. Numerous industrial companies have developed a cloud-based distributed video processing system. Eagle Eye Networks (https://www.eagleeyenetworks.com access date: 30.03.2020) has a real-time cloud-based video surveillance system. They presents RESTful APIs which has capability of recording, indexing and storing streaming videos. Intelli-Vision (https://www.intellivision.com access date: 30.03.2020) provides AI-based video analysis for smart cameras. They provide lots of services, such as vehicle counter, face recognition, traffic analytics, video search, and video summary. Google Vision API (https://cloud.google.com/video-intelligence access date: 30.03.2020) provides a video management and retrieval framework. They presents APIs also, but no intelligent video surveillance system support. IBM Intelligent Video Analytics (https://www.ibm.com/cloud access date: 30.03.2020) offers batch and real-time video processing, but they don't have API based services. In (Uddin, Alam, Tu et al., 2019),

they proposed a distributed and smart video analytics framework. Architecture in this framework is a distributed, robust, pluggable, layered, APIs and cloud based.

3. RESEARCH METHODOLOGY : HUMAN ACTIVITY RECOGNITION

3.1. Feature Extraction

3.1.1. 3D Convolutional Neural Networks

CNN (LeCun & Bengio, 1995) is a feed-forward neural network, it consists of neurons which have learnable weights and biases. It has convolution operation in it at least one of the layers. It can take an input with multidimensional array which include 2D input image, speech signal, or 1D time series data. Because of all these advantages, CNNs can be used in many applications such as image recognition, video recognition, natural language processing, speech recognition, handwriting recognition, media recreation, recommender systems, and so on. CNNs became popular after 2012. Since then, many high-tech companies started to use CNNs. Amazon uses CNNs for product recommendations, Google uses them for photo search, Facebook uses them for automatic tagging, Apple uses them for face detection, and Microsoft uses them for computer vision (Dev, 2017).

CNNs is not a new concept. LeCun et al in 1998 (LeCun, Bottou, Bengio et al., 1998) developed a CNN model working well to recognize hand-written digit (Dev, 2017).

At that time, CNNs were not able to work well with higher resolution images due to hardware and memory constraints, and there is no also large datasets so they lost their popularities (Dev, 2017).

The computational power increased with CPUs and GPUs, so it became possible to process large-scale datasets. Krizhevsky et al (Krizhevsky, Sutskever & Hinton, 2012) used CNNs to classify images with large dataset in the field of computer vision. Their model is called AlexNet and became very popular over the next few years (Dev, 2017).

A CNN is composed of four main types of layers : Convolutional layer, Rectified Linear Units layer, Pooling layer, and Fully-connected layer.

A regular CNN could have architecture below as seen Figure 3.1:

[INPUT – CONV – RELU – POOL – FC]

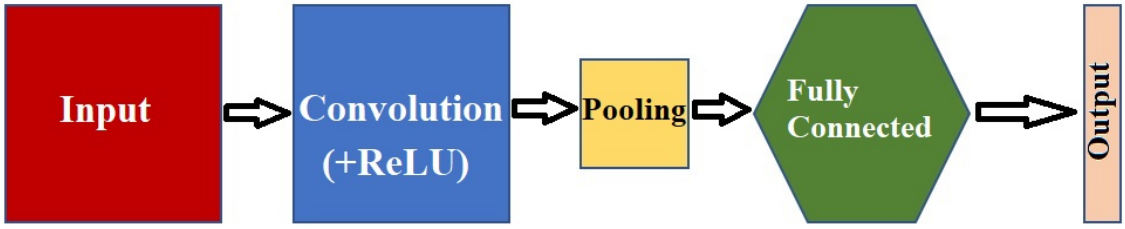


Figure 3.1 A regular CNN architecture (Dev, 2017).

A classic deep neural network will have structure below:

Input->Conv->ReLU->Conv->ReLu->Pooling->ReLU->Conv->ReLu->Pooling->Fully Connected (Dev, 2017)

AlexNet is a perfect example for this kind of architecture. The architecture of AlexNet can be shown in Figure 3.2.

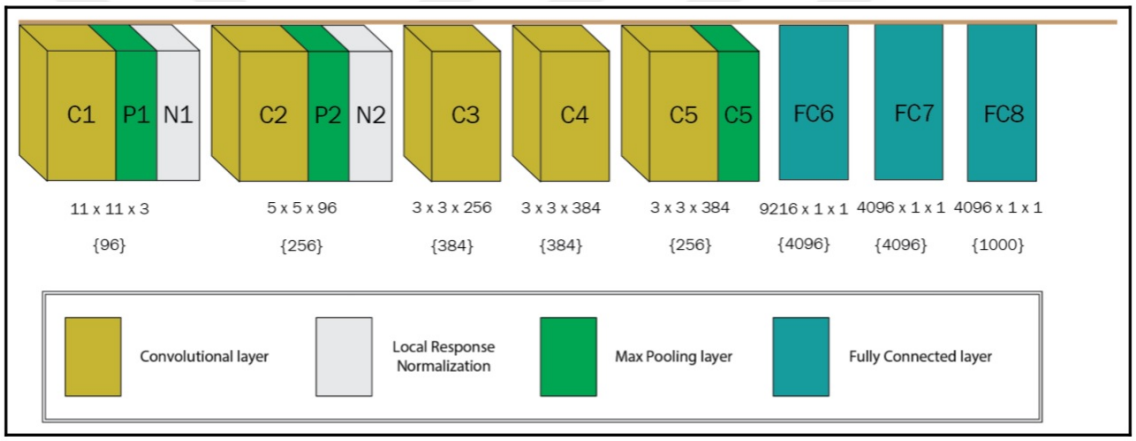


Figure 3.2 The architecture of AlexNet. It illustrates the depth and weight of the AlexNet. The number inside the curly braces is the number of filters (Dev, 2017).

A CNN can find dependencies between the spatial and temporal features in an image successfully. The architecture fits better to the image dataset because it reduces the number of parameters involved and ensures reusable weights. It can be trained to understand images better (Saha, 2018). Image classification processes in CNNs, as shown in Figure 3.3, pixels construct figures of edge combination, then these figures combine to form object parts, and finally object parts combine to form objects (LeCun, Bengio & Hinton, 2015).

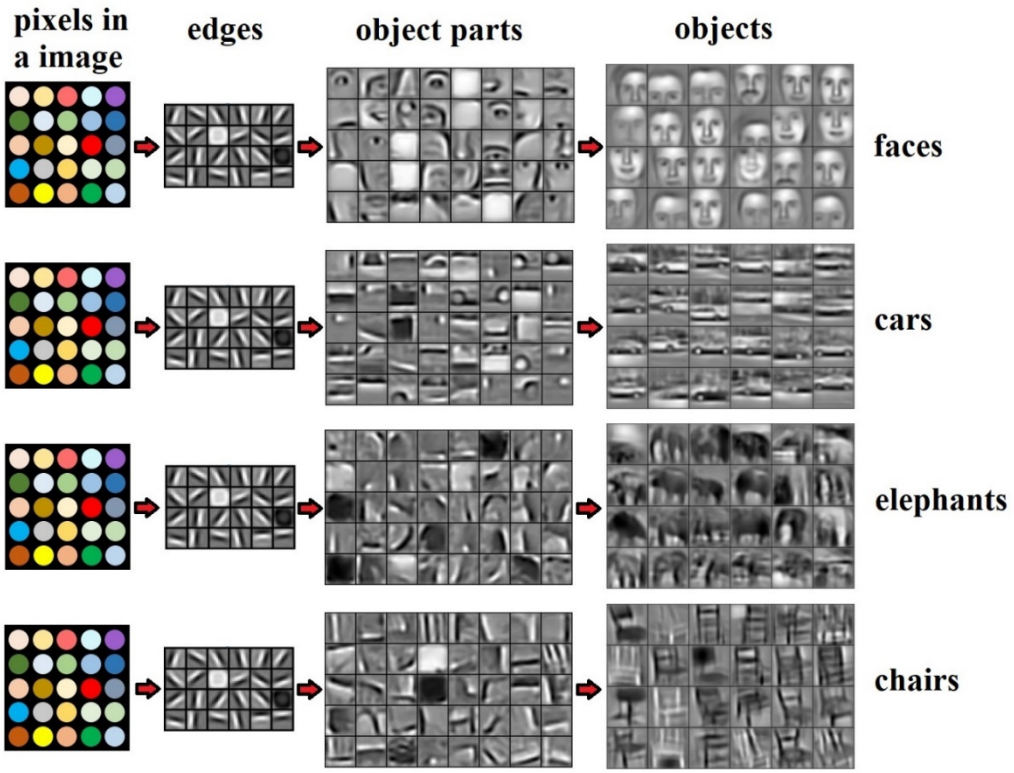


Figure 3.3 Representations about the object in various layers for specific object categories (Lee, Grosse, Ranganath & Ng, 2009).

Input Image :

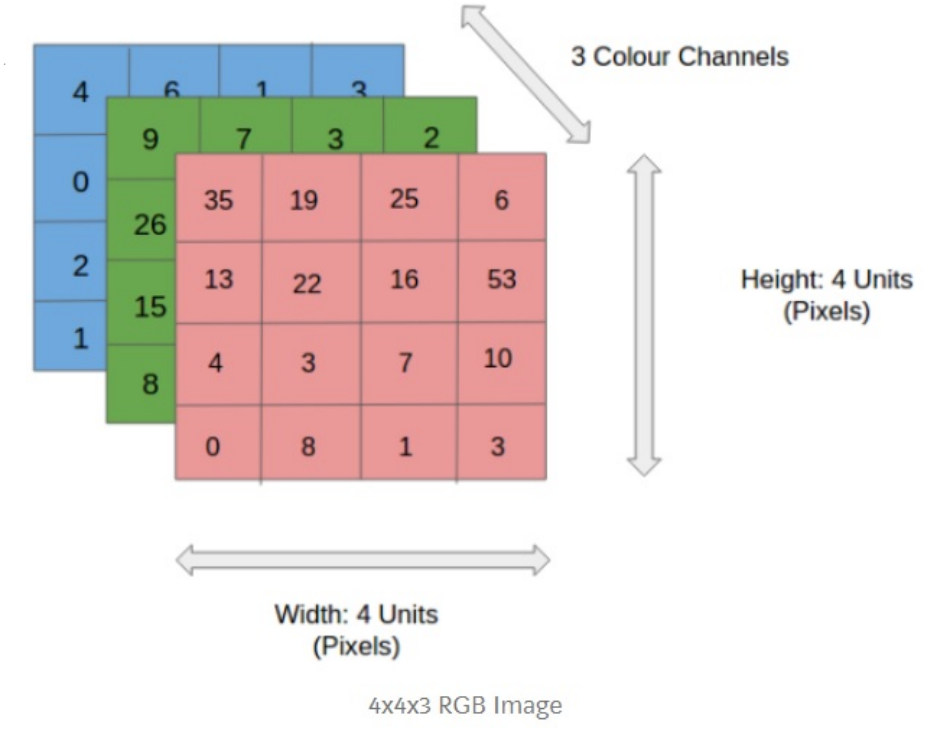


Figure 3.4 RGB image (Saha, 2018).

RGB image here : In the Figure 3.4 (Saha, 2018), there is an RGB image by three color planes : Red, Green, and Blue. So images can reach a huge dimensions and cause computational intensive. CNNs ensure reducing the images into an simpler form to process easily without losing features which are important to make a good prediction. So this makes it possible to process massive datasets (Saha, 2018).

For the CNN model to be successful, image resolution in the input layer is important. High resolution in the input image increases the success of the network. But it requires high memory and it causes longer test time. Low resolution in the input image requires less memory and it causes less test time. But it causes low success. Optimum resolution should be chosen to considering memory constraint and success of the network (İnik & Ülker, 2017).

Convolution Layer - The Kernel :

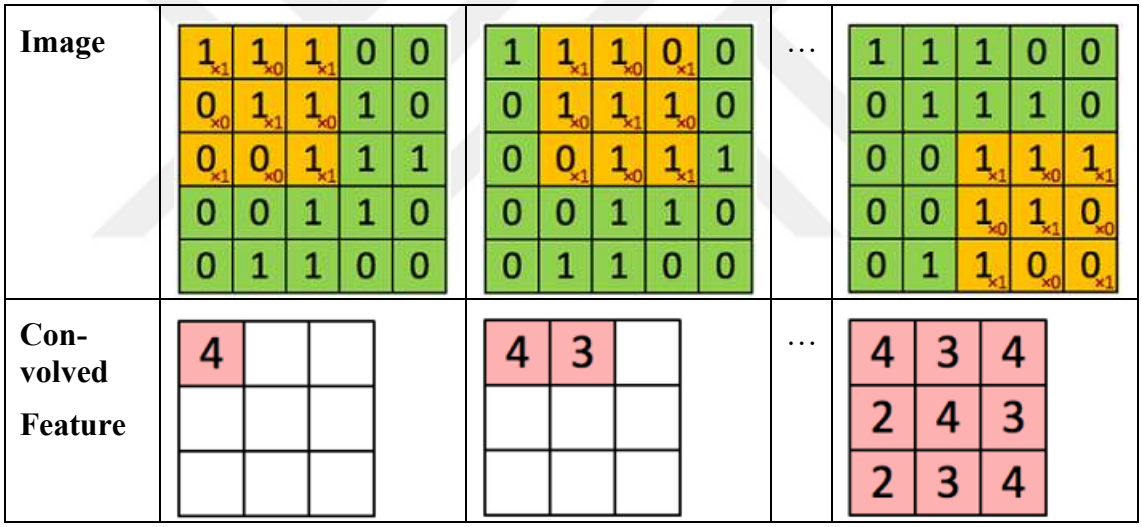


Figure 3.5 5x5x1 image convolution, kernel is 3x3x1 (Saha, 2018).

Kernel/Filter is the first part of convolution layer for carrying out the convolution operation. In the Figure 3.5, K is selected as a 3x3x1 matrix. The Kernel shifts 9 times because Stride Length is equal to 1 (Saha, 2018).

Usually, the first convolution layer captures the low-level features such as edges. By adding layers to the architecture , it is possible to understand high-level features of images in the dataset (Saha, 2018).

Valid Padding or Same Padding can be applied to control the size of the output volume. Valid Padding reduces output dimension compared to the input. Same Padding increases output dimension or leaves the same (Saha, 2018).

ReLU (Rectified Linear Units) Layers :

After performing convolution layer, an activation layer is added to current output. The activation function, namely the rectifier, is defined as follows (Dev, 2017):

$$f(x) = \max(0, x) \qquad \text{(Dev, 2017)} \qquad \textbf{(3.1)}$$

Rectified Linear Unit (ReLu) is used as an activation function on modern CNNs. Earlier, some non-linear functions such as tan h or sigmoid were used as an activation function, but later, ReLU layers were preferred because they work better than the others (Dev, 2017).

Pooling Layer :

The third stage of CNNs is pooling layer. After applying Rectified Linear Units, then pooling layer is applied (Saha, 2018).

The pooling layer is used to reduce the spatial size of the convolutional features. So, it decreases the computational power by reducing dimensions. Furthermore, it helps to extract dominant features, so it increases process effectiveness (Saha, 2018).

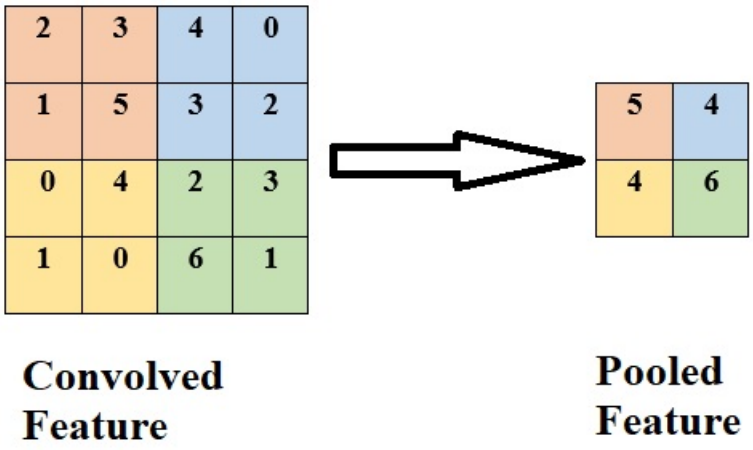


Figure 3.6 Pooling Layer with 2x2 max pooling (Saha, 2018).

