Deep Video Analysis Methods for Surgical Skills Assessment in Cataract Surgery

by

Ummey Tanin

A thesis submitted to the
Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Computer Science
Carleton University
Ottawa, Ontario
May, 2022

The undersigned hereby recommends to the
Faculty of Graduate and Postdoctoral Affairs
acceptance of the thesis

Deep Video Analysis Methods for Surgical Skills Assessment in Cataract Surgery

submitted by **Ummey Tanin**

in partial fulfillment of the requirements for the degree of

Master of Computer Science

Professor Matthew Holden, Thesis Supervisor

Professor Majid Komeii, School of Computer Science

Professor Jochen Lang,
School of Computer Science, University of Ottawa

Professor Alan Tsang, Chair,
School of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Computer Science
Carleton University
May, 2022

Abstract

It is important for a graduate surgical trainee in ophthalmology to have a strong understanding of how to proficiently perform cataract surgery. The surgical training curriculum should incorporate methodical assessments of surgical skills and improve trainee surgeons' expertise to maintain patient safety. Prior rating scales for cataract surgery are highly dependant on the subjective opinion of the observing grader and are time consuming. This project is intended to develop a deep learning model for skill evaluation in cataract surgery using raw surgery videos that can supplement human review. An advanced convolutional neural network model is leveraged in this work and was evaluated using a large custom dataset. Videos from four phases in cataract surgery were used to quantify the model performance. Our model yielded an average accuracy of 82% for all four phases of cataract surgery.

Acknowledgments

I would like to show my gratitude to my mentor and supervisor Professor Matthew Holden. I have been extremely blessed to be guided and encouraged by you. You are resourceful, knowledgeable, kind and supportive. I cannot hope to have anyone better to be my supervisor.

I would like to be grateful to all of my course instructors for transferring their knowledge to us and enriching our understanding of the course materials with interactive resources. I would like to thank the examination committee for reviewing the thesis report and assisting me with their valuable feedback.

I want to thank Compute Canada for their support.This work was supported, in part, by the Natural Sciences and Engineering Research Council (NSERC) of Canada (grant information: RGPIN-2020-05582).

Table of Contents

Abstract **iii**

Acknowledgments **iv**

Table of Contents **v**

List of Tables **viii**

List of Figures **ix**

1 Introduction **1**

1.1 Cataract Surgery 1

1.2 Skill Assessment in Surgeries 5

1.3 Computer Vision in Medical Imaging 5

1.4 Deep Learning in Medical Data Analysis 6

1.5 Action Recognition for Video Analysis 6

1.6 Objectives 6

2 Related Work **10**

2.1 Classical Approach for Video Analysis 11

2.2 Deep Learning Algorithms for Video Analysis 13

2.3 Algorithms for Surgical Tool and Phase Analysis 16

2.4 Approaches for Surgical Skill Evaluation using Tool Analysis 20

2.5 Models for Skill Assessment using Surgical Video 24

2.6 Summary 28

3 Background **30**

3.1 Convolutional Neural Network 30

3.1.1 Convolution Layer . . . 31
3.1.2 Activation function . . . 32
3.1.3 Pooling Layer . . . 32
3.1.4 Fully Connected Layer . . . 33
3.1.5 2D Convolution across input channels . . . 33
3.1.6 3D Convolution Across Depth . . . 34
3.1.7 1×1 Convolution . . . 35
3.1.8 Dropout . . . 36
3.1.9 Batch Normalization . . . 36
3.2 Recurrent Neural Network . . . 36
3.2.1 Long Short-Term Memory (LSTM) . . . 37
3.3 Histogram of Oriented Gradients (HOG) . . . 38

4 Methodology **40**
4.1 Model Design . . . 40
4.1.1 Basic 2D CNN with LSTM . . . 40
4.1.2 2D CNN-LSTM with Inception v4 Modules . . . 42
4.1.3 Feature Extraction Using 3D Convolutional Neural Network . . . 45
4.1.4 Model Ensembling . . . 46

5 Dataset And Preprocessing **48**
5.1 Cataract Dataset . . . 48
5.2 Dataset Preparation . . . 49
5.3 Data Processing Techniques . . . 51
5.3.1 Up-sampling of minor class . . . 51
5.3.2 Image Resolution . . . 51

6 Implementation Details **53**

7 Results and Discussion **56**
7.1 Discussion . . . 60
7.2 Interpretation . . . 64
7.3 Limitation . . . 65

8 Conclusion and Future Work **67**

List of References **69**

List of Tables

7.1 Model performance under various network configuration 57
7.2 Skill assessment accuracy on all four phases 57
7.3 Train accuracy for skill assessment on all four phases across five folds 57
7.4 Test accuracy for skill assessment on all four phases across five folds . 57
7.5 Sensitivity and AUC on Test Set 58

List of Figures

1.1 Sectional view depicting the layers of a lens [1] 2
1.2 Cataract with a brown nucleus. [2] 2
1.3 Video frame from Capsulorhexis phase in our dataset [58]. 3
1.4 Video frame from Hydrodissection phase in our dataset [58]. 4
1.5 Video frame from Viscoelasticity phase in our dataset. [58]. 4
1.6 Video frame from Phacoemulsification phase in our dataset [58]. . . . 4
1.7 Traditional workflow for manual skill assessment in cataract surgery. . 7
3.1 Basic structure of a CNN 31
3.2 Convolution Operation 32
3.3 2D convolution operation across channels 34
3.4 3D Convolution operation 35
3.5 Simple RNN 37
3.6 Single LSTM layer 38
4.1 Proposed Ensemble Model 41
4.2 2D CNN-LSTM model with LSTM layer 41
4.3 2D CNN LSTM model with modules from Inception V4 43
4.4 Basic Inception module 44
4.5 Inception A module 44
4.6 Reduction module 44
4.7 Extraction of multiple features from contiguous frames in 3D CNN . 45
4.8 3D CNN 46
5.1 Dataset Preparation 50
7.1 Baseline Accuracy 59
7.2 Accuracy with L2 Regularizer. 59

7.3 Real time prediction of the model on all the 16-second video snippets representing a never seen before video sample (Model identifies novice). Video clip duration (e.g. clip 1 : 0-16 seconds, clip 2: 17-32 seconds) for all 16 second clips are plotted along x axis. 62
7.4 Real time prediction of the model on all the 16-second video snippets representing a never seen before video sample (Model identifies expert). 63

Chapter 1

Introduction

1.1 Cataract Surgery

Cataract is one of the common eye diseases in the world which accounts for total blindness of 47.8% worldwide [56]. 121 million was spent by the government of Canada for cataract surgeries in the year of 2003 adding up to 0.098% of health care disbursement [3]. Canadian ophthalmologists performed over 250000 cataract surgeries in 2003 and this can be accounted for 8000 cases per million patients. This rate is high for Australia (9000/million) and America (7000/million) as well. The need for surgeries related to cataracts is expected to surge in the coming years [4]. Millions of Canadians are living with cataracts these days which is the second leading cause of blindness and functional vision loss in Canada.

The crystalline lens of the human eye has a clear shape [19]. It is held in its normal position by ciliary fibers from the ciliary body (Figure 1.1). The crystalline lens comprises of a capsule, lens epithelium, cortex and nucleus. The lens refracts light to focus a clear image on the retina and provides accommodation. Cataract is the opacification of the crystalline lens that leads to visual impairment and can make a person's vision blurry.