There are two types of pooling functions called Max Pooling and Average Pooling. Max Pooling returns the maximum value from a rectangular portion by taking a filter(generally of size 2x2), and then strides(generaly same length, 2). On the other hand, Average Pooling returns the average value from a rectangular portion (Saha, 2018).

Classification - Fully Connected Layer (FC Layer) :

The final layer of a CNN is a fully connected layer. The fully connected layer takes this input and outputs an N dimensional vector, where N is the number of different classes present in the initial input datasets. This layer takes input from previous layer and outputs an N dimensional vector, N means different classes in the initial input datasets. It identifies the specific feature that is mostly correlated to a particular class. For example, for predicting whether an image is a cat or bird, it will take high-level features such as legs or wings (Saha, 2018).

CNNs generally refer to 2 dimensional type of them. However, there are two other types of CNNs namely 1D and 3D CNNs (Verma, 2019).

2 dimensional CNNs :

This is the standard Convolution Neural Network mentioned in previous chapter. This is generally used in image data. As shown in the Figure 3.7, the kernel slides along 2 dimensions on the image. So because of that, it is called 2 dimensional CNN (Verma, 2019).

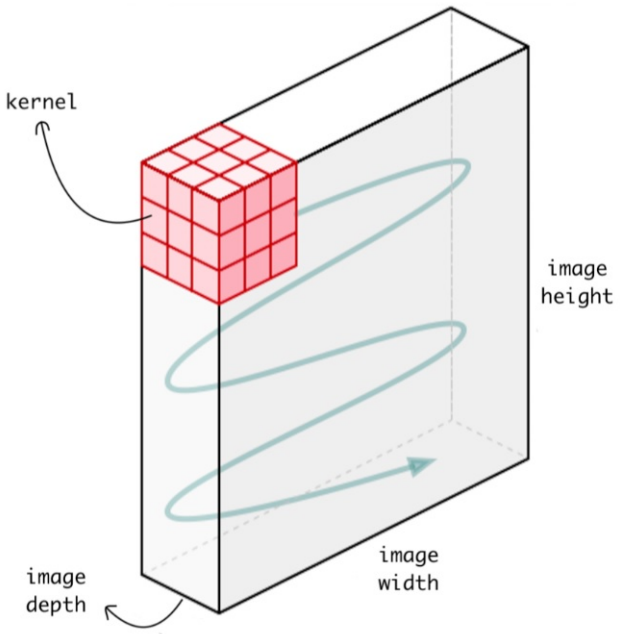


Figure 3.7 The kernel slides over 2d image (Verma, 2019).

Argument input_shape represents (height, width, depth) of the image. Argument kernel_size represents (height, width) of the kernel. Kernel depth is the same as the depth of the image such as 3 colour channels(RGB) (Verma, 2019).

1 dimensional CNNs :

The kernel slides along 1 dimension on the data. 1D CNNs are used for time-series data such as audio(sound), text, accelerometer, and sensory data (Verma, 2019).

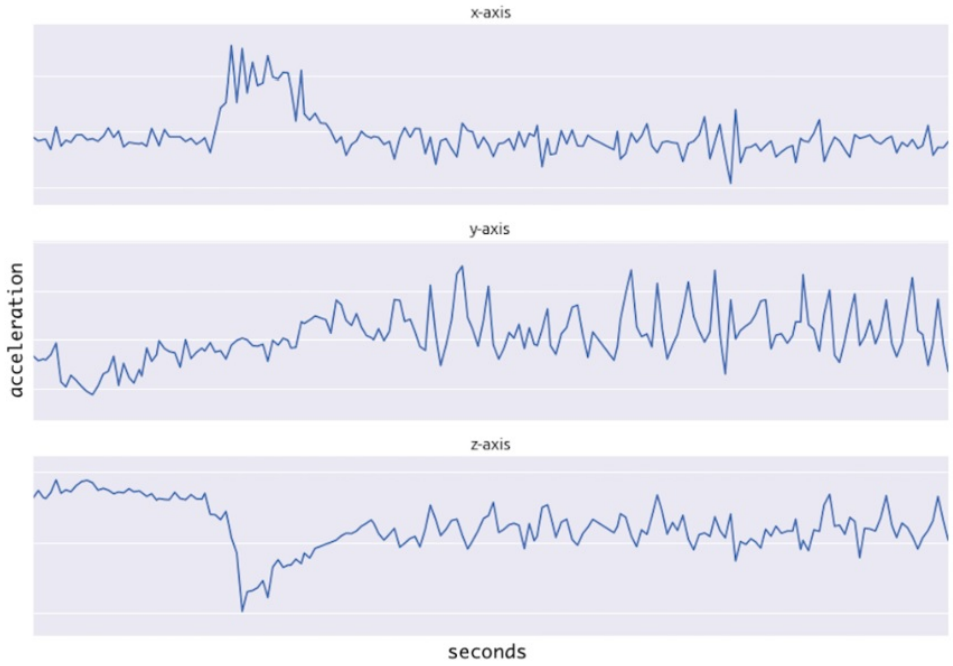


Figure 3.8. An accelerometer time series data (Verma, 2019).

This data is taken from an accelerometer from a wearable sensor in the arm. Acceleration are measured by 3 axes. So 1D CNN can perform activity recognition by using accelerometer data, such as standing, walking, jumping etc. This data has 2 dimensions. One of them is time-steps and other one is acceleration in 3 axes. Argument kernel_size is 5, it represents the width of the kernel. Kernel height is 3 and it represents 3 axes (Verma, 2019).

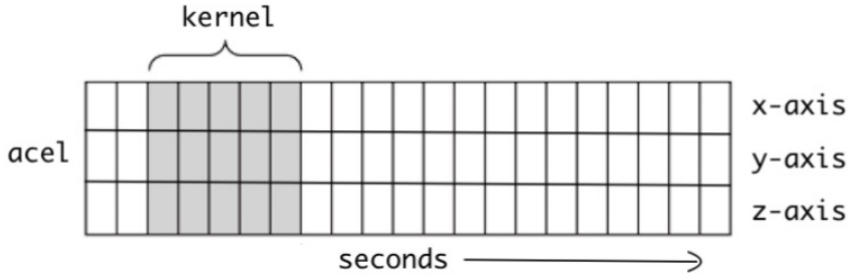


Figure 3.9 The kernel slides over 2d image (Verma, 2019).

3 dimensional CNNs :

The kernel slides along 3 dimensions (height, length, depth) on the data and produce 3D activation maps. 3 dimensional CNNs can be used to identify moving and 3D images, such as security cameras videos and cancerous tissue medical scans (https://missinglink.ai/guides/convolutional-neural-networks/understanding-3d-cnn-uses access date: 01.03.2020). They are generally used with 3D image data such as Magnetic Resonance Imaging data or Computerized Tomography (CT) data. MRI is generally used to examine brain and internal organs. 3d CNNs are used to classify this types of medical data and extract features from them (Verma, 2019).

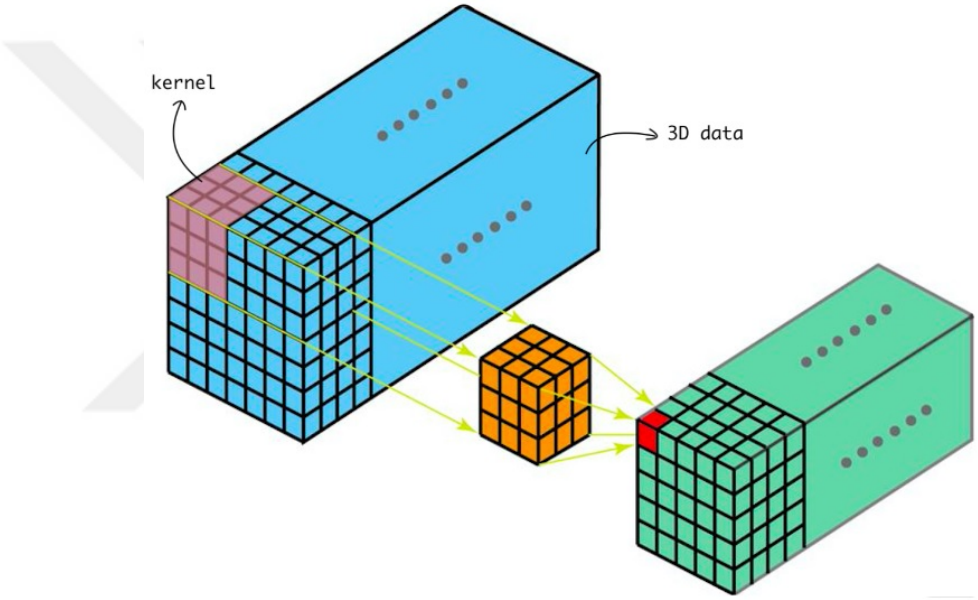


Figure 3.10 The kernel slides over 3d data (Verma, 2019).

A 3D image has 4-dimensional data, the fourth dimension represents 3 colour channels(RGB). Argument kernel_size represents (height, width, depth). The fourth dimension of the kernel represents 3 colour channels(RGB) (Verma, 2019).

The 3D activation map produced on 3D CNNs enable to analyze data which have temporal or volumetric context. Thanks to this ability, 3D CNNs ensure for activity recognition and evaluation of medical imaging (https://missinglink.ai/guides/convolutional-neural-networks/understanding-3d-cnn-uses access date: 01.03.2020).

During human activity recognition process, objects positions in a video (a sequence of 2D images) are being analyzed and then the context is being classified to either interpret or predict object movement. Activity recognition is used in various applications, such as smart homes for assistive technologies, surveillance or security systems and virtual reality systems to create decentralized meeting spaces. Human activity recognition process is complicated. Generally, a two-stream method which have spatial and temporal data is used independently to analyze data at the convolutional and pooling layers, and then two streams joint on Fully Connected Layer (https://missinglink.ai/guides/convolutional-neural-networks/understanding-3d-cnn-uses access date: 01.03.2020).

3D CNNs are used in medical scans for detection, diagnosis, and building patient-specific devices. Slices of the depth of the tissue are captured during medical imaging and then evaluated. But there is need to combine these static images in a context to identify cancerous cell and evaluate arterial health and map brain tissue. So 3D CNNs enable this type of processing (https://missinglink.ai/guides/convolutional-neural-networks/understanding-3d-cnn-uses access date: 01.03.2020).

3.1.2. Long short-term memory (LSTM)

Recurrent Neural Network (RNN) is a type of neural network that has an internal memory. The output of the previous step is sent to the current step as input. RNN is recurrent because it performs the same function to all inputs and the output of current input depends on past. Produced output is sent back into the recurrent network. The current input and the output of previous input are considered on RNNs for making decision. All the inputs in RNNs are related to each other. RNNs have hidden layers which remember some information. So RNNs have memory located in the hidden state to process sequences of inputs (Mittal, 2019).

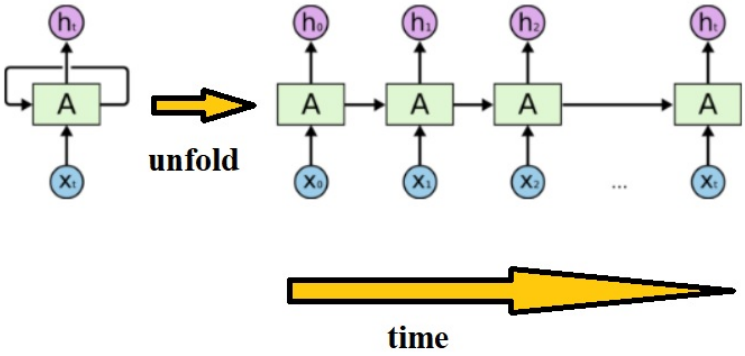


Figure 3.11 RNN structure being unfolded into a full network (Mittal, 2019)

A RNN is a deep neural network with indefinite number of layers when unfolded in time. The main function of RNN layers is to bring in memory, it is not a hierarchical processing. RNNs can receive inputs at each time step, and compute outputs at those intervals. In RNNs, new information is integrated into layers with iteration, and the network can be updated with this information. During the training phase, kept information and rejected information should be learned by the recurrent weights. This feature is the primary motivation for a special type of RNNs, called Long short-term memory (LSTM) (Dev, 2017).

RNNs can be used for time-series data, sequence of characters, and sequence of words. RNNs can be used in applications, such as connected handwriting recognition, speech recognition, word embedding, natural language processing, language translation, conversation modeling, image captioning, etc (Mittal, 2019).

RNNs have some disadvantages, such as gradient vanishing and exploding problems (Mittal, 2019).

Long short-term memory network (LSTM) is a type of RNN to solve vanishing gradient issues of RNNs (Greff, Srivastava, Koutnik and et al., 2017; Kent & Salem, 2019). In the mid-90s, German researchers Sepp Hochreiter and Juergen Schmidhuber (Hochreiter & Schmidhuber, 1997) proposed an updated version of RNNs, namely Long short-term memory (LSTM) to protect against the exploding or solve vanishing gradient problem (Dev, 2017). LSTMs provide better performance compared to RNNs. So they enable to solve vanishing gradient problem. LSTMs can be used for classification, process and prediction of time series with unknown duration (Mittal, 2019; https://developer.nvidia.com/discover/lstm access date: 01.03.2020; Greff, Srivastava, Koutnik and et al., 2017; Kent & Salem, 2019).

LSTMs have ability to learn long term dependencies in data. So they can be used in applications, such as language modeling and translation, speech modeling, speech synthesis, speech recognition, audio and video analysis, sequence prediction, handwriting recognition and generation, and protein secondary structure prediction (Mittal, 2019; https://developer.nvidia.com/discover/lstm access date: 01.03.2020).

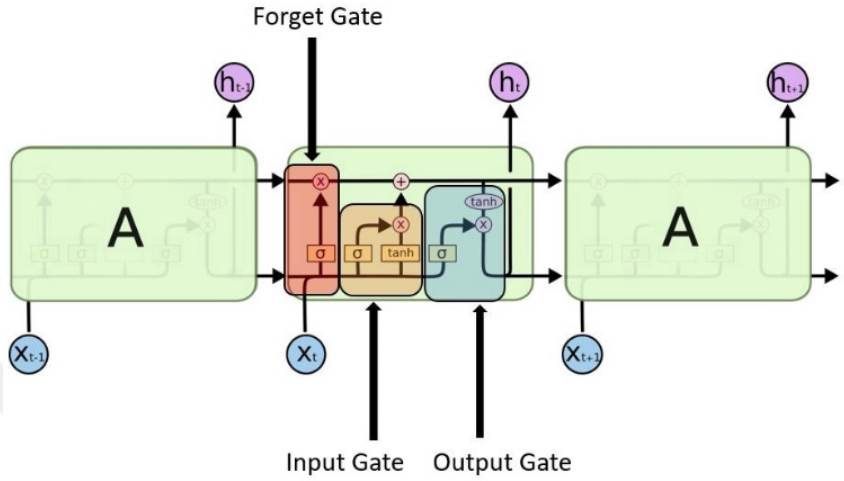


Figure 3.12 LSTM gates (Mittal, 2019).

LSTMs are composed of memory cells which can store information for long time (Dev, 2017). A simple LSTM model has a cell, an input gate, an output gate, and a forget gate.

1) Input gate :

It decides which input value will modify the memory. **Sigmoid** function chooses which values to let through **0,1**. **Tanh** function determines weight of the values ranging from **-1** to **1** (Mittal, 2019).

$$i_t = \sigma\left(W_i.[h_{t-1}, X_t] + b_i\right) \quad \textbf{(3.2)}$$

$$C_t = tanh\left(W_c.[h_{t-1}, X_t] + b_c\right) \quad \textbf{(3.3)}$$

2) Forget gate :

Sigmoid function decides details to be discarded from the block. It consideres the previous state($\mathbf{h_{t-1}}$) and the content input($\mathbf{X_t}$). It generates a number between **0** and **1** (0 means omit, 1 means keep) for numbers in the cell state $\mathbf{C_{t-1}}$ (Mittal, 2019).

$$f_t = \sigma\left(W_f.[h_{t-1}, X_t] + b_f\right) \quad \textbf{(3.4)}$$

3) **Output gate :**

Output is determined by the input and the memory of the block. **Sigmoid** function determines values to let through **0,1**. **Tanh** function determines weight of the values by deciding their level of importance ranging from **-1** to **1** (Mittal, 2019).

$$o_t = \sigma\left(W_o.[h_{t-1}, X_t] + b_o\right) \quad (3.5)$$

$$h_t = o_t * tanh(C_t) \quad (3.6)$$

3.1.3. Optical Flow

Optical flow refers to the movement of individual pixels on the image. Optical flow can be considered as the motion of objects between consecutive frames because of movement the object or camera (Zelkowitz, 2010; Ke, Liu, Bennamoun and et al., 2018).

Many methods are used to estimating the optical flow between two frames, such as differential-based, energy-based, region-based, and phase-based methods. Most of optical flow methods assume that the pixel color and intensity are invariant during displacement from one video frame to another. The most commonly used method among optical flow methods is the differential method that assumes the image brightness stability. Optical flow summarizes the regions of the image during motion and the velocity of motion. Optical flow can be uses in applications, such as tracking vehicles in traffic surveillance system, medical imaging, motion segmentation, fluid velocimetry, obstacle detection, motion recognition, etc (Zelkowitz, 2010; Ke, Liu, Bennamoun and et al., 2018).

Beauchemin et al. (Beauchemin & Barron, 1995) investigated the computation of optical flow. They classified widely known methods for estimating optical flow based on the hypothesis and assumptions they use. As shown in below, optical flow methods are classified in three different groups, namely intensity, frequency, and region-based methods (Anthwal & Ganotra, 2018).

1) **Intensity based methods :** Optical flow is computed by using spatiotemporal derivatives of image intensity. Lucas B.D and Kanade T. method (Lucas & Kanade, 1981) and Horn B.K.P. and Schunck B.G. (Horn & Schunck, 1981) method are in this category.
2) **Frequency based methods :** Optical flow is estimated by using energy output of velocity with Fourier domain. Heeger D.J. method (Heeger, 1988) is in this category.

3) **Region based methods :** Optical flow is obtained by matching features from two frames through maximizing similarity or minimizing distance. Anandan P. method (Anandan, 1989) and Singh A. (Singh, 1990) method are in this category.

Moreover, there are two main variants of the optical flow approach as sparse and dense. The flow vectors of a sparse feature set of pixels for interesting features such as the edges or corners in objects are calculated in sparse optical flow. The Lucas–Kanade method and the Horn–Schunck method are samples for sparse optical flow. The flow vectors of every pixel of each frame are calculated in dense optical flow. Dense optical flow has computationally expensive and calculation for it is slow but it demonstrates higher accuracy. Because of higher accurate results on dense optical flow, it is suitable for applications such as motion recognition and video segmentation. The most popular method for dense optical flow is the Farnebäck method (Lin, 2019; Zelkowitz, 2010).

Farnebäck Method

Farnebäck introduced a novel two-frame motion estimation algorithm. This novel technique calculates each neighborhood of two frames by quadratic polynomials. They presents a novel method to estimate displacement between two frames. Their method is based on polynomial expansion in an image. (Farnebäck, 2003; Anthwal & Ganotra, 2018).

Basic equations for Farnebäck method are given below :

Exact quadratic polynomial can be used to calculate neighborhood of each pixel :

$$f_1(x) = x^T A_1 x + b_1^T x + c_1 \text{ (Farnebäck, 2003)} \quad \textbf{(3.7)}$$

A new signal f_2 can be constructed by a global displacement by d :

$$f_2(x) = x^T A_1 x + (b_1 - 2A_1 d)^T d^T A_1 d - b_1^T d + c_1 \text{ (Farnebäck, 2003)} \quad \textbf{(3.8)}$$

$f_2(x)$ can be written a compatible form with first equation :

$$f_2(x) = x^T A_2 x + b_2^T x + c_2 \text{ (Farnebäck, 2003)} \quad \textbf{(3.9)}$$

$$b_2 = b_1 - 2 A_1 d \text{ (Farnebäck, 2003)} \quad \textbf{(3.10)}$$

$$d = -\frac{1}{2} A_1^{-1}(b_2 - b_1) \text{ (Farnebäck, 2003)} \quad (3.11)$$

3.2. Classification

3.2.1. Support Vector Machines

Support Vector Machine, abbreviated as SVM, is a supervised machine learning algorithm that can be mainly used for classification, regression and outlier detection tasks. It is widely used in classification problems (Wlodarczak, 2020; Ikram & Cherukuri, 2016).

The main goal of SVM is to find a hyperplane in an n-dimensional space that can classify the data points into two classes. n means the number of features (Wlodarczak, 2020).

SVM uses a maximized margin criterion and actually seperates the binary classes(k=2). However, many real world applications such as optical character recognition and speech recognition often require more than two categories of discrimination. SVM can perform multiclass classifications in several ways such as an ensemble of binary classifiers or by extending margin concepts. For example, the multi-class classification problems (k>2) are decomposed into a series of binary problems so that standard SVM can be applied directly (Ma & Guo, 2014; Aggarwal, 2014).

SVM is highly preferred over other machine learning algorithms because less computation is required and significant accuracy can be achieved (Wlodarczak, 2020).

The major advantages of SVM are:

- It gets efficient results in high dimensional spaces
- It works well on small datasets
- It is helpful in cases where the quantity of dimensions is greater than the quantity of samples.
- It uses memory efficiently. It uses only a subset of training points when running the decision making function (called support vectors)
- Many kernel functions available for the decision function. Common kernels can be used or custom kernels can be developed.

The disadvantages of SVM are: (Ray, 2017)

- Choosing the right kernel and parameters can cause computational intensity.
- It doesn't work well in large datasets because of higher training time
- It also doesn't work very well, if the dataset has more noise
- SVM doesn't directly provide probability estimates, they are calculated with an expensive five-fold cross-validation.

SVM has C parameter as the penalty parameter. The C parameter is a punishment factor to be used to decide less or high penalize misclassified data points.

SVM performs to classify the data linearly, but in some cases this is not possible. In order to handle this situation, the kernel trick is applied. If a new dimension is created, then classification can be done linearly (Salcedo, 2019).

Types of Kernel Functions

There are four types of kernel functions : (Salcedo, 2019)

- **Linear :** It is a simple function. It can work well with few nonlinear relationships.
- **Radial Basis Function (RBF) :** It is similar to an RNN. It works well with nonlinear relationships.
- **Polynomial :** It is a more complext function. It can work well with some nonlinear relationships.
- **Sigmoid :** It is similar to a two-layer neural network. It works well with nonlinear relationships.

Linear Support Vector Machine

In Figure 3.13, there is a hyperplane that can be separated two classes of data points as the linear classifier. The classifier has the set of pairs (w, b), where w is a weight vector, b is the bias, x_i is sample in the training set, y_i represents the class label: (Latah, 2017; Ikram & Cherukuri, 2016).

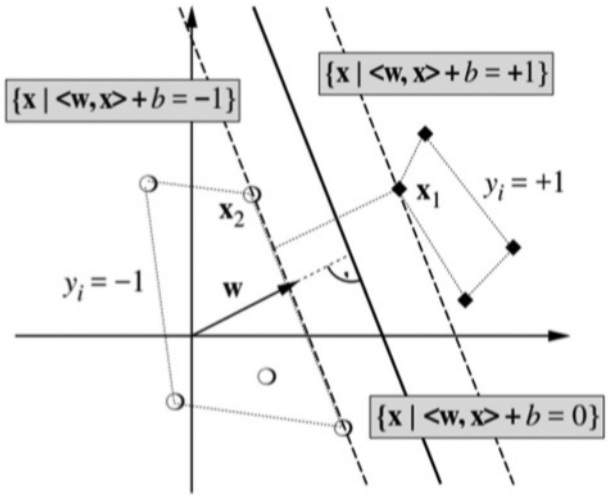


Figure 3.13 Linear support vector machine (Ikram & Cherukuri, 2016).

$\boldsymbol{A}\{(\boldsymbol{x_1}, \boldsymbol{y_1}), (\boldsymbol{x_2}, \boldsymbol{y_2}), \ldots (\boldsymbol{x_n}, \boldsymbol{y_n})\}$ is a given dataset, where $\boldsymbol{x_i} \boldsymbol{\epsilon} \boldsymbol{R^n}$; $\boldsymbol{y_i} \boldsymbol{\epsilon} \pm \mathbf{1}$, and i represents a label associated with each activity in the dataset, the set of all hyperplanes is described in Equation (3.12) and Equation (3.13).

$$w.x_i + b \geq +1;\ y_i = +1 \quad \textbf{(3.12)}$$

$$w.x_i + b \leq -1;\ y_i = -1 \quad \textbf{(3.13)}$$

$$\text{Minimize} : \| w \| \text{ subject to} : \quad y_i\ (w.x + b) \geq 1 \quad \textbf{(3.15)}$$

w is normalized with respect to a set of points x. So $min_i \mid \text{w}.x_i \mid = 1$. To maximize the distance between the hyperplanes, it is necessary to minimize ‖w‖. So, this is an optimization problem. Equation (3.12,3.13) can be converted to form in (3.15). Every hyperplane (w, b) in Figure 3.13 is a classifier which separates all patterns in the training set.

Radial Basis Function (RBF)

The RBF kernel is one of the most commonly used kernel functions. (Ikram & Cherukuri, 2016)

$$K(x, x_i) \geq \exp\left(-\gamma \|x - x_i\|^2\right) \quad \textbf{(3.16)}$$

$$f(x) = \sum_{i=1}^{m} \alpha_i \exp\left(-\gamma \|x_i - x\|^2\right) + \text{b} \quad \textbf{(3.17)}$$

$|| xi - x ||^2$ is the squared Euclidean distance between two vectors. σ is an unrestricted parameter. It is defined as :

$$\gamma = \frac{1}{2\sigma^2} \quad \textbf{(3.18)}$$

3.2.2. Video Classification

The video classification problem is not much different from the image classification problem. In the image classification process, the images are taken and the features are extracted, neural networks such as CNNs are used for feature extraction, and then the classification process is performed. There is one more step in the video classification process: Video frames are extracted from the videos (time base information) first. The steps in the image classification process can then be implemented.

In this section, digital video representation and microvideo definition are given.

Digital Video Representation

Digital video is a representation of a series of digital images displayed in rapid sequences as seen in Figure 3.14. While processing video, video frames are usually represented in a matrix format.

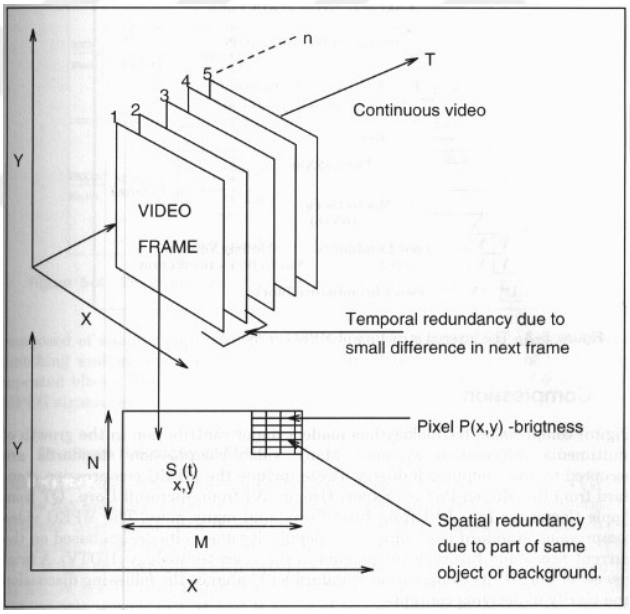


Figure 3.14 Representation of digital video
(http://people.cs.pitt.edu/~chang/265/C5/v2.htm. Access date: 01.03.2020).

640 x 480 pixels = 307,200 Pixels per frame : It describes video quality.

307,200 x 24 bits x byte/8 bits = 921,600 : Bytes per frame.

921,600 x 30 frames per second = 27.648 MB : 30 fps is desirable.

FPS (Frame per Second) : The number of frames that exist in the image in 1 sec.

The base fps value is generally accepted as 24 for a fluent image perception. For example; a movie at 30 fps means it will show 30 frames in 1 second. If a computer game is 120 fps, it means that 120 frames are given in 1 second.

Each pixel is identified by a certain number of bit combinations. For example, it may be possible to show 256 different colors, which are the 8th power of 2, with the combination of 8-bit.

Each pixel also consists of 3 points - red, blue and green (RGB). 3 pieces of 8-bit information can show 24-bit colors. Each 8-bit RGB element can take 256 different values, ranging from 0 to 255. For example, there may be three values (250,165.0). This combination creates (Red = 250, Green = 165, blue = 0) orange color.

Microvideo

A microvideo is a short form of video. A microvideo with optimal length meets desired outcomes such for knowledge transfer or some logical actions. It is generally 6-15 seconds long.

4. RESEARCH METHODOLOGY : HUMAN MOTION IMITATION

4.1 Introduction

Robotic applications try to find a map between world state and actions, namely policy. It enables a robot to choose an action in current world state. Policy development is subject to ML/DL also. Learning from Demonstration(LfD) is an approach to policy learning (Argall, Chernova, Veloso and et al., 2009).

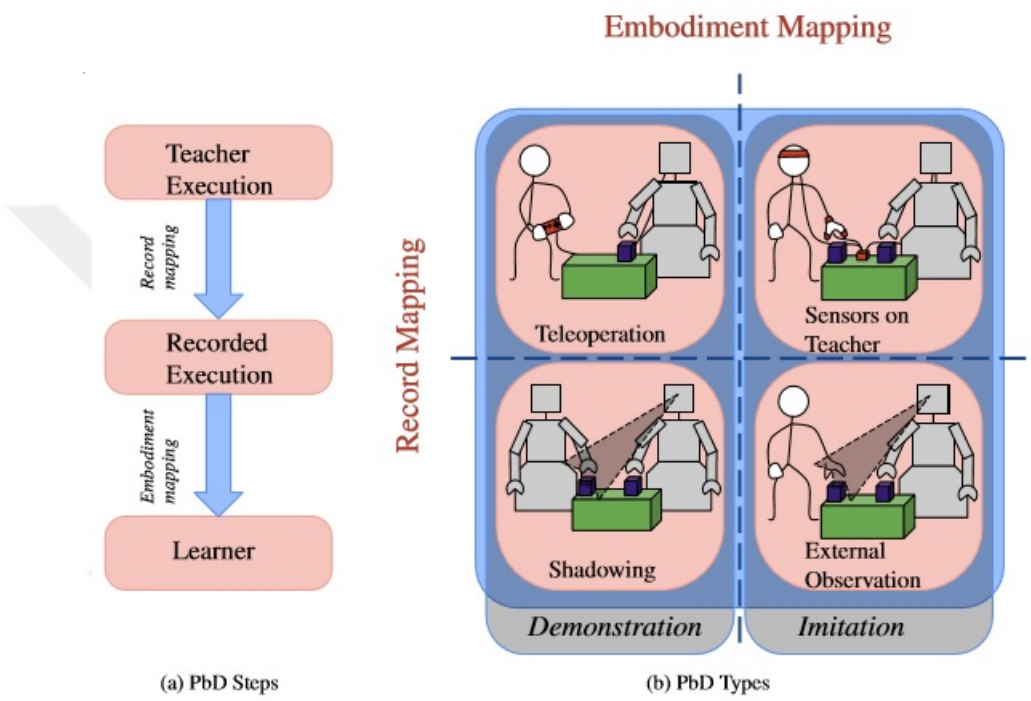


Figure 4.1 Programming-by-Demonstration classification according to (Argall, Chernova, Veloso and et al., 2009; Romeo, 2011).

There is a correspondence issue to map the teacher and the learner to transfer information from one to the other illustrated in Figure 4.1. There are four type of action mapping to develop policies for robot Learning from Demonstration (LfD) (Argall, Chernova, Veloso and et al., 2009).