Cataract surgery is referred to as lens replacement surgery. It is a well known surgical procedure for the removal of the natural lens of the eye that has formed an opacification [19]. The opacification is referred to as a cataract. During cataract surgery, it gets replaced with an intraocular lens. Aging is the most common cause of opacification. The use of tobacco products and exposure to ultraviolet radiation are the key factors that causes this visual impairment. Cataracts can appear in one

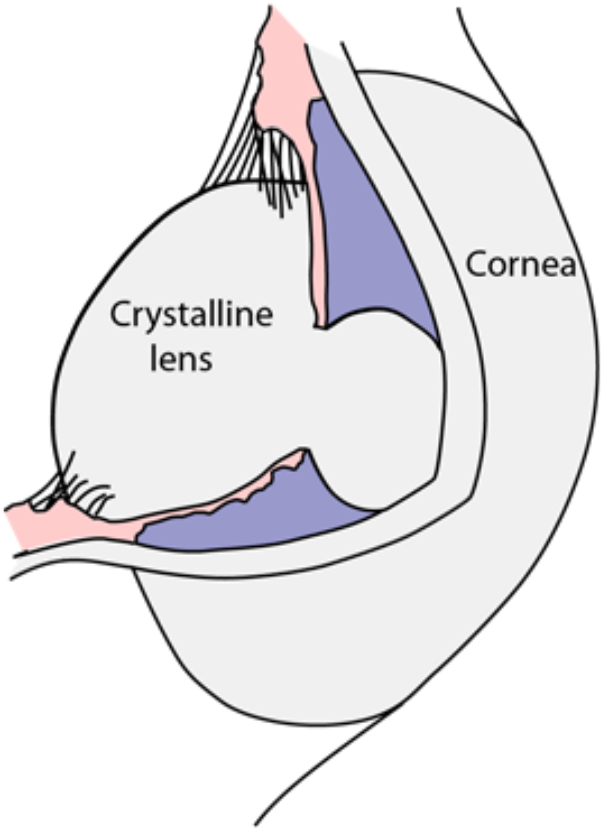

Figure 1.1: Sectional view depicting the layers of a lens [1]

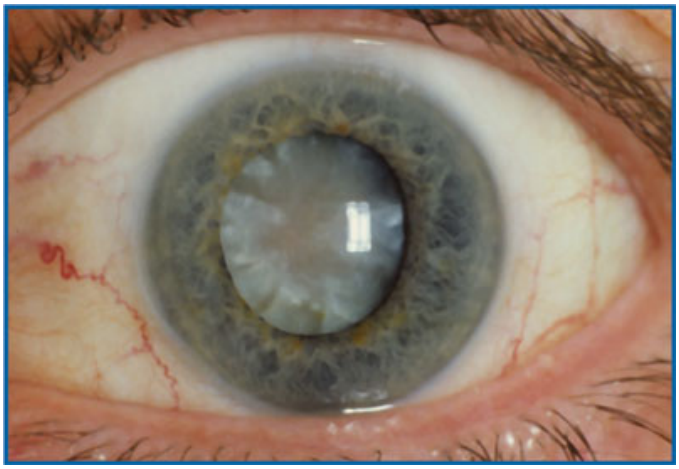

Figure 1.2: Cataract with a brown nucleus. [2]

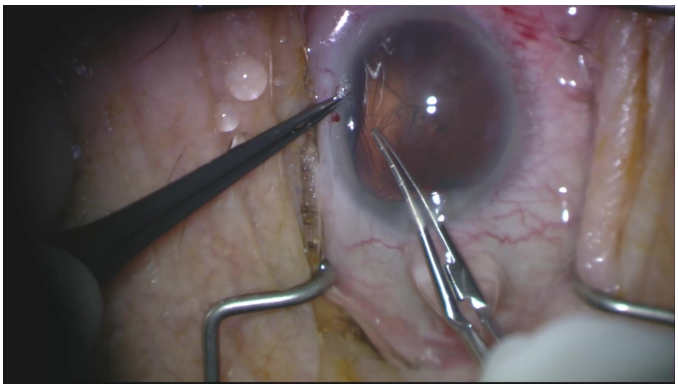

Figure 1.3: Video frame from Capsulorhexis phase in our dataset [58].

or both eyes. Some people might have cataract at a young age due to psychological trauma, diabetes and the use of steroid. It refers to an advanced or mature condition with vision impairment approaching blindness (Figure 1.2) [4].When this visual impairment becomes substantially noticeable, cataract surgery is the only recommended treatment.

There are 13 phases in cataract surgery. They are Paracentesis, Viscoelasticity, Wound, Capsulorhexis, Hydrodissection, Phacoemulsification, Viscoelasticity2, Intraocular Lens (IOL) Insertion, IOL Positioning, Viscoelastic removal, Hydration, Malyugin Ring Insertion and Removal. This work utilizes data from four phases which can be considered the most salient steps for identifying operative competence in cataract surgeries.

Capsulorhexis

Capsulorhexis or continuous curvilinear capsulotomy (CCC), is the most essential step in cataract surgery that can facilitate the surgical procedure. In this step, the capsule of the lens is removed from the eye by using stretch forces. It is necessary to perform other phases like Hydrodissection and Phacoemulsification (Figure 1.3).

Hydrodissection

Hydrodissection is the step that is performed right after Capsulorrhexis. It is the step where a fluid is injected to separate the lens capsule from the lens cortex (Figure

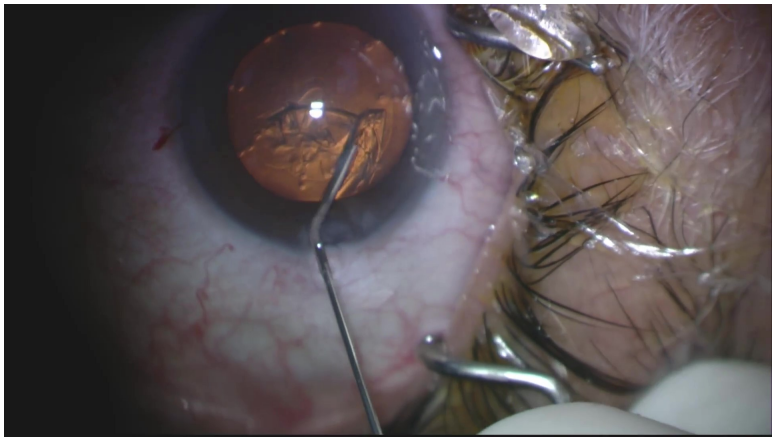

Figure 1.4: Video frame from Hydrodissection phase in our dataset [58].

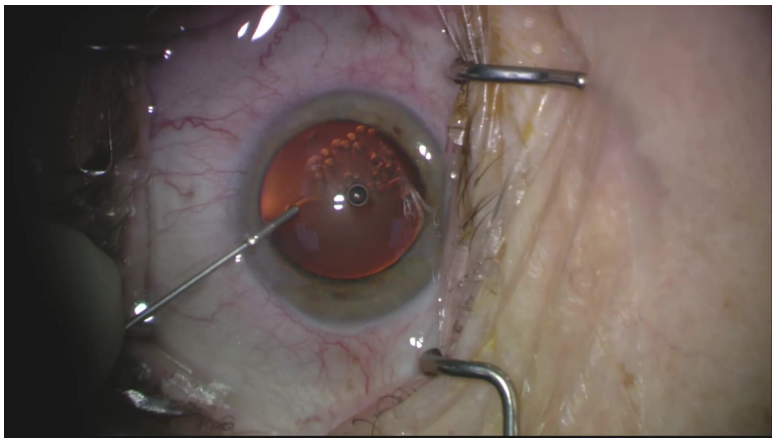

Figure 1.5: Video frame from Viscoelasticity phase in our dataset. [58].

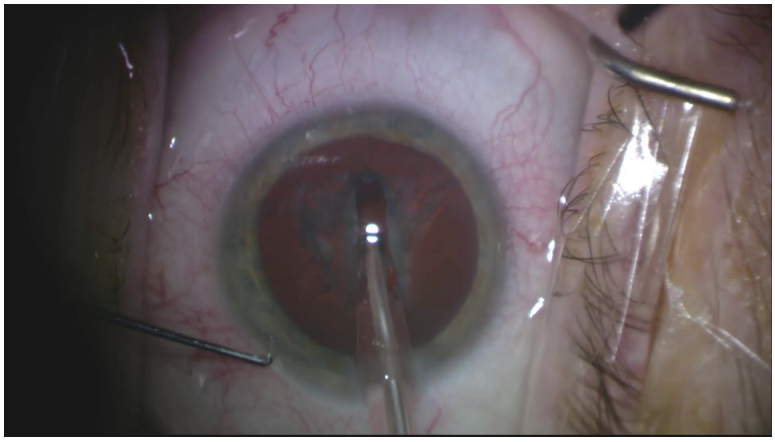

Figure 1.6: Video frame from Phacoemulsification phase in our dataset [58].

1.4).

Viscoelasticity

Viscoelasticity is preferred to make the completion of the Phacoemulsification step easier and safer. The viscoelastic agent is injected in the eye during cataract surgery to protect the corneal endothelium (Figure 1.5).

Phacoemulsification

Phacoemulsification is the phase where the cataract formed on the eye is emulsified by a needle tip that is inserted into the forward chamber of the eye. It is one of the most important steps that is used to remove the cataract (Figure 1.6).