- **Demonstration :** When teacher executions are demonstrated, there is no issue for embodiment mapping between the teacher and learner (Argall, Chernova, Veloso and et al., 2009).
    - **Teleoperation :** The robot learner is operated by the teacher with its own sensors. The record mapping is direct. The advantage of teleoperation is that it is the most direct method for LfD. The disadvantage of teleoperation

is that it requires the robot be manageable, so it is not suitable for all system (Argall, Chernova, Veloso and et al., 2009).

- **Shadowing :** The robot mimics the teachers motions during shadowing. The record mapping is not direct. Shadowing requires an algorithm to make the robot to track motion (Argall, Chernova, Veloso and et al., 2009).

- **Imitation :** When teacher executions are imitated, there is an issue for embodiment mapping between the teacher and learner (Argall, Chernova, Veloso and et al., 2009).
  - **Sensor on teacher :** This approach utilizes recording sensors such as human-wearable sensors directly located on the platform. There is no record mapping. The teacher gives sensitive measurements of the execution (Argall, Chernova, Veloso and et al., 2009).
  - **External observation :** Imitation is performed with external observation by external sensors. There is a record mapping. Robots don't record the teacher actions directly. Generally, first the teacher actions are extracted from recorded data, and then mapped the extracted actions to the learner. This method is more general than sensors-on teacher aproach (Argall, Chernova, Veloso and et al., 2009).

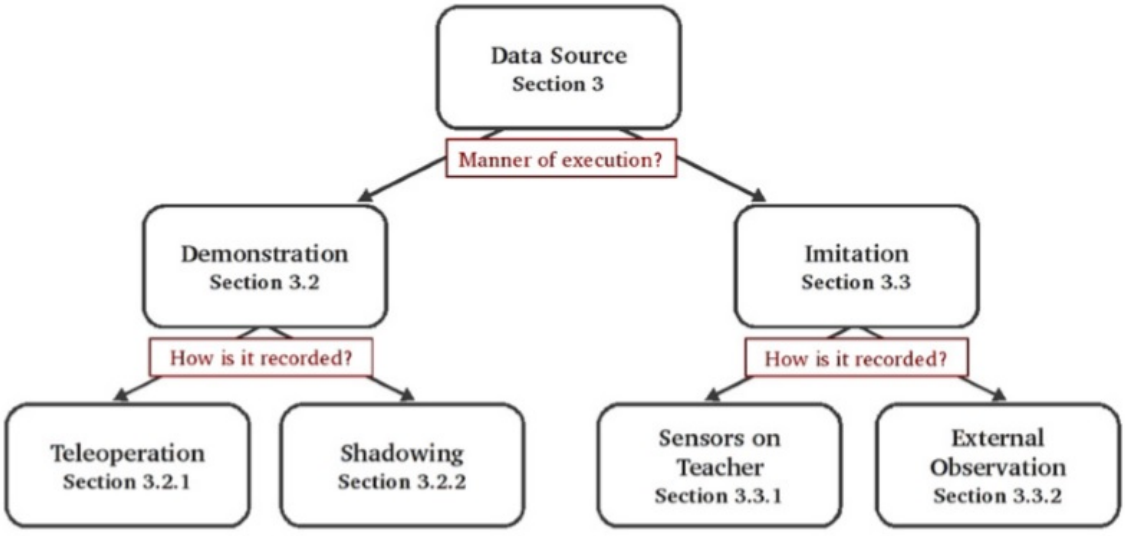


Figure 4.2 LfD aproach categorization (Argall, Chernova, Veloso and et al., 2009)

The following human movement imitation can be considered a LfD type of imitation by external observation.

4.2 Approaches to Human Motion Imitation

Reproducing human motion is the inspiration in the context of robotics research. One of the most important goals of robotic researchers is to reproduce some subsets of human motion capabilities. This chapter introduces human motion imitation (HMI), and the key approaches for modeling human motion and reproducing it by robots. The main challenge for reproducing human motion is to map kinematic and dynamic attributes between the human and the robot and to keep robot postural stability. This section introduces the general data processing pipeline in the process starting from capturing human motion and processing motion data to reproducing new motion data. Next, various human motion imitation approaches are introduced (Kulić, 2019).

HMI processing pipeline contains four stages : First step is human motion capturing. Second, motion data is segmented with preprocessing. Then, extracted data is converted to meaningful format for imitation. Next, motion model is created. Finally, the motion is reproduced by robots (Kulić, 2019).

Three main approaches are overviewed below (Kulić, 2019):

1) Direct human trajectory reproduction
2) Motion primitive based approaches
3) Control policy approaches.

4.2.1. Direct Reproduction with Separate Postural Control

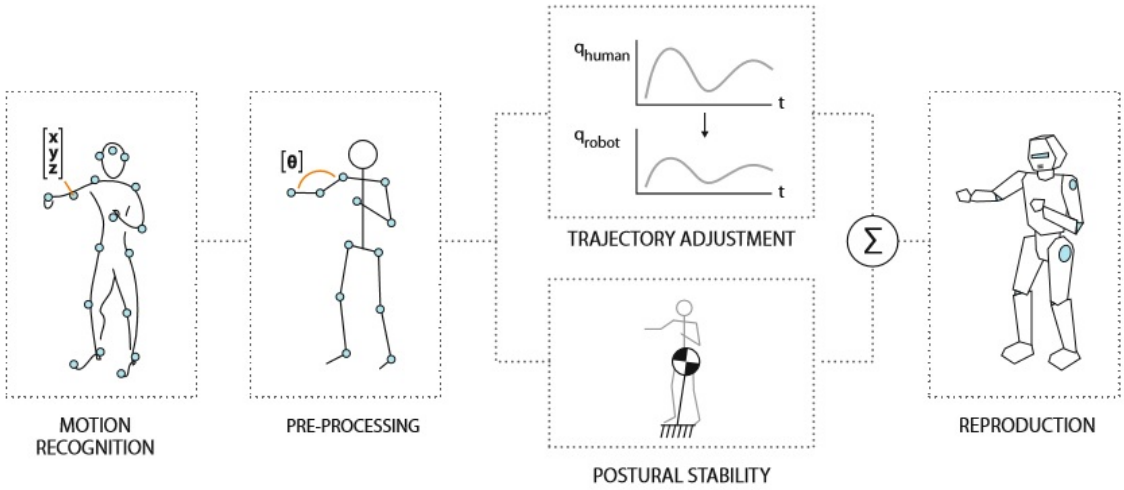


Figure 4.3 Overview of the direct reproduction approaches. First human motion data is captured and preprocessed. Next, joint angle data is mapped to robotic joint angle trajectories, while other method ensuring postural stability (Kulić, 2019).

In this approach, a robotic control method is formulated by following human joint angle trajectory while ensuring posture balance stabilization. This approach is illustrated in Figure 4.3.

With this approach, some researchers focus to reproduce upper-body motion, they allow lower body to ensure postural balance (Dariush, Gienger, Jian et al., 2008; Dariush, Gienger, Arumbakkam et al., 2009; Kim & Kim, 2013; Ott, Lee & Nakamura, 2008). Some researchers describe human motion generation to reproduce lower-body motion (Harada, Miura, Morisawa et al., 2009). A third class of researchers target both upper and lower body motion with formulation a motion controller (Yamane & Nakamura, 2003).

4.2.2. Motion Primitive-Based Approaches

Human motion is decomposed into a set of primitive motions, namely motion primitives. Then, motion primitives and their sequences are modeled, so motion can be imitated. In Figure 4.4, the motion primitive-based approach is shown (Kulić, 2019).

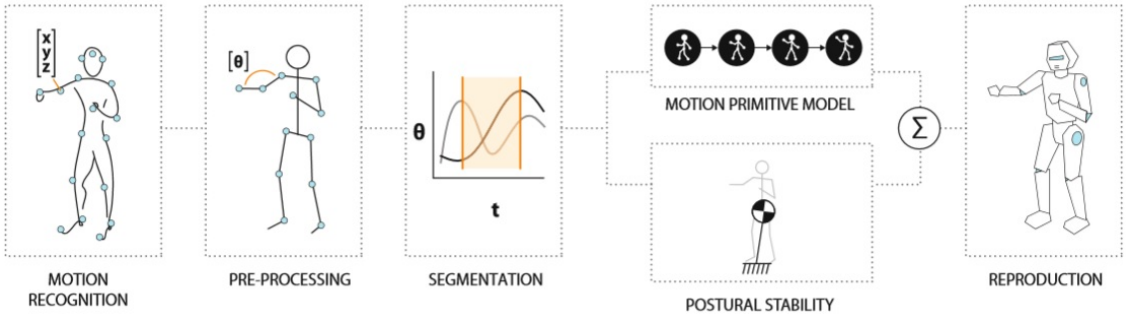


Figure 4.4 Overview of the motion primitive approaches. First human motion data is captured and preprocessed. Next, movement data is segmented into motion primitives. Subsequently, for each primitive, a dynamic model is learned while other method ensuring postural stability. The trajectory of the robot is generated by the motion primitive, and then updated by the postural stability component (Kulić, 2019).

We can divide this approach into two parts : Off-line learning of motion primitives and online learning of them. During off-line learning, the training data including motion primitives is segmented and labeled manually (Kulić, 2019).

Off-line learning of motion primitives :

Ikeuchi and colleagues have developed a pioneer approach to imitate dance movements by humanoid (Nakaoka, Nakazawa, Yokoi et al., 2004; Nakoka, Nakazawa, Yokoi et al., 2003; Okamoto, Shiratori, Kudoh et al., 2014). They separated the movement into upper- and lower-body movement. For the upper body, they used observed joint trajectories (Kulić, 2019).

For the lower body, they defined three motion primitives namely stand, step, and squat. They reproduced lower body motion as a sequence of these primitives. They modified zero moment point (ZMP) trajectory to ensure posture stabilization. They tested their methods both in simulation environment and the HRP-1S robot. Subsequently, they extended the motion primitive approach to upper-body motion. They defined key poses which includes starting posture segment and ending postures segment. In (Okamoto, Shiratori, Kudoh et al., 2014), they seperated body motions into upper, middle and lower body motion. They decomposed the dance motions to motion primitives. They applied their strategies to demonstrate traditional Japanese dance on the HRP2 humanoid robot (Kulić, 2019).

Takano et al. (Takano, Yamane, Sugihara et al., 2006) and (Takano, Yamane & Nakamura, 2005) proposed a hierarchic hidden Markov models (HMMs) to learn human

motion patterns and interactions between human during combat. The primitive motion patterns, such as kick, punch, etc. have been abstracted by the lower layer of HMMs. First, primitive motion patterns are learned, then motion is generated using similarity of HMMs(Kulić, 2019).

Online learning of motion primitives :

Kuli´c et al. (Kuli´c, Takano & Nakamura, 2008; Kuli´c, Ott, Lee et al., 2012) introduced a method for online learning of motion primitives and their sequences by observing human motion. They performed unsupervised segmentation with online observation (Kuli´c, Takano & Nakamura, 2009). They modeled each segment as an HMM and performed hierarchical clustering, and then motion primitives are learned incrementally (Kuli´c, Takano & Nakamura, 2008). Finally, they present a model for sequencing the primitives by learning observed transitions between the learned primitives (Kuli´c, Ott, Lee et al., 2012). They used proposed system to reproduce upper-body human motion by a humanoid robot (Kulić, 2019).

Extensions to the motion primitive framework enables the robot learn complex tasks including interaction with objects or the environment (Gams, Nemec, Ijspeert et al., 2014; Rozo, Calinon, Caldwell et al., 2016). The motion primitive approach enables to convert continuous motion data into discrete action units or symbols. These motion primitives can be formulated and sequenced to achieve the goal including complex actions (Kulić, 2019).

4.2.3. Control policy approaches

In this approach, it is assumed that human has a controller type to generate motion. The imitation learning system targets to learn human control policy with some formulation and then reproduce motion by robot. The control policy approach is illustrated in Figure 4.5 (Kulić, 2019).

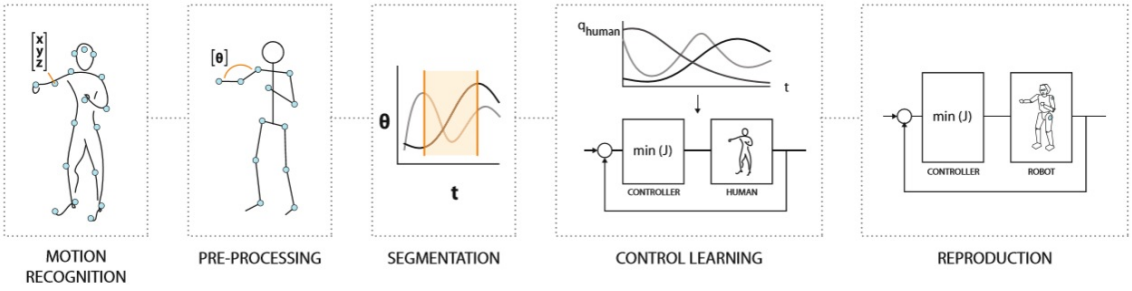


Figure 4.5 Overview of the control policy approaches. First human motion data is captured and preprocessed. Next, movement data is segmented into motion primitives. Subsequently, the controller is estimated with motion primitive data and human by observing trajectories. After the controller function is determined, it can be used for motion reproduction by robots (Kulić, 2019).

4.2.3.1. Model Learning

This approache is based to learn a dynamic mapping between trajectory of target and the action space (Cole, Grimes & Rao, 2007; Grimes, Chalodhorn & Rao, 2006; Chalodhorn, Grimes, Grochow et al., 2007; Kulić, 2019).

4.2.3.2. Reinforcement Learning

Imitation system learns the control policy iteratively by repeating to execute task, when the attempt is successful, then it is rewarded (Kober, Bagnell & Peters, 2013). In traditional RL, robots learn control policy by trial and error on their own, the motion results may not be like human motion (Kulić, 2019).

4.2.3.3. Direct Reproduction with Separate Postural Control

This approach assumes that an optimal controller using an unknown optimization function controls human motion. Discovering this optimization function with an observed trajectory is the main goal of optimal control (Kulić, 2019).

This function consists of a weighted sum of base functions. Finding the weights of the objective function is the main goal. The optimization problem is then formulated by minimizating the difference between the observed motion and generated motion of objective function results. Objective function is used to solve optimal control problem by iteration (Mombaur, Truong & Laumond, 2010; Kulić, 2019).

5. A HYBRID DEEP MODEL FOR HAR

5.1 Architecture of the Proposed Deep Model

In this work, a hybrid deep model for human activity recognition in videos is introduced as seen in Figure 5.1. There is a need to idendify motion vectors in the t and t-1 frames for this purpose. We combined three streams deep networks to contruct the feature vectors : 3D-CNN with multiple frames, 3D-CNN with dense optical flow, and LSTM with auxiliary motion information. To the best of our knowledge, this is the first attempt to recognize human activity using the combination of multiple frames, optical flow, and LSTM. And then, after feature extraction step, SVM is used to classify activities with 4 output classes.

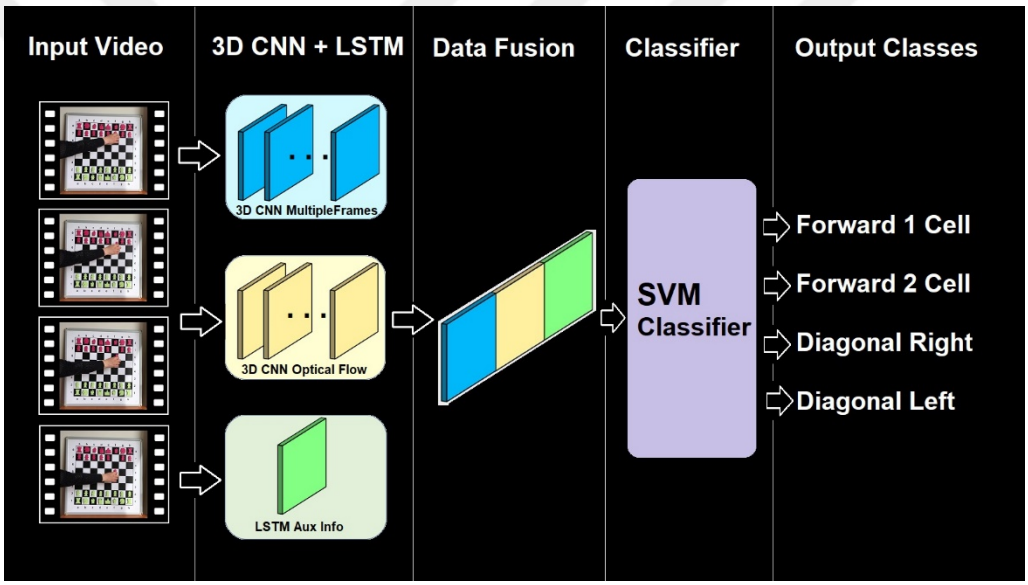


Figure 5.1 Block diagram of the proposed architecture for hybrid deep model.

Extracted features with 3D-CNN with multiple frames, 3D-CNN with dense optical flow, and LSTM from video are located in arrays and then files. Due to the size limitation, resolution of frames to be an input for 3D-CNNs in videos are decreased. And then multiple frames in frame t are evaluated as an input for 3D-CNN, which is first stream of hybrid deep model. Dense optical flow in frame t-1 are calculated by using the Farnebäck algorithm as an input for 3D-CNN, which is second stream of hybrid deep model. Hand-tracking and objects are gathered to provide auxiliary information as an input for LSTM, which is third stream of hybrid deep model. The LSTM is fed a model with a single shot detector (SSD) network architecture for hand tracking purposes and another neural

network embedded in TensorFlow Object Detection API (Francis, 2017) for object recognition. After obtaining features from each separate model, the features collected from 3D-CNN model with multiple frames and the features gathered from the 3D-CNN with the dense optical flow model are concatenated. The new feature vector is then enhanced with the new features acquired from the LSTM model. In summary, a novel feature vector is constructed using 3D-CNN with multiple frames, 3D-CNN with dense optical flow, and LSTM as shown in Figure 5.2.

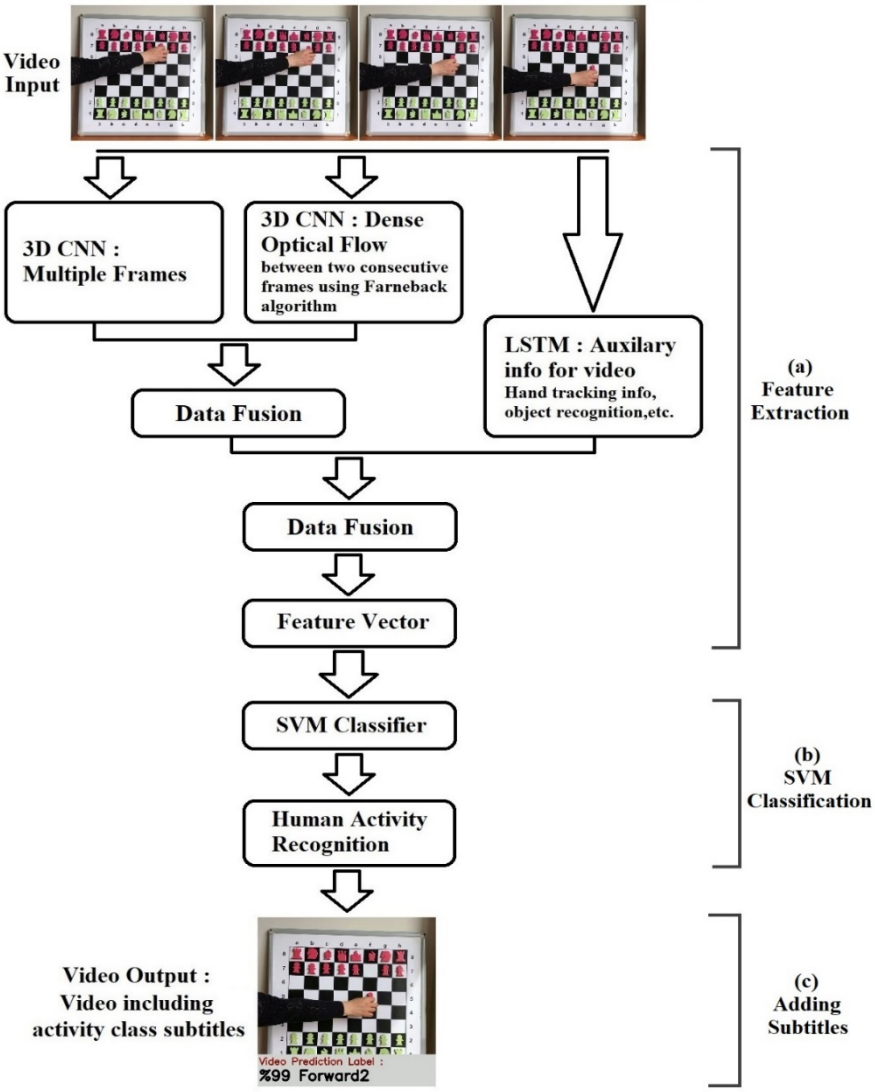


Figure 5.2 Flowchart of the proposed system.

In the proposed hybrid deep model, we followed this structure respectively : Convolution3D, SpatialDropout3D, MaxPooling3D, Convolution3D, MaxPooling3D, Dropout, Dense, Dropout for 2-stream 3D CNNs, and then third stream is LSTM via data fusion with concatenate as seen Figure 5.3. We used images which are resized to 50×50.

We used first 100 frames for each video sample. We used SpatialDropout3D dropout layer with a dropout rate of 40%, and then Dropout with a dropout rate of 50% to avoid overfitting. Convolution3D and Dense layers also use ReLU as activation function. Convolution3D layer with a kernel size 3×3×3, MaxPooling3D layer with a kernel size 2x2x2 and strides=(2, 2, 1) are used also. Then, one fully connected (FC) layer with 166 units and a classification layer with 4 output class is used.

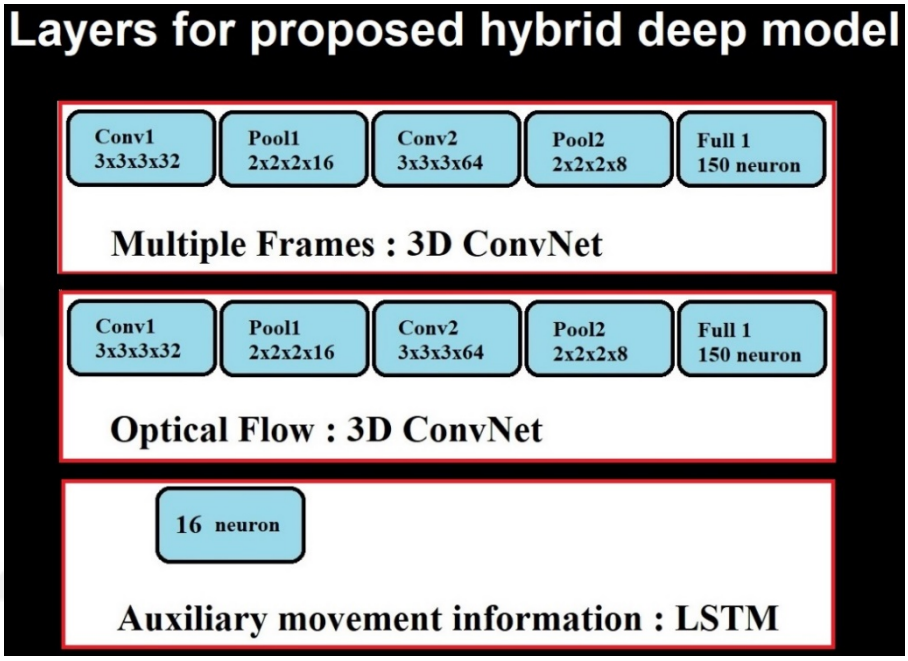


Figure 5.3 Proposed hybrid deep model architecture.

In this study, we offered a pipeline illustrated end to end model in Figure 5.4 to implement HAR for the particular problem definition of chess moves. We used our proposed deep model to recognize activities by chess player.

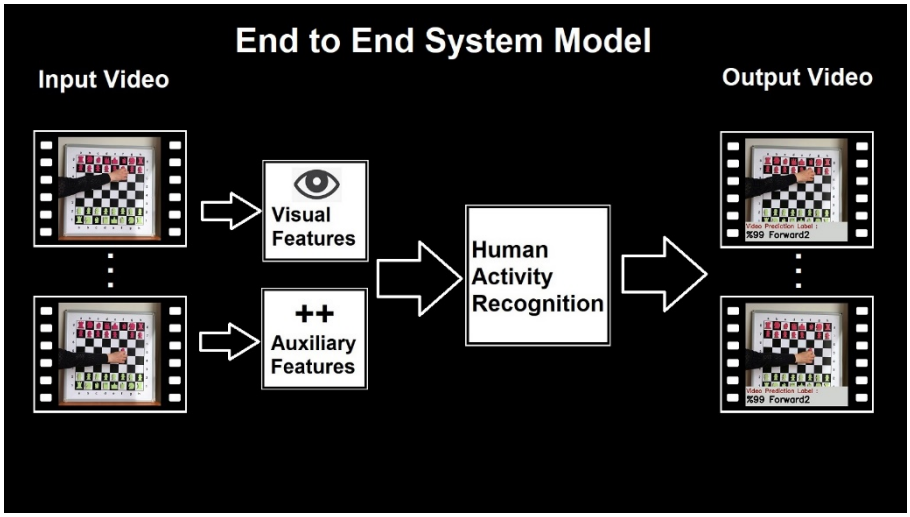
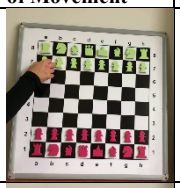
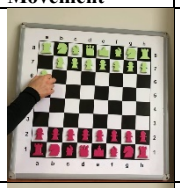
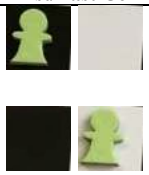
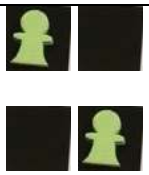
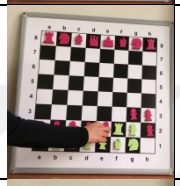
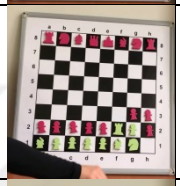
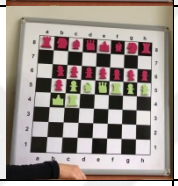
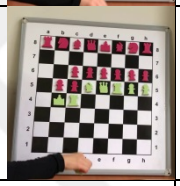
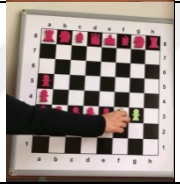
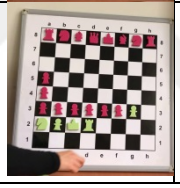


Figure 5.4 End to End system model for HAR implementation via deep learning approaches.

The problem of recognition of human activities is addressed as classification task performed by support vector machine (SVM) algorithm. At first, activity classes are tagged for the magnetic wall chess board dataset (MCDS) and the standard chess board dataset (CDS). The activity classes of the proposed system are Forward1, Forward2, DiagonalLeft, DiagonalRight, and Trash as observed in Table 5.1. We then focus on the classification task to determine the class of motion or activity when the motion whose previous class label is actually known is encountered in the video. We used precision, accuracy, recall, and F-measure as the evaluation metrics of the classification performance of the proposed system to demonstrate the contribution of our work.

Table 5.1 Examples of activity classes for magnetic wall chess board dataset (MCDS) with each corresponding video frame.

Activity Class	Frame Before Motion	Starting Frame of Movement	End Frame of Movement	Frame After Motion	Aux : Crop First/Last Cell
Forward1					
Forward2					
DiagonalLeft					
DiagonalRight					
Trash					

5.2 Dataset Acquisition and Preparation

We built three new datasets acquired by mobile phones to be used in the human activity recognition. First of them is a pawn dataset of a magnetic wall chess board to be used in object recognition. Second dataset is a video dataset for a magnetic wall chess board. Third dataset is a video dataset for a standard chess board. After data collection is completed, data preprocessing phase is started. In this phase, VivaVideo is used for formatting, ant then Bandicut is used to split videos into meaningful parts. Data preparation is conducted as described in detail below :

Data Preparation for Magnetic Wall Chess Board Video Dataset (MCDS): The dataset is preprocessed before the feature extraction step by applying the following phases:

1. Split all the videos into basic activity folders as DiagonalLeft, DiagonalRight, Forward1, and Forward2. These activity folders are class labels at the same time.
2. Extract as .jpg format of each frame for each video into a tmp_xxx folder.
3. Summarize the videos as their classes, frame numbers for some statuses, etc. in a CSV file.
4. Based on frame numbers in the CSV file:
    a. Split all the videos into all activity folders: DiagonalLeft, DiagonalRight, Forward1, Forward2, Trash.
    b. Extract as .jpg format for the empty chessboard for each video.
    c. Extract as .jpg format for the chessboard image before movement starts for each video.
    d. Extract as .jpg format for chessboard image after movement ends for each video.
    e. Crop as .jpg for the chessboard's first processed cell and second processed cell before movement starts for each video.
    f. Crop as .jpg for the chessboard's first processed cell and second processed cell after movement ends for each video.
5. The feature extraction step is run.

Data Preparation for Standard Chess Board Video Dataset (CDS): The dataset is processed before applying feature extraction as follows:

1. Split all the videos into activity folders as DiagonalLeft, DiagonalRight, Forward1, Forward2. These activity folters are class labels at the same time.
2. Extract as .jpg format of each frame for each video into a tmp_xxx folder.
3. Summarize the videos as their classes, frame numbers for some statuses, etc. in a CSV file.
4. Extract as .jpg format for the empty chessboard for each video based on frame numbers in the CSV file.
5. The feature extraction step is run.

5.3. Experimental Results

We used Ubuntu 16.04 system with an Intel Core i7 machine and NVIDIA GTX1080 Ti GPU for wide comparative experiments. We developed needed libraries

with Python. We used magnetic wall chess board dataset (MCDS) and standard chess board dataset (CDS) in the experiments. We randomly divide the datasets into two parts whereby 70% of the data are used for training and 30% for testing. First, we collected videos in 1080x1080 resolution format, and then resized each frame to 50x50 resolution for size limitations. In the feature extraction step, we used OpenCV library to read and write videos. We extracted total 100 frames for each video instance. We extracted features using TensorFlow Object Detection API (Francis, 2017) for custom objects including pawn recognition, patterns in chess boards (https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html#re-projection-error access date: 25.05.2019), and real time hand detection with single shot detector (SSD) network architecture in TensorFlow (Dibia, 2017; Ren, He, Girshick et al., 2017). Furthermore, in the feature extraction step, we used Keras functional api to construct the proposed multi-stream hybrid deep model. We used both SpatialDropout3D and Dropout techniques in order to handle overfitting.

The classification performance of the proposed system is evaluated with precision, accuracy, recall, and F-measure as evaluation metrics using the confusion matrix seen in Table 5.2 to demonstrate the contribution of our work.

Table 5.2 Confusion matrix.

	Class 1 Predicted	**Class 2 Predicted**
Class 1 Actual	TP (true positive)	FN (false negative)
Class 2 Actual	FP (false positive)	TN (true negative)

Precision, Recall, Classification Rate/Accuracy, and F-measure are calculated as follows:

$$Precision = \frac{TP}{TP + FP} \qquad Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall = \frac{TP}{TP + FN} \qquad F\text{-}measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{5.1}$$

In all tables, abbreviations are employed for deep models as follows: HDM: Flexible multi-stream hybrid deep model which is the proposed model in this work, CNN-OF: Convolutional neural network model combined with dense optical flow which is a sub-model of HDM, HDM – MCDS: The classification performance of HDM on the MCDS dataset, HDM – CDS: The classification performance of HDM on the CDS dataset, CNN-OF – MCDS: The classification performance of CNN-OF on the MCDS

dataset, CNN-OF – CDS: The classification performance of CNN-OF on the CDS dataset. The best classification results acquired are indicated in bold in the tables. First, we analyze the classification performance of the proposed system in terms of precision, recall, and F-measure results for different human activities on magnetic wall chess board (MCDS) and standard chess board (CDS) datasets using HDM and CNN-OF.