1.2 Skill Assessment in Surgeries

Analyzing surgical videos to assess surgical competence is ongoing research in the domain of healthcare intervention. Surgical videos are processed in computer-assisted interventions to build surgical workflow and reports to support the surgeons with real-time recommendations and warnings. Computer vision techniques have been employed to quantify objective and subjective skills in cataract surgeries. Many scholars have explored deep learning models to automate the procedure of rating operative skills in surgeries. Action recognition in videos is a well-established approach for categorizing events in video data. Video classification models have been studied to recognize technical efficacy in surgeries which can be exploited to train novice surgeons and improve healthcare facilities.

1.3 Computer Vision in Medical Imaging

Computer Vision is the field of study that includes concepts on how computers retrieve information from digital images and videos. It can enable us to examine things with a higher focus and bigger field of view as compared to the human eye. It aims to imitate the human visual system. Furthermore, it can analyze things that are difficult for the human eye to perceive. Computer vision-based systems are being utilized for improving accuracy and at the same time the speed of medical image

analysis. Sophisticated applications might include identifying impurities such as increased darkness of the skin, cancer detection, machine assisted diagnosis and health monitoring. Microsoft InnerEye project can identify tumors using 3D imaging [9]. Computer vision already has a broad range of applications in healthcare. It has an impact in medical fields such as radiology, dermatology, cardiology, embryology and oncology.

1.4 Deep Learning in Medical Data Analysis

Deep learning models can learn to mimic human mind so that it can observe images in more accurate ways than a human can do. It can generate more precise decisions in medical imaging than humans and essentially can reduce the need for human intervention. It can enhance the quality of medical imaging and provide high-quality medical services. Deep learning systems are being incorporated in operating rooms to reduce potential risks during surgeries and increase success rates. Recent advances in deep learning algorithms for object classification and action recognition can greatly benefit medical imaging. Successful applications of deep learning in healthcare research are tumor detection, cancer detection, remote patient monitoring, machine-assisted diagnosis and medical training.

1.5 Action Recognition for Video Analysis

Action recognition refers to the task of characterizing different activities from a sequence of 2D frames(images) in a single video clip where the activity may be fully or partially performed in the entire video clip. So, action recognition or video classification is substantially a transition from image classification to the classification of a sequence of images in a video clip.

1.6 Objectives

An important part of graduate surgical training in ophthalmology is the understanding of how to proficiently perform cataract surgery. Surgical instructors should formulate a curriculum to effectively train novice ophthalmologists. The curriculum should incorporate methodical assessments of surgical skills and improve trainee surgeons'

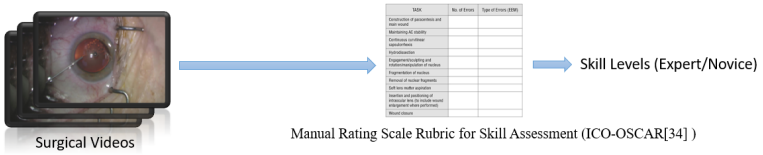

Figure 1.7: Traditional workflow for manual skill assessment in cataract surgery.

expertise to deliver adequate accountability so that patient safety is maintained. Although there has been a radical change in graduate surgical training, the effectiveness of residency training in ophthalmology is restricted to subjective and tool level analysis of videos for evaluating intraoperative surgical proficiency and competency. Over the years, tertiary-level education in ophthalmology has progressed with proficiency-based teaching.

To evaluate technical competencies for cataract surgery, various rating scales have been established. All of these scales highly depend on the subjective opinion of the observing grader. ICO-OSCAR is frequently utilized for surgical skill assessment where an assessor naively examines the performance of a surgeon in a recorded microscopic video [27]. In ICO-OSCAR, feedback is manually written down on a tabular sheet based on what is observed in the video (Figure 1.7). It also requires the faculty surgeons or graders to invest a considerable amount of time to manually validate the expertise of resident ophthalmologists. This process of investigating the surgical videos manually does not justify the reliability of the assessment. The assessment can also be biased as it might be difficult for a human to study a microscopic video. The assessment method does not have any criteria for the objective evaluation of intraoperative technical skills. Therefore, the training programs for residents are dependent on either unreliable processes or structured rating scales to evaluate surgical skills.

Additionally, a faculty surgeon has to spend a considerable amount of time to get associated with the training program for preparing the resident surgeons over time. An expert surgeon might not have enough free time in their exhaustive work schedules to review post-operative videos in person. Consequently, the effectiveness of the training program is not good enough due to the lack of effective measures to assess the performance of the residents. Thus, the evaluation of operative skill in cataract surgery is time consuming and labour-intensive.

Cataract surgery is performed on the human eye using tiny surgical tools and the surgical region is observed using a microscope. Cataract surgery, refractive surgery

and retina laser treatments could be some of the most popular ophthalmic surgery. Trainee ophthalmologists need to practice thoroughly to conduct these operations which are usually difficult. It requires a trainee doctor to master unique operation skills to precisely complete such surgical procedures. Since the surgery is performed under the microscope, it becomes difficult for a few trainee surgeons to directly observe such a long surgical procedure.

Nowadays, videos of entire surgeries are recorded to utilize them for academic as well as training purposes. Thus, cataract surgeries are now recorded in real-time and these recorded microscopic videos are used for assessing skill, teaching and training. It requires an expert to spend a lot of time going over a long surgery video to minimize chances of errors in the process of grading surgical competence. It is important to analyze the surgical videos automatically so that the recorded videos can be efficiently used for skill assessment.

We are mostly familiar with video analysis for object detection. Due to the ubiquity of video data in this digital era, video classification has emerged as an efficient approach to recognize different real-world activities. Video analysis is now predominantly being studied for identifying activities [60], generating textual description [22], generating video summaries [28], answering questions [36] and semantic segmentation. Some application domains of video analysis are robotics, surveillance, surgical workflow analysis and traffic monitoring. Various large-scale neural network models and datasets have been introduced in recent years to recognize actions in videos. Observing the success of video classification in many other fields, we decided to investigate action recognition models for designing skill assessment tools. Since skill assessment in cataract surgery requires deep analysis of surgery videos, action recognition models can be an effective tool in learning operative competence in Cataract Surgery.

Most prior video processing studies utilize publicly available large-scale datasets such as YouTube-8M and Kinetics-400 which are published along with baseline models to process those datasets. All state-of-the-art video processing models are pre-trained on preprocessed videos. The other challenge is that surgical tools used in a dataset recorded from cataract surgery are very tiny compared to the large objects in public datasets. Researchers have performed classification and segmentation on readily available videos of cataract operations. All datasets introduced in Cataract Grand Challenges [31] are recorded with very good lighting conditions and perfect focus which is different from what is observed in real surgical videos. The recorded surgical

videos available in clinics do not have such good quality as in the public datasets. Raw videos suffer from out of the focus problems of the microscope and surgical tools or the retina where the surgery is performed is not always in the microscopic view. Surgical view is sometimes occluded by surgeon's hand in the videos. It becomes difficult to work with raw cataract surgery videos using baseline models published along with the public datasets [61]. Authors in [61] evaluated the generalization performance of surgical tool detection models that were published with the publicly datasets. They have reported that the published models do not generalize well on the custom datasets collected from local hospitals. Thus, we decided to explore a customized convolutional neural network for this purpose.

An advanced convolutional neural network model is introduced in this project to automate the process of grading surgical competence in cataract surgeries. This work aims to develop a computerized grading tool for assessing the objective performance of surgeons in cataract surgeries that can save a significant amount of time by reducing human involvement while also producing highly accurate results.

Chapter 2

Related Work

Deep learning architectures have shown a remarkable advancement in learning information from image data effectively. Some of these deep learning models have been reused for recognizing tasks from video data. Action recognition is an essential aspect of video-based applications. Some of the widespread applications of video analysis are personal recommendation, autonomous driving and intelligent surveillance [53]. Although there are various deep learning approaches for learning visual representation from data, Convolutional Neural Networks and Recurrent Neural Networks are mostly exploited for performing action recognition in video data. Furthermore, some of the earlier studies have also been researched computer vision algorithms for video analysis over the years.

2.1 Classical Approach for Video Analysis

Many computer vision (CV) theories were studied before deep learning showed a large-scale triumph in the task of image-based classification. In computer vision, the visual structure of events in images or image sequences is computed using either edge-based or gradient-based descriptors and motion-boundary-based descriptors by tracking the region of interest. Feature encoding of video data is obtained by following a bag of visual word approaches or by applying hierarchical and k-means clustering. Methods such as the use of dense and sparse trajectories are also utilized for video representation. For activity detection, machine learning classifiers such as SVM [14]or Random Forest are used to train on the bag of visual words model.