Table 5.3 Precision, recall, and F-measure results of the proposed HDM model and CNN-OF for different activities on the MCDS and CDS datasets.

		MCDS			**CDS**		
Model	**Activity Class**	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
HDM	DiagonalLeft	0.88	0.96	0.918	1.00	1.00	1.00
	DiagonalRight	0.94	0.94	0.94	1.00	1.00	1.00
	Forward1	1.00	0.92	0.958	0.91	0.84	0.874
	Forward2	1.00	1.00	1.00	0.73	0.84	0.781
	Trash	0.92	0.92	0.92	NA	NA	NA
CNN-OF	DiagonalLeft	0.79	0.92	0.85	1.00	1.00	1.00
	DiagonalRight	0.88	0.88	0.88	1.00	1.00	1.00
	Forward1	0.96	0.96	0.96	0.94	0.78	0.853
	Forward2	1.00	0.86	0.925	0.68	0.89	0.771
	Trash	0.92	0.92	0.92	NA	NA	NA

Table 5.3 demonstrates the classification results of different activities in terms of precision, recall and F-measure for the proposed models on the MCDS and CDS datasets. Forward2 movement as a human activity is recognized with 100% classification success with the proposed HDM model on the MCDS dataset when considering the F-measure results. It is followed by Forward1 with 95.8%, DiagonalRight with 94.0%, Trash with 92.0%, and DiagonalLeft with 91.8% for the MCDS dataset. With the CNN-OF model, the best classification success is achieved by Forward1 movement with 96.0% performance on the MCDS dataset. The classification success of human activities for the CNN-OF model on the MCDS dataset can be summarized as follows: Forward1 > Forward2 > Trash > DiagonalRight > DiagonalLeft. As a result, usage of the HDM model

can provide 100% classification success on the MCDS dataset. On the other hand, DiagonalLeft and DiagonalRight movements are recognized with 100% classification performance with the HDM method on the CDS dataset. The classification performances for the CDS dataset are ordered as follows: DiagonalLeft = DiagonalRight > Forward1 > Forward2 > Trash. In the CNN-OF model, the best classification success is shared between DiagonalLeft and DiagonalRight with 100% classification performance as observed before in the HDM model. However, with the CNN-OF model, the Forward1 and Forward2 movements exhibit 2% and 1% lower classification success, respectively when compared to the HDM model. As seen in Table 5.3, the proposed HDM model generally exhibits remarkable classification success compared to the CNN-OF method in terms of F-measure results.

Table 5.4 demonstrates confusion matrix results for both MCDS and CDS datasets by comparing the HDM and CNN-OF models in terms of recall results. In this table, the x-axis denotes the predicted labels while the y-axis represents the actual labels.

- HDM – MCDS: Maximum confusion is observed between Forward1 and DiagonalLeft movements whereas the best results are achieved by the Forward2 class.
- HDM – CDS: Maximum confusion is seen between Forward1 and Forward2 activities while DiagonalLeft and DiagonalRight movements yield the best classification results.
- CNN-OF – MCDS: Forward2 and DiagonalLeft represent the maximum confusion while the Forward1 and Trash activity classes yield the best results.
- CNN-OF – CDS : Between Forward1 and Forward2 movements, maximum confusion is clearly observed while DiagonalLeft and DiagonalRight activities exhibit the best classification performances.

Some classes with similar activities are more readily confused with each other. A possible reason for them interfering with each other in this way is the similarity of features and representations among the activities.

Table 5.4 Confusion matrix results in term of recall metric.

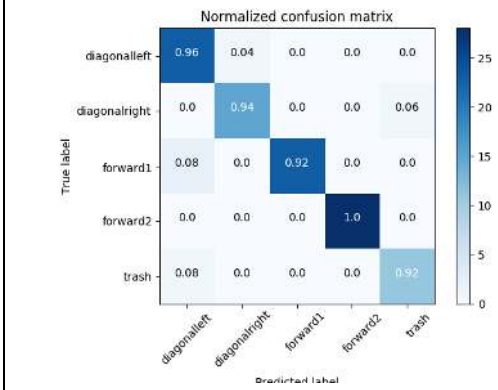
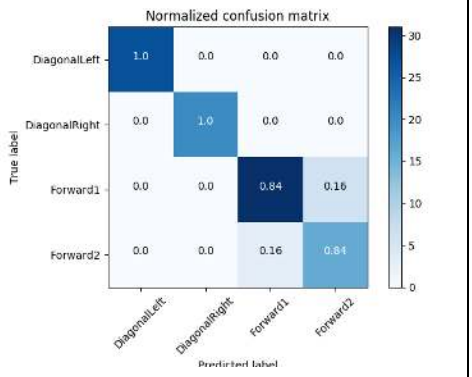
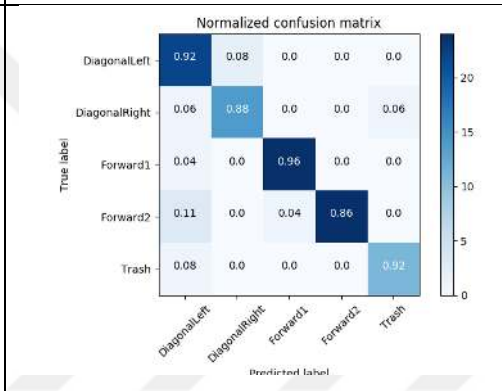
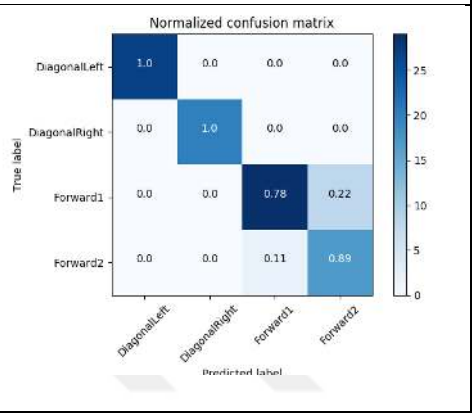
Model	MCDS	CDS
HDM		
CNN-OF		

Table 5.5 displays the overall accuracy results of the HDM and CNN-OF models on the MCDS and CDS datasets. In order to compare our proposed model with the CNN-OF model trained with our new datasets, the overall accuracy of the proposed system is calculated using Equation (5.1). The proposed method achieves 95.2% accuracy on the MCDS and 91.3% accuracy on the CDS dataset while the CNN-OF model exhibits 90.5% accuracy on MCDS and 90.3% accuracy on CDS. This means that the newly proposed hybrid deep model outperforms CNN-OF with nearly 5% enhancement on the MCDS dataset and with 1% improvement on the CDS dataset. The performance of the proposed HDM model on the MCDS dataset is marked by almost 4% more improvement compared to the CDS dataset. The main reason for this result is that there is no auxiliary information contained in the CNN-OF model, and the auxiliary information of the MCDS dataset used in the HDM model is more than that of the CDS dataset. As seen in Table 5.5, the usage of the HDM model improves the classification performance of the system thanks to the auxiliary information.

Table 5.5 Overall accuracy results of HDM and CNN-OF models on MCDS and CDS datasets.

Model	MCDS	CDS
HDM	**0.952**	**0.913**
CNN-OF	0.905	0.903

Table 5.6 provides the accuracy results of the proposed HDM model and CNN-OF for different activities on the MCDS and CDS datasets with accuracy results for each class.

Table 5.6 Accuracy results of the proposed HDM model and CNN-OF for different activities on MCDS and CDS datasets.

		MCDS	CDS
Model	**Activity Class**	**Accuracy**	**Accuracy**
HDM	DiagonalLeft	0.962	1.000
	DiagonalRight	0.980	1.000
	Forward1	0.980	0.913
	Forward2	1.000	0.913
	Trash	0.980	NA
	OverAll Accuracy	**0.952**	**0.913**
CNN-OF	DiagonalLeft	0.922	1.000
	DiagonalRight	0.960	1.000
	Forward1	0.979	0.903
	Forward2	0.960	0.903
	Trash	0.979	NA
	OverAll Accuracy	**0.905**	**0.903**

- DiagonalLeft: HDM outperforms CNN-OF with approximately 4% enhancement on the MCDS dataset for DiagonalLeft class. Both models show the same classification success on the CDS dataset for DiagonalLeft class.

- DiagonalRight: HDM represents almost 2% improvement compared to CNN-OF on the MCDS dataset for the DiagonalRight class. Both models exhibit the same classification performance on the CDS dataset for DiagonalRight class.
- Forward1: The classification performances of the models are almost the same for the MCDS dataset for the Forward1 class. For the CDS dataset, the performance difference between the two models is nearly 1% for the Forward1 class.
- Forward2: HDM surpasses from CNN-OF by 4% enrichment in classification accuracy on the MCDS dataset for the Forward2 class. On CDS dataset, the classification improvement of the HDM model reaches 1% compared to CNN-OF for the Forward1 class.

Table 5.7 Video classification result for DiagonalLeft movement.

Sample results of video output are demonstrated in Table 5.7 and 5.8 for the DiagonalLeft and Forward2 classes as inspired by (Saikia & Das, 2013). Moreover, human activity class names with their precision rates are embedded in video as subtitle. Thus, new video is generated with class subtitle as output video.

Table 5.8 Video classification result for Forward2.

It is difficult to compare the performance of our proposed model with other studies in the literature because of the lack of works with similar combinations of deep learning architectures in the feature extraction step as in the HDM model, and also due to the newly

produced datasets. Although the datasets employed in the experiments are different, the performance of the CNN-OF model can be compared with the state-of-the-art study in (Latah, 2017). For the CNN-OF model which is the same as that in (Latah, 2017), the classification accuracies of our system suggest almost the same classification performance with 90.5% accuracy on MCDS and 90.3% accuracy on CDS while the study presented in (Latah, 2017) yields 90.34% accuracy results. For the CNN-OF model, the performance of the system is consistent because of the similarity of the accuracy results in both studies (ours and (Latah, 2017)) even if the datasets are different. On the other hand, our proposed HDM model outperforms the results of (Latah, 2017; Aires, Santana & Medeiros, 2008) which yield accuracies of 90.34% of and 86.10%, respectively, while our model exhibits 95.2% accuracy on the MCDS dataset and 91.3% accuracy on the CDS dataset.

5.3. Conclusion, Use Cases and Future Studies

Human activity recognition is a challenging problem to be solved yet. It is in used in many applications in fields such as autonomous driving, visual surveillance, human-computer interaction, and entertainment. There are many possible motion estimation approaches to overcome this issue.

In this study, we introduced a multi-stream hybrid deep model for HAR. The proposed architecture is built combining a dense optical flow approach and auxiliary movement information in videos using deep learning methodologies. First, deep learning models, namely 3D convolutional neural network (3D-CNN) with multiple frames, 3D-CNN with optical flow, long short-term memory network (LSTM) are combined to determine the motion vectors. Support vector machine algorithm is performed for video classification. Extensive comparative experiments are conducted with two newly generated chess datasets, namely magnetic wall chess board video dataset (the MCDS), and standard chess board video dataset (CDS). So we demonstrated the contributions of the proposed study. Finally, the experimental results show that the proposed multi-stream hybrid deep model exhibits considerable classification success compared to the state-of-the-art studies.

In conclusion, the experimental results demonstrate that the proposed architecture represents significant advantages for recognizing and classifying human activities in videos. First, the proposed hybrid deep model is flexible, extendable and customizable as

it is able to determine many complex activities in various video datasets including playing chess, playing checkers, playing peg solitaire, playing football and playing cards. Second, any number of features even if it has multiple data format such as voice or text can be easily consolidated as auxiliary information for the proposed architecture. Third, proposed multi-stream hybrid deep model allows the fusion of optical flow and auxiliary information which has multiple data format such as voice, text and visual. In addition to these advantages, the proposed hybrid deep model architecture allows the connection of other deep learning models to our proposed model as auxiliary features for purposes such as object recognition, hand tracking, and so on.

As future work, we plan to enrich the set of features by employing other deep learning techniques.

6. CHESS PLAYING ROBOT SIMULATOR WITH V-REP

6.1 V-REP Robot Simulator

V-REP is the Virtual Robot Experiment Platform where you can create the physical and software model of any robotic system. It is a product of Coppelia Robotics (http://www.coppeliarobotics.com/index.html acces date: 20.05.2017). The training version is available for free. V-REP is cross-platform and works on Windows, Linux and MAC operating systems. Seven different languages such as C/C++, Python, Java, Matlab, Octave, Lua & Urbi can be programmed in V-REP. We developed our application with Lua script. We used ABB IRB360 delta robot as a robot model. Since the 3-D model of the robot in the V-REP software environment provides inverse and forward kinematic equations, it is sufficient to develop the communication interface software using Lua script language to simulate robot movements.

In this work, we developed a simulation infrastructure for chess-playing delta robot. This simulator is capable to be controlled by an external system via remote API.

6.2 Design

In this study, the software developed on Windows operating system is as below :

- **V-REP Communication : (.dll) :** Transfer commands read from the file to the V-REP tool with remoteApi.
- **V-REP Lua Scripts :** GUI : ABB IRB360 robot model takes commands via remoteApiCommandServer, then it moves the chess pieces on the screen via threads.

6.3 Software Implementation

In this paper, software programs have been implemented for simulation and communication We can classify the developed software in three different modules.

A. chess_v_rep_kinect.ttt :

The V-REP screen consists of the following parts :

- Lua scripts
- Delta robot model
- Models of chess pawns
- Chessboard model

- Remote Api Command Server

The simulation on V-REP can work independently. It can also be controlled by external system via Remote API.

B. try_v_rep.dll remoteApi.dll:

This library is written in C++ to transfer commands from an external system to V-REP. It provides integration between V-REP and external system.

Remote Api :

The remote API is part of the V-REP API. V-REP uses a remote API allowing to control a simulation from an external application or a remote hardware (e.g. real robot, remote computer,e.t.c).

Remote API functions interact with V-REP via socket communication. The Remote API functionality has two separate parts :

- The Client Side (your application) : Many different programming languages can be used to implement the remote API on the client side. Following languages are supported currently : C/ C++, Python, Java, Matlab, Octave, Urbi and Lua.
- The server side (V-REP) : By default, the server-side Remote API has been implemented with a plugin that is loaded by V-REP.

We used following functions in communication :

- simxSetIntegerParameter : try_v_rep.dll (remoteApi.dll) uses this function on the client side.
- simGetInt32Parameter : V-REP uses this function on the server side to receive commands by the client side.

User Interface / V-REP :

In this project, V-REP was used to implement robotic simulator (Figure 6.1).

chess_v_rep_kinect.ttt : The V-REP screen consists of the following parts :

- Lua scripts
- Delta robot model (ABB IRB360 robot model)
- Chessboard model
- Chess pieces model

- remoteApiCommandServer

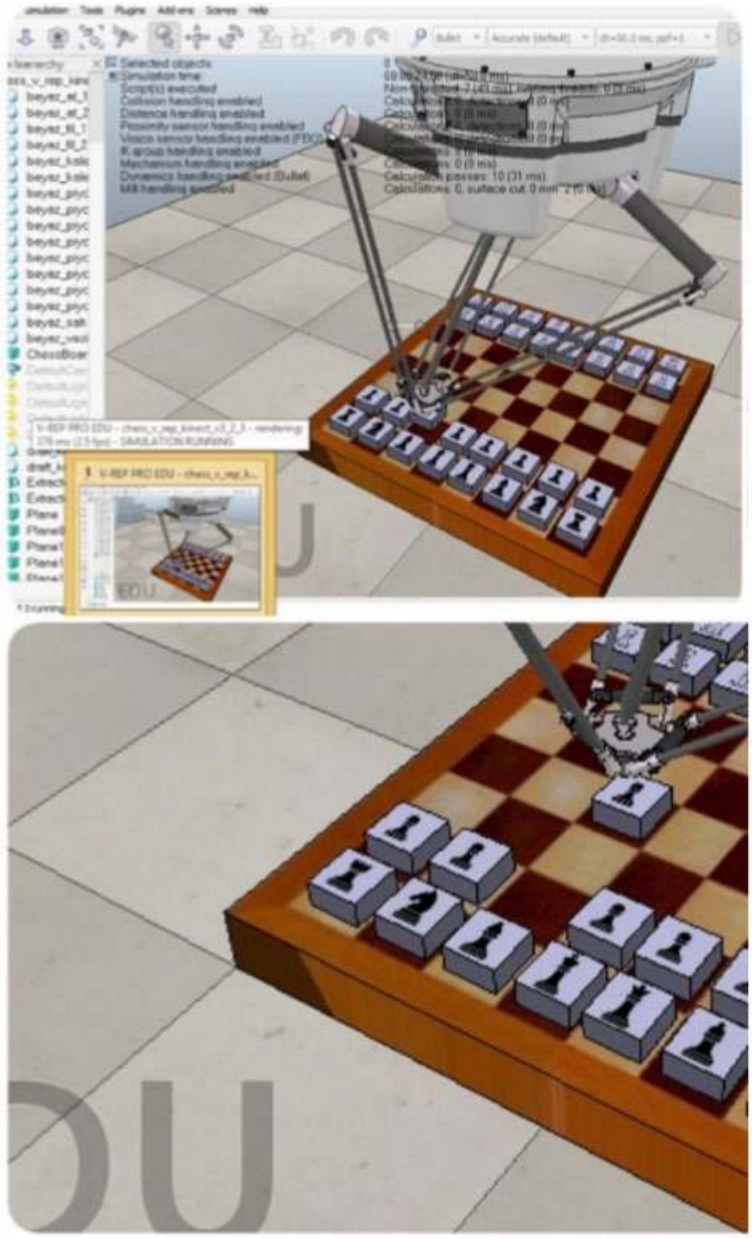


Figure 6.1 Developed software interface.

6.4 Chess Playing Robot Simulation

In this project, we developed a robotic simulator infrastructure which can be run standalone or controlled by an external system. Meaningful primitive hand movements made by the chess player can be sent to the V-REP via the remote API which will communicate with the robot simulator, and then simulator runs. We tested simulator system for Reti and English opening scenario.

7. A SIMPLE DL VIDEO ANALYTICS FRAMEWORK FOR HUMAN MOTION IMITATION

7.1 Introduction

In recent years, massive amounts of video data are produced every moment via cameras in public places. This data is unstructured and includes lots of knowledge regarding real world events. With video analytics, video data can be turned into useful and valuable information. Video analytics digitally analyzes video inputs. It transforms them into usable intelligent data so it helps to decision making for specific business rules.

Video analytics usually uses multidisciplinary algorithms to monitor, analyze and manage a huge volume of video data automatically. Supported by Deep Learning and AI, video analytics is able to do object recognition, object tracking, classification of objects or attributes like type, color, speed, and direction, and video indexing, so that videos can be searchable and understandable. It makes it possible to take some actions can be taken for safety, security, and operational decision making. It also makes it possible to receive early warning about specific situations with potential risks. It helps to analyze customer behavior and improve customer experience.

Analysis of a huge amounts of video data is a big challenge for video management. On video analytics market type, there are lots of application like intrusion management, incident detection, traffic monitoring, people/crowd counting. Video analytics service types and cloud deployment are expected to grow at a higher rate near future. As mentioned in the reference (https://www.researchandmarkets.com/reports/4530884/video-analytics-market-by-type-software-and access date: 01.03.2020), the video analytics market size is expected to grow from USD 3.23 billion in 2018 to USD 8.55 billion by 2023 (approximately 21 percent growth rate) (https://www.researchandmarkets.com/reports/4530884/video-analytics-market-by-type-software-and access date: 01.03.2020).

Some use cases for video analytics :

- Informing staff regarding some incidents : cars driving the wrong way, people moving in restricted areas. It gives alert about an incident.

- Retail management system : For example, in airports, it can provide queue analysis in check-in point, so it can help direct staff, it reduces waiting time for travelers.
- Retail management system : Sending customized advertisements, based on analysis of gender identification, age, capturing glance time on products.

In addition, there are lots of applications supported by video analytic system like smart surveillance, sport analysis and exercise training, medical, video content analysis and retrieval, video conferencing, and robotic. In our work, we designed a simple DL video analytic framework for a specific use-case that a robot in virtual environments imitates human motion in the video. In our particular scenario, human in the video plays chess.

7.2 Proposed DL Video Analytics Framework

The framework, depicted in Figure 7.1, which consists of four main layers, Video Data Processing Layer, Video AI/ML/DL Layer, Knowledge Layer and Service Layer respectively.

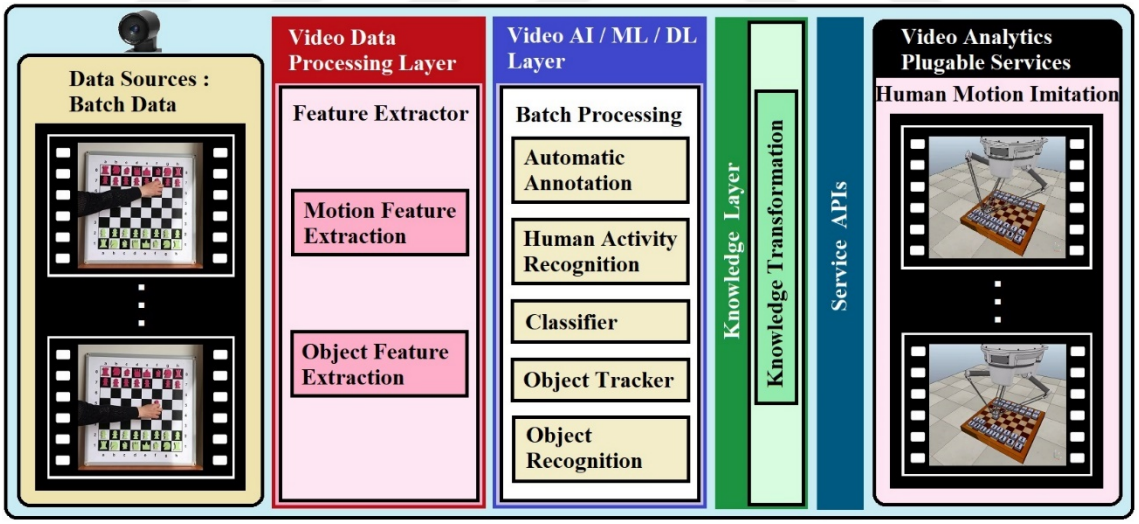


Figure 7.1 Proposed DL Video Analytics Framework for Human Motion Imitation.

The main contributions of this section are :

- We designed and developed a novel layered framework inspired by SIAT (Uddin, Alam, Tu et al., 2019) for human motion imitation use-case. It has offline batch processing for videos.

- We used our introduced hybrid deep learning model for human activity recognition in the video analytics framework.
- We performed experiments for application scenarios in offline mode to ensure effectiveness.

7.2.1. Video Data Processing Layer

In the beginning, for the offline video data processing, batch video data is prepared for AI/ML/DL computing. In our work, this layer is mainly in charge of processing image and video raw data. Feature extractor component on this layer is also responsible to extract important features of image and video data. It is composed of two main extractions, namely, object feature extraction and motion feature extraction.

In this work, we implemented feature extraction step with Python and OpenCV libraries in our proposed hybrid deep model (Tanberk, Kilimci, Tukel et al., 2020). Object feature extraction is built by using TensorFlow Object Detection API in (Francis, 2017) and patterns in chess boards in (https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html#re-projection-error access date: 25.05.2019). Motion feature extraction is built by using real time hand detector in (Dibia, 2017). The input of this step is a microvideo dataset consisting of meaningful parts of movement made by the chess player. The output of this step is a file consists of feature tensors. This feature files are prepared to be consumed by AI/ML/DL.

DL video analytics framework is very modular and customizable. It makes development life-cycle easy and simple. Feature extractor component can be improved, so that new feature extraction algorithms can be built and integrated with the platform based on business specifications. Video data processing layer can be improved also. For example, video segmentation algorithms can be built and added to the system, then long sequence of videos can be analysed also.

7.2.2. Video AI/ML/DL Layer

Video AI/ML/DL layer is responsible for generating semantic outcomes from extracted features by video data processing layer. Object recognition refers chess pawn recognition in our microvideo dataset (Francis, 2017). Object tracker refers hand detector in (Dibia, 2017). We used SVM as classifier to recognize human activities and objects in

the video. Human activity recognition refers to classifying human activities and finding direction of movement in the video. Automatic annotation refers to automatically adding subtitles including object names and activities performed on the video. The result of video AI/ML/DL layer is consumed by knowledge layer as high-level information.

Algorithms of all elements in this layer are deep learning based. TensorFlow Object Detection API (Francis, 2017) has been used for object recognition. According to our work, object in this API refers to chess pawn. Hand-detector using neural networks (SSD) on Tensorflow (Dibia, 2017) has been used for object tracking. According to our work, object in this API refers to hand. Hybrid deep model (Tanberk, Kilimci, Tukel et al., 2020) has been used for human activity recognition. According to our work, human activity in this deep model refers to a meaningful movement part by the chess player, namely motion primitive.

Deep learning methods are very efficient in extraction hierarchical features from raw data. They are very successful in achieving meaningful results with high accuracy from the data. Due to these characteristics, DL has attracted the attention of researchers in recent years and has been used in many applications.

7.2.3. Knowledge Layer

Knowledge layer is dedicated to representing information in framework to be used by external business services. In our work, it is utilized to solve complex tasks such as human motion imitation. It analyzes videos with AI/ML/DL techniques in previous layer and computes knowledge representation. Then it transforms knowledge to information to be used external human motion imitation service. Transformed information in this layer refers an array including activity class and positions.

7.2.4. Service Layer

Service layer is built on the top of low-level AI/ML/DL APIs. In this work, we designed our framework as platform as a service. So external business services don't need to know insight in framework. In this way, end users can use the framework including video analytics implemented by AI/ML/DL for complex tasks via customized services. From this point of view, the main advantage of the framework is that the developer can easily build high level modules using these domain specific service APIs without

considering their insights. Furthermore, new customized APIs can be built easily and can be integrated with the system.

7.3. Robot Grasp Execution

One of main goals of robotic applications is to repreduce human movement on robotic platforms. We applied motion primitive-based approaches in our work. After mapping real world state to motion primitives, the primitives can be composed and sequenced consecutively. So humanlike motion can be created and then complex tasks can be solved by the robotic applications. In this work, we present a human motion imitation solution prototyping, focusing specifically on a chess playing robotic application.

We have the infrastructure for controlling the delta robot (Tanberk & Tukel, 2017) with V-REP robotic simulation tool. This tool in (Tanberk & Tukel, 2017) has the ability to communicate via V-REP remote API. This remote API allows to control the simulation in V-REP from an external application. We used the delta robot in (Tanberk & Tukel, 2017) as a reproduction part of human motion imitation system. It is one of video analytics plugable services.

We formulated motion primitives as human activities recognized by DL video analytics framework. With service API in the framework, motion in microvideo recognized by DL has been taken and then transferred to the delta robot (Tanberk & Tukel, 2017). Algorithm was tested according to one chess opening scenario.

7.4. Conclusion, Use Cases and Future Studies

In this work, we proposed a robust, pluggable, layered and service-oriented simple DL video analytics framework via customizing the framework in (Uddin, Alam, Tu et al., 2019). We specifically worked and contributed to human activity recognition and human motion imitation for video analytics framework. Motion in input microvideo has been recognized with hybrid deep model in (Tanberk, Kilimci, Tukel et al., 2020) inside proposed framework. Then recognized motion primitive by DL has been transferred to delta robot in V-REP chess play system in (Tanberk & Tukel, 2017). One of chess opening scenario using proposed simple DL video analytics framework has been run for evaluation and experimental reasons. This work is a prototype for DL video analytics framework for human motion imitation domain.

This DL video analytics framework for human motion imitation is ongoing research project. It needs further experiments and DL models in it to make it more efficient. In the future, we can add video segmentation DL model, video indexing and retrieval with DL model, background substraction component, and so on. So motions in long video stream can be recognized and transferred as motion primitives to robotic system to imitate human motion. It enables to imitate motions in long video stream by robotic system via composing and sequencing motion primitives. Besides, in the future, it can be run online also. It also can be connected to big data and cloud-based system.

8.CONCLUSION AND FUTURE STUDIES

In this thesis, the problem of human activity recognition has been studied with deep learning approaches in the core and a DL powered video analytics framework has been proposed for human motion imitation. The aim of the proposed video analytics framework is to create a flexible, pluggable and adaptable human-robot imitation interfaces.

A novel multi-stream hybrid deep model is proposed for the purpose of HAR. The proposed architecture combines multiple frames and dense optical flow approaches, and then auxiliary movement information, over video frames. 3D-CNNs fed by multiple frames and optical flow and LSTM fed by auxiliary information are combined to extract motion features. Video classification task for recognizing human activities is processed by SVM. This study also provides a basic human activity recognition pipeline in videos. Then, this human activity recognition pipeline including multi-stream hybrid deep model can be adapted for different activities in terms of preprocessing, feature extraction, and feature representation steps.

Two new datasets, namely magnetic wall chess board video dataset (the MCDS), and standard chess board video dataset(CDS), are captured and generated. These datasets consist of microvideos that contain the moves of the chess player. DiagonalLeft, DiagonalRight, Forward1, and Forward2 features are extracted and they are used for classification. The classification performance is measured by precision, accuracy, recall, and F-measure metrics. The experimental results show that the proposed multi-stream hybrid deep model exhibits considerable classification success compared to the state-of-the-art studies.

A new robotic chess simulator infrastructure is developed by V-REP. The delta robot in simulator can run standalone or being controlled by an external system. We tested the chess simulator for Reti and English chess opening scenarios.

An early version of deep learning video analytics framework is proposed for human motion imitation with motion primitive approach. This framework uses the multi-stream hybrid deep model to recognize human activities during video analysis. It also provides knowledge to external domains for human activity recognition task with the help of API. So robotic chess simulator gets and runs the information for imitation via the ai-powered

framework. By performing an example of the task by imitating the human with the robotic chess simulator, we carried the moves of the chess player in the videos from human activity recognition to robotic grasping.

This thesis can play a significant role in the studies such as video analysis and robotic imitation. The proposed multi-stream hybrid deep model and DL video analytics framework and the findings of the thesis can lead to improve applications such as video understanding, assisted technologies, security applications, smart home, remote robot controller, and interactive game.

Video data volume has increased in recent years and it is expected to increase in the future. Thanks to the developing AI technology, video analysis is gaining importance. In the future, the number of robotic devices will increase and this type of devices will be able to imitate people in remote location by using ai-powered video analytics framework.

REFERENCES

Agahian S.(2018). Action recognition from 3D human movements with spatio-temporal bag-of-poses. (PhD Thesis). Access address: http://acikerisim.ktu.edu.tr/jspui/handle/123456789/317.

Aggarwal C.C.(2014). *Data Classification : Algorithms and Applications*. Chapman and Hall/CRC.

Aires K. R., Santana A. M. and Medeiros A. A. D. D.(2008). Optical flow using color information: Preliminary results. *Proc. ACM Symp. Appl. Comput.*, vol. 3, pp. 1607–1611.

Anandan P.(1989). A Computational Framework and an Algorithm for the Measurement of Visual Motion. *International Journal of Computer Vision*, vol. 2, pp. 283–310.

Anthwal S., and Ganotra D.(2018). Optical Flow Estimation in Synthetic Image Sequences Using Farneback Algorithm. *Advances in Signal Processing and Communication*, pp. 363-371.

Argall B.D., Chernova S., Veloso M.M., and Browning B.(2009). A survey of robot learning from demonstration. Robotics and Autonomous Systems. 57(5):469–483.

Arif S., Wang J., Hassan T. and Fei Z.(2019). 3D-CNN-Based Fused Feature Maps with LSTM Applied to Action Recognition. *Future Internet*, vol. 11, no. 2.

Beauchemin S. S., and Barron J. L.(1995). The computation of optical flow. *ACM Computing Surveys*, vol. 27, no. 3.