Feature descriptors can represent high-dimensional visual features that can describe any specific region in a video. Histogram of Oriented Gradients (HOG) and Histogram of Optical Flow (HOF) descriptors have shown promising results in providing a compact representation and an interpretation of a video. Dalal and Triggs [17] studied the significance of the HOG descriptors for representing feature sets in a human activity dataset. HOG filters can capture edge or gradient structure which is an important characteristic of local information and provide fine-scale gradients. Feature vectors obtained from HOG are fed to Linear SVM for object classification. HOG-based Human Activity detector was tested on MIT pedestrian database and showed excellent performance [17]. Dalal et al. [18] empirically showed that motion-based descriptors such as HOF combined with HOG can improve the performance of static or moving human detection in a video or film.

Wang et al. [68] introduced dense trajectories for action recognition. Trajectories are extracted by densely sampling points in multiple spatial scales with the computation of optical flow fields. Tracking is performed along 15 frames which limits the length of trajectory to 15. Three different descriptors namely HOG, HOF and MBH (Motion Boundary Histogram) are encoded along the dense trajectories which surpassed the performance of state-of-the-art descriptors. A bag of visual words is built by using k-means clustering so that the features can be combined into a fixed size video label description where extracted video features get associated to a visual word or nearest cluster center. A separate code book or visual vocabulary was used for each descriptor (Trajectory, HOF, HOG and MBH). A non-linear SVM is trained on the bag of visual words for identifying actions in the videos.

Contrary to dense trajectories [68], visual features in a video are extracted at a

sparse set of interest points [44] for capturing spatio-temporal features from a video. Dollar et al. [21] proposed their own spatio-temporal interest point detector to find features in a local region of interest and they have evaluated the performance of their detector on a Human Activity Dataset collected by Schuldt and Caputo [14] where each activity is represented by a 4 seconds video clip.

Laptev [44] followed the idea of the Harris and Forstner point operators [32] [26] where spatial points with a significant local variation of image intensities are referred to as interest points. The recommended detection algorithm was an extension of the idea of spatial interest points into spatio-temporal domain for detecting interesting events in a video. It does not require the computation of optical flow, segmentation and feature tracking. A differential operator was defined using Normalized Laplacian Operator by Lindeberg and Bretzner [48] to assume interest points that are simultaneous maxima over spatio-temporal scale and can represent an event in video. These points were tracked to find space-time locations at a new scale. It has been shown by the author that local space-time features extracted by their suggested detector can be used to represent spatio-temporal events in video data.

Improved Dense Trajectory was the state-of-art among the other CV algorithms. iDT was proposed by Wang and Schmid [69]. In this paper, sample feature points are tracked densely for each frame in a video. SURF [11] descriptor and dense optical flow were used to match feature points between two consecutive frames. The feature matches extracted from two consecutive frames were utilised to build a homography using RANSAC. The estimated homography was then warped with the second consecutive frame. Optical flow was again calculated between the first and warped second frame. Histogram of optical flow (HOF) and Motion boundary histogram was computed using the output of warped optical flow. Many descriptors including HOG were combined along with the base trajectory descriptor to improve the performance of dense trajectory. According to the authors, HOF descriptor does not perform well until it is combined with other descriptors. Bag of feature approach and Fisher encoding was exploited for encoding features. The video classification was performed using SVM. iDT was tested on 4 datasets such as HMDB51 [43] and Hollywood2 [50]. The authors showed that the performance of Dense Trajectories can be effectively improved with combined descriptors.

2.2 Deep Learning Algorithms for Video Analysis

Numerous algorithms have been studied by researchers for recognizing activities in videos. 2D convolutional neural network served as a crucial tool for designing action recognition models over the years. The design strategy has been evolved from simple 2D Convolutional networks to extensive networks with the integration of numerous fusion techniques. The fusion strategy did not confine to the blending of Convolutional neural network layers only. Authors have also investigated many large network infrastructures by combining either 2D Convolutional nets with 3D Convolutional Nets or 2D convolutional nets with Recurrent Neural Network modules.

Karpathy et al. [39] investigated multiple ways to fuse temporal features extracted from consecutive frames by using 2D convolutions in single-stream network. This paper shows how to modify an existing convolutional neural network to make it capable of processing temporal dependencies in video [39]. Four models were used for fusion and they are Single frame, Late Fusion, Early fusion and Slow Fusion. For single frame fusion technique, features from all frames are fused together at the end in a single network architecture. Two nets with shared parameters were used in Late fusion where only the first and last frame from the video clip was fused. A contiguous segment of 10 frames was used as a representation of the video in the early fusion model. In the Slow Fusion model, contiguous frames were fused consecutively in the convolutional layers. Two separate inputs were fed to two convolutional networks in the proposed multi-stream neural network. The authors discovered that the outcome of all these extensive computations did not improve the performance significantly as compared to iDT and other hand-crafted algorithms. Additionally, earning features from a diverse dataset like UCF101 was not possible for the single stream network.

Simonyan and Zisserman [60] processed spatial and temporal information separately with a two-stream network. The network has two separate networks to model both spatial and temporal information independently. Single frames from a video was passed to the spatial stream convolutional network whereas the temporal stream network takes stacked optical flow across multiple frames as input. SVM was utilized to combine the prediction from two individual networks. However, this network improved the performance of single stream network slightly on UCF101 dataset which was also close to the results of some other work that have used Improved Dense Trajectory(iDT) [69]. It still has some major drawbacks. It needs a separate computation

of optical flow vectors which needs to be stored. Along with all this extensive computation of different fusion techniques and separate training of individual networks, it still does not improve the performance notably.

Donahue et al. [22] built on the idea of single stream convolutional network by incorporating a recurrent neural network like LSTM (Long Short Term Memory) on the feature maps from simple convolutional layers. They also observed the impact of RGB and Optical flow fields as a separate input to the model and have concluded that a combination of both works well. Linear Convolution Blocks were followed by blocks of stacked LSTM in this model while keeping it end-to-end trainable for action recognition. This network with the use of LSTM blocks improved performance over single frame model like Spatial Stream Convnet.

Du tran et al. [65] introduced 3D convolution for extracting features from videos. 3D convolution on the video volume was applied instead of using 2D convolution across each frame from a video. They have discovered that 3D convolution with a 3*3*3 kernel is highly effective in extracting spatiotemporal features from video. They also performed deconvolution which can learn spatial appearance in the first few frames followed by salient motion in the later frames of a clip. It is a shallow network with five convolutional layers followed by two fully connected layers and five pooling layers in between convolution layers. 2 seconds video clips were extracted from each video in the UCF101 dataset and 10frames representing each clip was used as input. C3D combined with iDT and Support Vector Machine worked well compared to other popular architecture like two-stream network [60] and iDT [69]. However, it still has a large number of parameters. It is obvious for any framework with 3D convolutional layers. It took the researchers about two months to train the C3D on the Sports-1m dataset which has almost 34.8 million parameters. Due to the use of 3D convolutions and deconvolutions, C3D is computationally expensive in terms of memory requirements.

Feichtenhofer et al. extended the two stream network [60] with some noteworthy improvements [24]. Like the two stream network, it consists of a spatial net as well as a temporal net. Spatial appearance is captured from any specific pixel location in the data by Spatial Net whereas temporal net extracts motion information related to that region. For each specific region in data, the feature map related to spatial information is combined with its corresponding temporal feature map in the earlier level of the Two Stream Fusion Network. This is done by fusing both spatial and temporal net

at an earlier level so that temporal dependencies can be mapped along with its space-related dependencies for any specific region in the scene. This network was able to beat C3D and iDT with its capability of modeling long-term temporal dependencies along with static information in the video. This network has the same weakness as base two steam net as it needs two sub-networks to be trained independently.

Zhu et al. [73] attempted to improve two-stream networks by including a Hidden Stream subnet that can compute optical flow information and learn long-range relations before the input is forwarded to Temporal Net. This work tries to resolve the issue of the need for precomputed optical flow by generating optical flow input in the hidden stream subnet. The extra subnet is stacked on top of the base two-stream network. The authors discovered that this network named MotionNet has a similar accuracy rate as the base Two Stream Net but it is approximately 10x faster. It is also real-time applicable as it does not require precomputed optical flow to be stored independently.

Zisserman et. al explored different approaches to extend the base C3D network by incorporating it into two stream network [15]. In this approach, two stream inflated 3D convnet was proposed. Two different 3D CNN nets for both spatial and temporal stream in Two Stream Net were exploited. 2D convolution filters and pooling layers were replaced by 3D filters and pool layers. The Kinetic dataset was introduced by the authors which were also recognized as a benchmark dataset for action recognition. Two stream I3D with the association of 3D convnets trained on sequential RGB frames and optical flow frames was capable of improving the performance of action recognition over other existing models. It still suffers from the problem of not being end-to-end trainable.