Chalodhorn R., Grimes D.B., Grochow K., Rao R.P.N.(2007). Learning to walk through imitation. *IJCAI*, vol. 7, pp. 2084–2090.

Cheng G., Wan Y., Saudagar A.N., Namuduri K. and Buckles B.P.(2015). Advances in human action recognition: A survey. *Computer Vision and Pattern Recognition*.

Chuankun L., Hou Y., Wang P. and Li W.(2018). Multi-view Based 3D Action Recognition Using Deep Networks. *IEEE Transactions on Human-Machine Systems*, pp. 1-10.

Cole J.B., Grimes D.B., Rao R.P.N., Learning full-body motions from monocular vision: dynamic imitation in a humanoid robot(2007). *Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems*, pp. 240–246.

Dariush B., Gienger M., Arumbakkam A., Zhu Y., Jian B., Fujimura K., Goerick C.(2009). Online transfer of human motion to humanoids. *Int. J. Humanoid Robot*, vol. 6, pp. 265–289.

Dariush B., Gienger M., Jian B., Goerick C. and Fujimura K.(2008). Whole body humanoid control from human motion descriptors. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2677–2684.

Dev D.(2017). *Deep Learning with Hadoop*. Birmingham, UK : Packt Publishing; 70-99.

Dibia V.(2017). “How to Build a Real-Time Hand-Detector Using Neural Networks (SSD) on Tensorflow”. Access address: https://towardsdatascience.com/how-to-build-a-real-timehand-detector-using-neural-networks-ssd-on-tensorflow-d6bac0e4b2ce. Access date: 27.06.2018.

Dollár P., Rabaud V., Cottrell G. and Belongie S.(2005). Behavior recognition via sparse spatio-temporal features. *International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 65-72.

Farnebäck G.(2003). Two-Frame Motion Estimation Based on Polynomial Expansion. *Scandinavian Conference on Image Analysis*, pp. 363-370.

Francis J.(2017). “Object detection with TensorFlow”. Access address: https://www.oreilly.com/content/object-detection-with-tensorflow. Access date: 27.06.2018.

Gams A., Nemec B., Ijspeert A.J., Ude A.(2014). Coupling movement primitives : interaction with the environment and bimanual tasks. *IEEE Trans. Robot,* vol. 30, pp. 816–830.

Gao L., Li X., Song J., and Shen H.(2019). Hierarchical LSTMs with Adaptive Attention for Visual Captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Greff K., Srivastava R.K., Koutnik J., Steunebrink B.R., and Schmidhuber J.(2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*. vol. 28, no. 10, pp. 2222–2232.

Grimes D.B., Chalodhorn R., Rao R.P.N.(2006). Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. *Proceedings of Robotics: Science and Systems*.

Harada K., Miura K., Morisawa M., Kaneko K., Nakaoka S., Kanehiro F., Tsuji T., Kajita S.(2009). Toward human-like walking pattern generator. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1071–1077.

Heeger D.J.(1988). Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, vol. 1, pp. 279–302.

Hochreiter S., and Schmidhuber J.(1997). Long short-term memory. *Neural computation,* vol. 9, pp. 1735-1780.

Horn B.K.P., and Schunck H.G.(1981). Determining Optical Flow. *Elsevier B.V.*, vol. 17, pp. 185-203.

Hossain M.A.(2014). Framework for a Cloud-Based Multimedia Surveillance System. *Int. J.Distrib. Sens. Netw,* vol. 2014.

Hu C., Xue G., Mei L., Qi L., Shao J., Shang Y., et al.(2017). Building an intelligent video and image analysis evaluation platform for public security. *Proceedings of 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6.

Ikram S.T., and Cherukuri A.K.(2016). Improving Accuracy of Intrusion Detection Model Using PCA and Optimized SVM. CIT. *Journal of Computing and Information Technology*, vol. 24, no. 2, pp. 133–148.

İnik A., and Ülker E.(2017). Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri. *Gaziosmanpasa Journal of Scientific Research*. vol. 6, no. 3, pp. 85-104.

Karpathy A., Toderici G., Shetty S., Leung T., Sukthankar R., and Fei-Fei L.(2014). Large-scale video classification with convolutional neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1725-1732.

Ke Q., Liu J., Bennamoun M., An S., Sohel F., and Boussaid F.(2018). Computer Vision for Human–Machine Interaction. *Computer Vision for Assistive Healthcare*, pp. 127-145.

Keçeli, A. S.(2018). Viewpoint projection based deep feature learning for single and dyadic action recognition. *Expert Systems with Applications*, vol. 104, pp. 235-243.

Keçeli A. S., Kaya A., and Can A.B.(2018). Combining 2D and 3D deep models for action recognition with depth information. *Signal, Image and Video Processing*, vol. 12, pp. 1197-1205.

Kent D., and Salem F.(2019). Performance of three slim variants of the long short-term memory (LSTM) layer. *IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 307-310.

Kim J.-Y., Kim Y.-S.(2013). Whole-body motion generation of android robot using motion capture and nonlinear constrained optimization. *Int. J. Humanoid Robot*. vol. 10.

Kober J., Bagnell J.A., Peters J.(2013). Reinforcement learning in robotics: a survey. *Int. J. Robot. Res*. vol. 21, pp. 1238–1274.

Krizhevsky A., Sutskever I., and Hinton G.E.(2012). Imagenet classification with deep convolutional neural networks. *NIPS'12: Proceedings of the 25th International Conference on Neural Information Processing Systems*. vol. 1. pp. 1097–1105.

Kulić D. (2019). Humanoid Robotics: A Reference. *Springer*, Dordrecht, pp. 1657-1677.

Kuli´c D., Ott C., Lee D., Ishikawa J., Nakamura Y.(2012). Incremental learning of full body motion primitives and their sequencing through human motion observation. *Int. J. Robot. Res*. vol. 31, pp. 330–345.

Kuli´c D., Takano W., Nakamura Y.(2008). Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden Markov chains. *Int. J. Robot. Res*. vol. 27, pp. 761–784.

Kuli´c D., Takano W., Nakamura Y.(2009). On-line segmentation and clustering from continuous observation of whole body motions. *IEEE Trans. Robot*. vol. 25, pp. 1158–1166.

Laptev I.(2005). On space-time interest points. *International Journal of Computer Vision (IJCV)*, vol. 64, no. 2–3, pp. 107–123.

Laptev I., and Lindeberg T.(2003). Space-time interest points. *Proc. Ninth IEEE International Conference on Computer Vision (ICCV),* vol. 1, pp. 432-439.

Latah M.(2017). Human action recognition using support vector machines and 3D convolutional neural networks. *International Journal of Advances in Intelligent Informatics*, vol. 3, no. 1, pp. 47-55.

Lee H., Grosse R., Ranganath R., Ng A.Y.(2009). Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. *Proceedings of the 26th Annual International Conference on Machine Learning,* Montreal, Quebec. ICML 2009. 77. 10.1145/1553374.1553453.

LeCun Y., and Bengio Y.(1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*.

LeCun Y., Bengio Y., and Hinton G.(2015). Deep learning. *Nature.* 521(7553): 436-444.

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

Lin C.(2019). "Introduction to motion estimation with optical flow". Access address: https://nanonets.com/blog/optical-flow/. Access date: 01.03.2020.

Liu J., Luo J. and Shah M.(2009). Recognizing realistic actions from videos "in the wild". *IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR 2009)*, pp. 1996-2003.

Liu J., Shahroudy A., Xu D., and Wang G.(2016). Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition. *Computer Vision - Eccv 2016*, vol. 9907.

Liu Z., Zhang C. Y., and Tian Y.L.(2016). 3D-based Deep Convolutional Neural Network for action recognition with depth sequences. *Image and Vision Computing*, vol. 55, pp. 93-100.

Lucas B.D., and Kanade T.(1981). An Iterative Image Registration Technique with an Application to Stereo Vision. *Proceedings of the 7th international joint conference on Artificial intelligence (IJCAI'81),* vol. 2, pp. 674–679.

Ma Y., Guo G.(2014). *Support Vector Machines Applications*. Springer Publishing Company.

Mittal A.(2019). "Understanding RNN and LSTM". Access address: https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e. Access date: 01.03.2020.

Mombaur K., Truong A., Laumond J.P.(2010). From human to humanoid locomotion – an inverse optimal control approach. *Auton. Robot*. v. 28, pp. 369–383.

Nakoka S., Nakazawa A., Yokoi K., Hirukawa H., Ikeuchi K.(2003). Generating whole body motions for a biped humanoid robot from captured human dances. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3905–3910.

Nakaoka S., Nakazawa A., Yokoi K., Ikeuchi K.(2004). Leg motion primitives for a humanoid robot to imitate human dances. *J. Three Dimens. Images,* vol. 18, pp. 73–78.

Niebles J. C., Chen C. W. and Fei-Fei L.(2010). Modeling temporal structure of decomposable motion segments for activity classification. *Proc. European Conference on Computer Vision (ECCV 2010)*, vol. 6312, pp. 392–405.

Okamoto T., Shiratori T., Kudoh S., Nakaoka S., Ikeuchi K.(2014). Toward a dancing robot with listening capability: keypose-based integration of lower-, middle-, and upper-body motions for varying music tempos. *Trans. Robot*. vol. 30, pp. 771–778.

Ostheimer D., Lemay S., Ghazal M., Mayisela D., Amer A., Dagba P.F.(2006). A modular distributed video surveillance system over IP. *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, pp. 518–521.

Ott C., Lee D., Nakamura Y.(2008). Motion capture based human motion recognition and imitation by direct marker control. *Proceedings of the IEEE International Conference on Humanoid Robots*, pp. 399–405.

Qian H., Mao Y., Xiang W., and Wang Z.(2010). Recognition of human activities using SVM multi-class classifier. *Pattern Recognition Letters*, vol. 31, no. 2, pp.100-111.

Ray S.(2017). "Understanding Support Vector Machine(SVM) algorithm from examples (along with code)". Access address: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code. Access date: 01.03.2020.

Ren S., He K., Girshick R., and Sun J.(2017). FasterR-CNN:Towardsreal-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149.

Romeo J.(2011). From human to robot grasping. (PhD Thesis). Access address: www.csc.kth.se/~jrgn/phd_thesis.pdf.

Rozo L., Calinon S., Caldwell D.G., Jimenez P., Torras C.(2016). Learning physical collaborative robot behaviors from human demonstrations. *IEEE Trans. Robot*. vol. 32, pp. 513–527.

Saha S.(2018). "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way". Access address: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. Access date: 01.03.2020.

Saikia P. and Das K.(2013). Head gesture recognition using optical flow based classification with reinforcement of GMM based background subtraction. Access adress: https://arxiv.org/abs/1308.0890.

Salcedo J.(2019). *Machine Learning for Data Mining*. Packt Publishing.

Simonyan K. and Zisserman A.(2014). Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, vol. 1, pp. 568-576.

Singh A.(1990). An estimation-theoretic framework for image-flow computation. *Proceedings Third International Conference on Computer Vision*.

Sivic J. and Zisserman A.(2003), Video Google: A text retrieval approach to object matching in videos. *Proceedings Ninth IEEE International Conference on Computer Vision*, vol. 2, pp. 1470-1477.

Song J., Yuyu, G., Gao L., Li X., Hanjalic A., and Shen H.(2017). From Deterministic to Generative: Multi-Modal Stochastic RNNs for Video Captioning. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30.

Sun J., Wang J. and Yeh T.(2017). "Video Understanding: From Video Classification to Captioning". Stanford University. Access address: http://cs231n.stanford.edu/reports/2017/pdfs/709.pdf, Accessed date: 27.06.2018.

Takano W., Yamane K., Nakamura Y.(2005). Primitive communication of humanoid robot with human via hierarchical mimesis model on the proto symbol space. *Proceedings of the IEEE/RAS International Conference on Humanoid Robots*, pp. 167–174.

Takano W., Yamane K., Sugihara T., Yamamoto K., Nakamura Y.(2006). Primitive communication based on motion recognition and generation with hierarchical mimesis model. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3602–3608.

Tanberk S., Kilimci Z., Tukel D., Uysal M. and Akyokus S.(2020). A Hybrid Deep Model Using Deep Learning and Dense Optical Flow Approaches for Human Activity Recognition. *IEEE Access*. vol. 8.

Tanberk, S., Tukel, D. (2017). Kinect controlled chess playing robot. *IEEE 17th International Conference on Smart Technologies*, Ohrid, Macedonia. pp. 594-598.

Uddin Md A., Alam A., Tu N.A., Islam Md S. and Lee Y.(2019). SIAT: A Distributed Video Analytics Framework for Intelligent Video Surveillance. Symmetry. vol. 11, no. 7.

Verma S.(2019). "Understanding 1D and 3D Convolution Neural Network | Keras". Access address: https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610. Access date: 01.03.2020.

Vrigkas M., Nikou C., and Kakadiaris I. A., A review of human activity recognition methods (2015). *Frontiers in Robotics and Artificial Intelligence*, vol. 2, no. 28.

Wang X, Gao L., Song J., and Shen H.(2017). Beyond Frame-level CNN: Saliency-Aware 3-D CNN With LSTM for Video Action Recognition. *IEEE Signal Processing Letters*, vol. 24, pp. 510-514.

Wang X., Gao L., Wang P., Sun X., and Liu X.(2018). Two-stream 3D convNet fusion for action recognition in videos with arbitrary size and length. *IEEE Transactions on Multimedia*, vol. 20, pp. 634 - 644.

Wang H., Kläser A., Schmid C. and Cheng-Lin L.(2011). Action recognition by dense trajectories. *IEEE Conference on Computer Vision & Pattern Recognition (CVPR 2011)*, pp. 3169-3176.

Wang H., Ullah M. M., Klaser A., Laptev I. and Schmid C.(2009). Evaluation of local spatio-temporal features for action recognition. *British Machine Vision Conference (BMVC 2009).*

Wang H., Wang P., Song Z. and Li W.(2017). Large-scale multimodal gesture recognition using heterogeneous networks. *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 3129-3137.

Wlodarczak P.(2020). *Machine Learning and Its Applications*. CRC Press.

Yamane K., Nakamura Y.(2003). Dynamics filter – concept and implementation of online motion generator for human figures. *IEEE Trans. Robot. Autom*. v. 19, pp. 421–432.

Zelkowitz M.V.(2010). *Advances in Computers* : Academic Press.

Zhang H. B., Zhang Y. X., Zhong B., Lei Q., Yang L., Du J.X. et al.(2019). A comprehensive survey of vision-based human action recognition methods. *Sensors*, vol. 19, no. 5.

http://people.cs.pitt.edu/~chang/265/C5/v2.htm. Access date: 01.03.2020.

http://www.coppeliarobotics.com/index.html. Access date: 20.05.2017.

https://cloud.google.com/video-intelligence. Access date: 30.03.2020.

https://developer.nvidia.com/discover/lstm. Access date: 01.03.2020.

https://missinglink.ai/guides/convolutional-neural-networks/understanding-3d-cnn-uses. Access date: 01.03.2020.

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html#re-projection-error. Access date: 25.05.2019.

https://www.eagleeyenetworks.com. Access date: 30.03.2020.

https://www.ibm.com/cloud. Access date: 30.03.2020.

https://www.intellivision.com. Access date: 30.03.2020.

https://www.researchandmarkets.com/reports/4530884/video-analytics-market-by-type-software-and. Access date: 01.03.2020.

CURRICULUM VITAE

Senem Tanberk was born in Düzce, Turkey, in 1976. She received the B.S. degree in control and computer engineering from Istanbul Technical University, in 1998, and the M.S. degree in mechatronics from Marmara University, in 2013. She started Ph.D program in Dogus University Computer Engineering section in 2014. She is also an IT Professional in telecommunications, finance, and banking sector for over 15 years in Istanbul. She has special expertise in Telco pricing and billing, revenue and financial reports, DWH support, software design and development, and advanced PL/SQL (Oracle) perfection. She is also an Oracle PL/SQL Trainer in Telco. She has been a Research and Development Consultant in a private company, Istanbul. Her research interests include deep learning, machine learning, big data, data science, robotic programming, embedded programming, simulation programming, and graphical programming.