Diba et al. aggregated temporal relationships in data across variable depth in a Temporal 3D convolutional network called T3D [20]. It was built on the idea of using two Stream Inflated 3D where two different networks were used. One is a Linear 2D convolutional net while the other part is a 3D Convolutional Net. The author suggests a liner 3D convolutional net which is inspired by the structure of DenseNet [34]. In the 3D DenseNet model, the temporal pooling layer with different depths is stacked on top of each Dense block for extracting temporal data at different levels. This pooling layer with different kernels is called Temporal Transition Layer which takes the output from every Dense Convolutional block. This network can also be referred to as a transition of 2D DenseNet into 3D DenseNet with the Temporal Transition

Layer in between the convolution layer.

Bhardwaj et al. [12] came up with the idea of a memory-efficient HRNN model where the model uses fewer frames such as every jth frame from a single video in order to reduce redundant frames. One RNN is used to process all frames from the input video which then trains one more RNN to process every jth frame only for video classification. The authors used a bag of frames dropping intermediate frames from each input video of YouTube-8M dataset. The idea of dropping frames randomly might result in the loss of information in the context of sensitive data such as surgical or surveillance video. RNN is always a good choice for sequence learning but it does not learn spatial features as much as CNN's.

State-of-the-art video processing models like C3D [65] and I3D [15] have complex architecture. These models require excessive and high computational resources. For example, I3D is not real-time applicable or end-to-end trainable because it requires optical flow calculations over the raw data. So, there is still a need for a memory-efficient video processing model which the referred paper [12] tries to address.

2.3 Algorithms for Surgical Tool and Phase Analysis

The trend of surgical video analysis in the operating room started with the recognition of surgical instruments and phases at the beginning. Automated tool and phase detection in surgical images or videos can essentially facilitate the process of surgical workflow analysis as well as technical skill evaluation in the operating room. Researchers explored various hand-crafted feature extraction algorithms and machine learning models for studying surgical workflow. Some of these earlier works were mostly focused on tool identification. Instrument detection works as a fundamental step in gathering information about tool trajectories and usage of individual instruments in an entire surgical video. The study of the surgical instrument and phase recognition is then utilized to compute descriptive metrics for assessing operative skill in surgical procedures. Objective assessment in surgeries heavily relied on these descriptive metrics.

Raju et al. [5] ensembled VGGNet [38] and GoogleNet [63] to perform surgical tool detection on laparoscopic videos. The ensembled models were trained using frames extracted from the videos. The two networks were ensembled using normal

averaging, weighted averaging and geometric averaging to obtain the final result. They have validated their work on M2CAI 2016 Challenge dataset [52] which has videos of the Cholecystectomy surgery captured using a laparoscope and were the winners of the challenge. They have applied different data augmentation techniques such as rotation, horizontal and vertical flipping. The work concluded that normal averaging yielded better results than the other ensemble techniques.

Sahu et al. [49] developed a CNN architecture that has the same structure as AlexNet for tool detection. The CNN model was trained using transfer learning features from ImageNet to generate contextual features based on the appearance of the objects in the laparoscopic images of m2cai16 dataset. To perform tool detection, the features are then fed to a Random Forest classifier for training on the images of surgical instruments. The authors performed random flipping and cropping to artificially augment the data. For phase recognition, features from ten consecutive time points are calculated. A random forest classifier is trained to predict based on these features. The prediction probability is again combined to time series to make the prediction in a more localized manner.

Twinanda et al. [67] presented EndoNet which is modeled by fine tuning AlexNet network and using weights of ImageNet dataset [57] to perform multiple tasks on the images of laparoscopic videos from m2cai2016 dataset. EndoNet was designed to jointly recognize phases and detect tools in the images. This was done by making the 3rd fully connected layer output probability scores for each of seven nodes where each node refers to the confidence that an image contains any specific tool. The output from first two fully connected layers is concatenated to another dense layer namely fc_phase to generate features for phase recognition. Feature vectors learned from the last dense layer are passed to a Support Vector Machine (SVM) which yields a probability score for the detected phase. The probability score from SVM is again passed to a 2-level hierarchical HMM. The HHMM uses the confidence score obtained from SVM as the given observations. The final prediction for the phase is made using the forward algorithm in the HHMM. They have extended their work using the Cholec80 dataset by experimenting with different architectures based on AlexNet base network configuration [66].

Zhang et al. [71] constructed a modulated anchoring network using Faster RCNN

for detecting surgical instruments in laparoscopic videos. Modulated anchoring network consists of 3 parts and they are anchor box location prediction, its shape prediction and modulated feature module. Anchor location branch is designed to generate a probability map and the map tells the region where the center of any object or instrument may exist. The anchor shape prediction branch tries to identify the shape of the instrument at the detected anchor box location. Modulated feature module incorporates the shape information of the instrument or object produced by the shape prediction branch into a feature map to further compare the features with the shape information of available instruments in the surgery. A relation module is embedded in the network where it attempts to compute the relative association of different instruments in any scenario of the videos. The base feature detection network is formed using ResNet-101 with Feature Pyramid Network [46] [47]. The modulated anchoring network outputs bonding box labelling for each surgical tool in a video frame. The authors further studied the movement of detected instruments by generating heat maps for each surgical tool. Operative efficiency in the surgical videos is evaluated by examining the tool usage patterns through the heat maps and the timeline of how long each tool is used in an entire video. The author showed that Faster RCNN based network achieved a higher tool detection accuracy than other existing approaches.

Bodenstedt et al. [13] investigated the performance of an active deep learning system based on Deep Bayesian Network (DBN) for annotating surgical images and videos recorded in real-time surgery. AlexNet is selected as a standard CNN to design the proposed DBN for active learning. It has been pre-trained on ImageNet before being modified by the authors to transform it into a DBN. AlexNet consists of five 2-dimensional convolutional blocks and three fully connected layers where each convolution layer is followed by a pooling layer. A dropout layer is added after each convolution layer in the base AlexNet to extend it to a DBN for instrument detection in laparoscopic images. In the DBN, each dropout layer is followed by a dropout layer. The simple AlexNet with just an addition of a dropout layer gets exploited for image-based surgical tool recognition.

To perform video-based phase recognition, a complex form of recurrent neural network like Long Short Term Memory (LSTM) is added at the end of the first fully connected layer in the simple DBN which is described above [13]. The new Recurrent DBN allows the system to look at any surgical video from the perspective of consecutive frames representing the relationship between each frame in the entire

video. The LSTM layer can learn information from the previous frame and aggregate the learned information from the previous frame to its next consecutive frame to make video-based surgical phase segmentation possible. Surgical videos are usually long and processing an entire video can be time-consuming. To eliminate this issue, each video is segmented into smaller video clips where each clip has a short duration. Video frames from the Cholec80 dataset were processed with a resolution of 384×216 pixels. DBN was trained using Adam [42] as an optimizer. They have evaluated their system on Cholec80 dataset from the m2cai2016 challenge and showed that the DBN could be effective for image frame or video frame annotation.

Prellberg and Kramer [54] built a model using a 50-layer deep residual network [33] to distinguish different surgical tools in the videos of cataract surgery. Their system is built on top of ResNet50 which is a variant of the deep residual network model. It consists of 48 convolutional layers along with one max pool and one global average pool layer. ResNet50 is finetuned by freezing the first few layers randomly to reduce memory requirements instead of training the entire network. The authors investigated the impact of both Global average pooling and max pooling by pooling out image features from the last convolutional layer of base ResNet50 and passing the pooled over features to the last fully connected layer for instrument classification. The authors also experimented ResNet50 as a fixed feature extractor in a different setting by using the first random number of layers with fixed weights. Similarly, feature vectors obtained from the feature extractor are then fed to either the average pooling or max pooling layer to identify surgical tools. Every sixth image frame from each surgery video is used to train ResNet in order to reduce extra redundant frames from data as well as the training time of the model. Video images were augmented by flipping them horizontally and rotating them with a random rotation ranging from -15° and 15°. It was concluded by the authors that fine-tuned ResNet showed a better tool classification performance on the Cataract Grand Challenge dataset [7] as compared to the setting where ResNet50 was used as a fixed feature extractor.

Multi-image fusion strategy is applied by Hajj et al. [8] to improve tool presence detection in cataract videos which functions as an initial step in performing surgical workflow analysis. A sequence of consecutive frames representing each video is deployed in this work instead of an individual video frame. A simple convolutional neural network (CNN) is designed which consists of a few convolutional layers. Each convolutional layer is preceded by a pooling layer. The last two fully connected layers

which are intended to predict tool presence are again followed by the dropout layer. The CNN takes in a sequence of 16 consecutive video frames as input. The optical flow between each consecutive image in a sequence is computed and the output from optical flow computation is processed by the first few layers in the CNN. For fusing information from one image to another image in a sequence, the activation map for one image is fused to the activation maps of the next consecutive image. The resultant activation map from two previous images is then again fused with the activation map of the latter image in the sequence. The CNN is first trained on each independent video frame and then retrained again on image sequences to further inspect the changes in the model performance for different input data. The authors showed that the use of image sequence is better for improving the tool detection performance of the CNN as compared to the conventional approach of making a CNN process each individual video frame.

Hajj et al. [31] recapitulated various instrument detection models with high accuracy for 21 different types of instruments using their dataset released for the CATARACTS challenges. The authors also summarized all research works from the submissions of the challenge on cataract dataset. Different machine learning solutions were presented in Cataract Grand Challenge to automatically annotate surgical instruments in Cataract videos. All participating teams have exploited different variants of pretrained 2D neural networks such as VGGNet [38], ResNet [33], DenseNet [35] and Inception-v3 [64]. All the proposed Convolutional neural network models were trained on the video frames of cataract Surgery videos where several teams were recognised based on the quantifiable performance of their proposed tool detectors. These challenges also paved a way for the researchers to navigate advanced deep learning systems for improving healthcare facilities in the operating rooms.

2.4 Approaches for Surgical Skill Evaluation using Tool Analysis

Most of the publications in the area of skill assessment using surgical videos were dedicated to surgical workflow analysis using automated segmentation of instruments and phases in surgical videos. The paradigm in skill assessment has been shifted from frame-level tool motion analysis to video-based assessment over the years.

Jin et al. [37] analyzed tool usage information in the videos of laparoscopic surgeries to assess the performance of surgeons based on instrumental tool tracking. The researchers introduced an approach for detecting tool presence and tool movement in surgical images by leveraging region-based Convolutional Neural Network (RCNN) [55]. Video frames are fed to a VGG16 network as a base network for learning meaningful visual features from data. It was pre-trained on ImageNet features and further fine-tuned using the m2cai16-tool-location dataset. A Region Proposal Network (RPN) uses convolutional features from the base VGG16 for detecting spatial bounds of any tool in an image by generating a bounding box around the detected tool. One node of the network is designed to predict the objectness score where this score refers to the likelihood of whether a tool is present in the bounding box or not. Features from the region of interest that is specified by the bounding box are pooled over to perform tool presence detection in the video frames. The proposed RCNN outputs a bounding box for each detected tool in frames. It took the RCNN network 2 days to get trained on NVIDIA Tesla K49 GPU.

Timelines for each detected tool were produced where each timeline specifies the duration of how long a tool was used in a surgery video [37]. Heat maps were generated for each bounding box occurrence. Trajectory maps were created for tool usage in a video along with a total distance of how long each tool traveled in a frame. Surgical competency was evaluated based on the tool usage heatmap, timelines, tool movement patterns and duration. A new dataset was introduced by the authors for detecting tools in surgical images of Cholecystectomy surgery which has binary annotations of tool presence and the performance of the approach was validated on the new dataset.

Funke et al. showed that 3D convolutional network can be a better resource for spatiotemporal feature learning which can facilitate skill assessment tasks in videos related to basic surgical tasks [25]. Each simulated video is subdivided into a shorter segment where each segment contains a sequence of consecutive video frames. Each video segment has a length of 10 seconds which are extracted from an original video of 5 minutes. The authors hypothesized that the use of short snippets instead of using the entire 5 minutes video could reduce the size and complexity of the problem. I3D is first pre-trained on kinetics dataset [40] instead of training it from scratch. To investigate the performance of different input modality, optical flow field is computed between successive video frames as the optical flow can capture motion between two frames. Before training the 3D CNN, each video segment is represented either with a

stack of consecutive RGB frames or a successive frame sequence of optical flow fields. The suggested framework is converted to a Temporal Segment Network following the Siamese Architecture where a few 3D Convnet instances share the same weights and each 3D Convnet instance classifies one video snippet at a time. The TSN developed using the I3D is trained using either a stack of RGB frames or a stack of optical flow fields representing an individual video snippet. The work is tested on the publicly available JIGSAWS dataset [6] which contains simulated videos of basic surgical suturing and knot tying. The temporal segment network demonstrated a higher level of skill classification accuracy on the JIGSAWS dataset.

Glarner et al. [23] quantified technical skill in the operating room by studying spatiotemporal characteristics of hand movement in the videos of a plastic surgery procedure called mammoplasty. Instead of using Kinematics which needs to be collected from an electromagnetic sensor, the original surgery video was segmented into smaller clips representing different tasks in surgery with the help of task-specific video analysis software. Video segments were distinguished based on surgical tasks such as cutting with a scalpel, suturing and instrument tying. Kinematic properties of the surgeon's hand movement were analyzed using ROI software. The kinematics property includes displacement, velocity and acceleration. ROI (Region of Interest) motion trajectory between consecutive video frames was tracked using a template matching algorithm. Descriptive matrices were computed by statistically analyzing the changes in the velocity, displacement and acceleration of the surgeon's hand in consecutive video frames. An objective assessment of surgical skills was performed using these descriptive matrices.

Sharma et al. [59] presented a new framework to automatically evaluate surgical skills in videos following the criteria of Objective Structured Assessment of Technical Skills (OSTATS). Motion dynamics in the video frames are encoded using frame kernel matrices where spatio-temporal interest points are computed along with HOG-HOF (Histogram of oriented gradients of optical flow) descriptors. K-means is applied to the extracted features with 5 clusters where the clusters are moving entities in the videos. The moving entities are surgeons' hands and three other instruments which are visible in the videos of basic surgical tasks. Frame kernel matrices are computed between two frames using a kernel function which is expected to identify similarity between two frames. Features related to motion texture are obtained from the frame kernel matrices for encoding motion dynamics in surgery videos by applying Gray

Level Co-occurrence Matrix (GLCM) and Local Binary Pattern (LBP) [30]. A linear regression model is trained on the motion texture features obtained from the GLCM and LBP approach. To make the Linear regression model predict scores for skill assessment, reduced dimensionality feature space is derived using Linear Discriminant Analysis where three skill levels with a score range are grouped. The three skill levels are based on the criteria of lower, intermediate and higher level OSATS scores. The regression model is trained to predict only two skill levels for the two dimensions in the reduced feature space where a lower score indicates to the novice surgeon and a higher score signifies the expert level skill in the surgery.

A new system is introduced by Zia et al. [76] to evaluate the expertise of surgeons by computing frequency coefficients in time series for repetitive tasks such as suturing or knot tying that are commonly observed in basic surgical activities. To get motion information from video clips, Spatiotemporal interest points (STIPs) are extracted using Harris3D detector. Likewise, Sharma et al [59], HOGHOF (Histogram of oriented gradients of optical flow) descriptors are computed. Motion clusters are learned from the extracted motion features by applying k-means clusters where each motion class represents different moving parts such as arms and surgical tools in the video. To transform motion data into time series, STIPs belonging to each video are assigned to one of the motion classes where minimum Mahalanobis distance is used to learn motion clusters from data. Frequency coefficients are obtained from the time series data by applying Discrete Cosine Transformation (DCT) and Discrete Fourier Transformation (DFT). The frequency coefficients representing different motion data in the surgical tasks are employed to assess skill in the videos. The framework was tested on different video clips of trainee residents performing basic surgical tasks in the laboratory. The authors demonstrated that frequency coefficients processed from sequential time data can improve skill classification accuracy over other state-of-the-art frameworks.

Kinematic data computed from the hand movements of surgeons in the videos of basic surgical tasks is exploited by Azari et al. [10] to automatically generate computer ratings for estimating surgical efficiency. Three expert surgeons rated the performance of suturing and knot tying in the video clips of basic surgical tasks on a scale of 0 to 10 using subjective measures. The subjective measures involved three motion scales. The motion scales are the fluidity of motion, motion economy and tissue handling in suturing and knot tying tasks. The r region of interest (ROI) was

traced on a noticeable segment of the surgeon's hand using video tracking software within consecutive video frames. The location of ROI for tracing the surgeon's hand gesture was selected in each video clip. The tracked record of ROI location involved the displacement, speed and acceleration of the surgeon's hand movement. Different regression models were investigated to predict ratings for each of the three motion scales that are carefully chosen following the criteria of OSATS (Objective Structured Assessment of Technical Skills). The authors revealed that kinematics measures acquired by tracking the hand movement of surgeons can be an effective resource to design a computerized system for rating surgical skills in videos and hand movement can be directly traced in recorded videos without using any marker or sensor.

2.5 Models for Skill Assessment using Surgical Video

There are numerous processes for feature extraction that have been used to categorize the skill level of surgeons using video data.

Zia et al. [78] explored three different processes to extract features from videos and compared the classification accuracy for each type of feature. Initially, spatiotemporal motion information (STIPs) are captured from video data using HOG (Histogram of oriented gradients) and HOF (Histogram of optical flow) descriptors as described in [76]. Motion information learned from the videos are then encoded to generate time series representation for each video clip following [76]. Three different feature categories are studied using the time series representations and the categories include symbolic features used in hidden Markov models (HMMs) and Bag of word approach, visual texture-based features derived using frame kernel matrices and frequency-based features computed using DCT-DFT.

For symbolic feature modeling, time-series data identifying each video was converted into a set of discrete symbols using k-means clustering [78]. HMM was implemented with Gaussian Mixture Models (GMMs) which are intended to represent feature space. The bag of words model also uses symbolic features. The visual codebook is derived using the spatiotemporal features (STIPs). The motion classes are obtained using the feature vectors derived from the vocabulary referring to the codebook. Feature vectors are then mapped to the words on vocabulary where each video gets represented by a histogram of words. A k-nearest neighbour classifier

is trained using the feature representation to categorize the videos into a different level of surgical skill. Motion dynamics in videos are encoded in the form of Motion texture in a frame kernel matrix as demonstrated in [59]. In the process of sequential motion texture analysis, time series representation for video data is divided into equally spaced temporal windows where each window identifies the biggest motion class present in that surgical video. In frequency-based feature modeling, Discrete Cosine Transformation and Discrete Fourier Transformation are applied to convert time series representation into frequency domain as explained in [76]. The authors argued that the performance of skill evaluation improves as the experiment progresses from symbolic features to fine-grained frequency-based feature modeling. The best accuracy for skill classification in traditional surgical activities is achieved with frequency-based features.

Surgical skill assessment is performed by Zia et al. on the video and accelerometer data recorded from the simulation of basic surgical events such as suturing and knot tying using Entropy-based features [77]. To learn motion information, STIPs (Saptio temporal interest points) are extracted from video data following their earlier work in [76] which are then clustered using k-means to find motion classes in video clips. Each video is then encoded as a multidimensional time series using the motion classes learned by the k-means model. The asynchrony between two successive time series can be measured using cross approximate entropy (XApEn). Cross approximate entropy is calculated between each two time series in the data to capture repetitiveness of motion in the task of suturing and knot tying. A separate nearest neighbor classifier is trained for skill assessment using Entropy-based features computed from the time series representation of both video and accelerometer data individually. A higher accuracy for skill classification was observed using video data. The researchers anticipated that videos could act as a better resource for retrieving information related to skill evaluation as compared to accelerometer data. The classification accuracy with entropy-based features outshines other existing modalities with various feature extraction types such as frequency-based features and sequential motion texture.

Zia et al [74] developed RP-Net-V2 to identify different steps of robotic-assisted radical prostatectomies and performed operative skill assessment using the phase detection result from the RPNet-V2. RP-Net-V2 is designed using VGG19 with ImageNet pre-trained weights where VGG19 acts as a base feature extractor for the images. A stream of consecutive images is fed to separate convolutional neural networks

where each input image is processed by an individual VGG19-based CNN which learns visual features from a single input image. Each of the VGG19-based CNN shares the same weights. Feature vectors from each CNN are concatenated together by stacking their output on top of each other. The concatenation is done before the final feature vector is passed to another single LSTM network which outputs the final prediction for task recognition in surgical videos.

Zia et al [74] also employed kinematics data and event-based system data which are collected from the robotic system for computing task-wise performance reports. Kinematic data includes economy of motion in surgery and speed whereas event-based data comprises of camera control, hand controller clutch on/off and arm swap. A separate LSTM extracts features from the system data which then gets combined with the image-based features. The combined features from both image data and system data (Kinematics and Events) are utilized further to generate a performance report for each surgical task and to assess surgical efficiency in the operating room. The performance of the work is evaluated on the Prostatectomy dataset introduced by [75] by comparing the model's annotation for each task with the ground truth labels.

A system is presented in [72] for automatically grading surgeons' efficiency in simulated cataract surgeries by combining performance metrics of spatiality score and motion score to derive assessment results for skill evaluation. Only the Capsulorrhexis procedure is considered for simulating the activities in cataract surgery as it is one of the most important basic steps. Skill assessment is evaluated based on three matrices namely spatiality, duration and motion which are also included in the curriculum of the expert grading.

In the proposed framework [72], the pupil is detected as the region of Interest (ROI) in a sequence of images representing each video sample. For procedure segmentation, ROI location across successive video frames is tracked to categorize frames into action frames, downtime frames and key-frames. A frame is categorized as an action frame if any surgical activity is performed. A key-frame identifies the last frame that indicates the completion of all surgical activities in an independent Capsulorrhexis procedure. The starting point of an individual Capsulorrhexis is identified and the frame prior to a set of frames where changes in the ROI (region of interest) position are observed is considered as the key-frame or the last frame indicating the completion of an individual procedure. All other frames except the key-frame and

action frame are identified as downtime-frame. The completion time of a procedure is measured using the action frames and key-frames to record the duration score for each video. Unnecessary light reflection in any video frame is removed before proceeding to image segmentation. K-means clustering is applied on the RGB color space of each key-frame to identify three cluster regions representing the membrane regions, the peeled regions and the black color regions. The membrane region refers to the lens membrane that needs to be peeled and the peeled region is where the peeling of cataracts is performed. The black color region indicates the tools used by the surgeon. The three regions refer to the cluster centroid. A linear regression model is then trained using spatial features extracted from segmentation where the model produces a spatiality score for each video following 3 factors. The spatiality score is predicted using the segmentation information on the key-frame only leaving all other frames in an image sequence unused.

To further encode motion data, the motion regions that contain surgical tools only are considered and optical flow fields for the motion regions in all frames in a video are computed to find the motion changes of the tool [72]. A support vector machine is trained using the optical flow values to make the model predict motion score for the assessment of surgical expertise. The efficiency metrices predicted by the model were compared to expert grading and it was found that the model was able to match human-level ratings for skill assessment to a certain degree.

Kim et al. performed an objective assessment of technical skills using the videos of cataract surgeries [41]. The authors of the paper followed the idea of using information about instrument usage in surgical videos and performing skill assessments based on the retrieved information. Time series representations of the video data were computed to apply 1-dimensional temporal convolution and extract motion information from the videos. Optical flow fields are computed for a set of video frames representing each video. As one form of data representation, the optical flow vectors are computed instead of the rgb frames. For encoding local features, tool trajectories representing a set of frames from the videos were obtained using crowd annotation from the Amazon Mechanical Turk framework. To apply 1-dimensional convolution, trajectories were transformed into time series of tool tip velocities. The video data can be represented using the time series representation of tool tip velocities. Both the data representation is again flattened to 1-dimensional vector instead of 2D vector so that the data can be convolved using 1D convolutional layers.

In this work, a temporal convolutional network (TCN) is introduced following the base structure of a Temporal Convnet [41]. The TCN consists of few 1-dimensional convolutional layers. Each convolution layer is followed by a max pool layer. The third convolutional and pool layer is followed by a global average pooling layer before the features are forwarded to the fully connected layer. The neural network is trained using different combinations of data representation that include the optical flow fields (FF), tool tip velocity (TV) and tool tip positions. The TCN is also trained on each data representation separately. The neural network model is evaluated on their custom dataset. It consists of 99 videos of the capsulorhexis phase which were recorded through a microscope. The data was annotated by one surgeon who followed the ICO-OSCAR rubric. This work performs binary skill classification for assessing technical skills on Capsulorhexis videos. The authors reported an accuracy of 84% and an AUC of 86% for the data representation of tool tip velocities. The outcome on tool tip velocities was higher than the results for other data representations such as flow fields or a combination of tool tip positions, tool tip velocities and optical flow fields.

2.6 Summary

Based on the review of the literature, it is observed that some authors explored different computer vision approaches such as HOG [78], HOF [78], DCT [76] and DFT [76] to extract features from videos to perform skill assessment. Traditional machine learning models such as k-means and k-nearest neighbour were also utilized for skill categorization [77] [78]. However, the skill classification accuracy were improved when convolutional neural network was utilized [74]. Many authors studied 2D or 3D convolutional neural network architectures or pre-trained convolutional neural networks for surgical tool analysis.(e.g. ResNet-50, Inception v3 and NASNet Mobile). Recent studies demonstrated noteworthy performance using tool movement information [41]. Trajectories of surgical instruments are required to analyze tool motion. However, additional equipment for tool tracking or availability of robotic surgery systems such as da Vinci Surgical System is required to obtain trajectories of tool usage. Furthermore, tool recognition by extracting frames from the videos and then using those extracted frames for phase recognition and skill classification is an exhaustive process. Therefore, this project intends to develop a deep learning model for skill evaluation

using raw surgery videos that is cost-effective, end-to-end trainable and can assit human raters with a reduced amount of human interaction.

Chapter 3

Background

3.1 Convolutional Neural Network

The convolutional neural network is also known as Convnet or CNN was first introduced by Yann LeCunn in 1998 [45]. It is one form of artificial neural network that operates on a grid-like topology for processing data [29] [51]. Each neuron in a CNN explores data by responding to its specific receptive field only so that it can mimic the human brain. CNN is designed following the connectivity patterns in the human brain.

Multiple layers of artificial neurons are placed together in a convolutional neural network following the neuron pattern in the human biological system. Mathematical functions are used as artificial neurons that are expected to calculate the weighted sum of multiple inputs and yield the activation value as output.

Computers usually take an image as an array of pixel values depending on the resolution of the image. When an image is fed to a CNN, it detects patterns and features related to the visual appearance of each unique entity or object in the image. A CNN is capable of effectively capturing the subjective context of objects in an image by applying relevant filters. It consists of 3 specific layers in general (Figure 3.1). The first layer is the convolution layer which is followed by the activation function and pooling layer. The last layer is called the fully connected layer which also acts like the classification module.

Zeiler and Fergus [70] precisely demonstrated the process of feature extraction at different levels of a CNN in detail. The shallow layers of CNN look at more detailed features and the deeper layers go for more generalized features related to

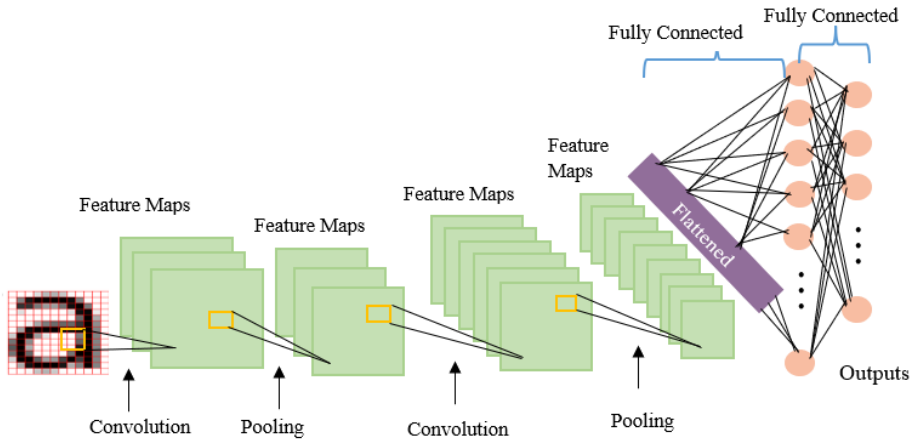

Figure 3.1: Basic structure of a CNN

objects. The first few layers usually function by detecting detailed features such as horizontal, diagonal, vertical edges, curves and gradient orientation in an image. The output of the first layers acts as the input to the second layer where more complex features related to the combination of edges and corners are retrieved from the input data. The features obtained by the first shallow layers do not contain meaningful information about any specific entity or object. The convolution layer, activation layer and pooling layer get repeated several times to make the network grow deeper and systematically capable of learning deeper context in data.

3.1.1 Convolution Layer

It is the first layer in a CNN that is responsible for learning visual context in the input data [29] [51]. It is the core building block of CNN. Convolution is the dot product of the image matrix and a kernel for filtering relevant information.

Convolution with different kernels results in different feature extraction (Figure 3.2). A kernel is a matrix of weights where the values are defined to filter out any specific visual information. The kernel for filtering out edges is different than the one for sharpening an image or identifying any object. It slides over the image matrix from left to right until it finishes the width. Once it reaches the width again, it hops down and starts sliding from left to right. It traverses the entire image matrix by sliding over the image following the network stride. A stride is the number of pixels shifts that the kernel needs to maintain while sliding over the input image from left to

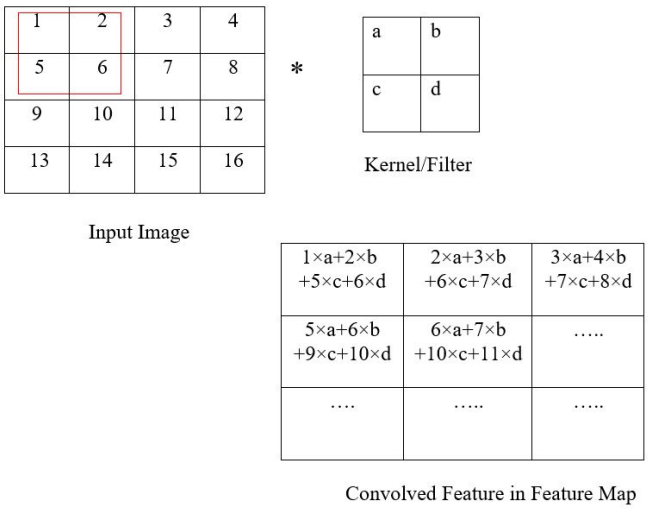

Figure 3.2: Convolution Operation

right. It specifies how much the kernel or filter should move towards right direction or downward position while applying convolution.

The dimension of the convolved feature map gets reduced in figure 3.2 whereas the dimension of the input image was bigger. It could be reduced, increased and remain the same as the dimension of the original input image depending on the padding style. Valid padding is one type of padding that can decrease the dimension of the output feature map as compared to the input image. Same padding is another type of padding that preserves the dimensionality of input data in the output feature map.

3.1.2 Activation function

Activation function is a non-linear transformation that helps the neuron to decide whether information should be fired to the next layer of neurons as the input or not. Sigmoid, ReLU and Leaky-ReLU are some of the activation functions that are applied in CNN.

3.1.3 Pooling Layer

The number of parameters in the network increases when the input data is large and this keeps growing as the network goes deeper. The pooling layer is designed to reduce the spatial dimension of the feature map. It helps the network process data

with a reduced amount of computational power. It abstracts dominant features that are invariant from the perspective of positional and rotational views. Pooling is also referred to as subsampling or down-sampling as it decreases the dimensionality of the feature map while pooling over substantial information. Max pooling and average pooling are two types of pooling operations that are frequently applied. Max pooling takes the largest value from the section of the input where the kernel is applied. Average pooling returns the average of all pixel values from the region of the image matrix where the kernel is located.

3.1.4 Fully Connected Layer

The fully connected layer (FC layer) is also referred to as the classification layer. It is a useful approach for discovering non-linear combinations of high-level features from the output feature map of the convolution layer. Feature vectors obtained from the convolutional layers are converted to a 1-dimensional column vector so that it can be forwarded to the FC layer. The FC layer consists of a fully connected neural network with one or more layers for performing classification. The flattened column vector is redirected to a feed forward neural network and backpropagation is applied to every training iteration of the model.

3.1.5 2D Convolution across input channels

In the 2D Convolutional neural network, convolution is the elementwise multiplication of the pixel values in the image matrix with the weight values in the kernel or filter matrix.

If the convolution operation is performed elementwise, the output feature map is always 2-dimensional where it has width and height only (Figure 3.3). The size of the output map for applying elementwise 2D convolution with the same padding on an input image of size 32×32×3 with a filter of size 5×5×3 will be 32×32×1. This is because the 3D filter moves across the height and width of the input image only. When 2D convolution is applied across RGB channels, the 3-dimensional filter matrix moves along the height and width of each color channel. In the case of 3D filter performing 2D Convolution, each of the 3-kernel matrices in the 3D filter is applied to each channel in the input image. In elementwise convolution, the outputs of convolution across each channel are summed up together which forms a 3×3 output

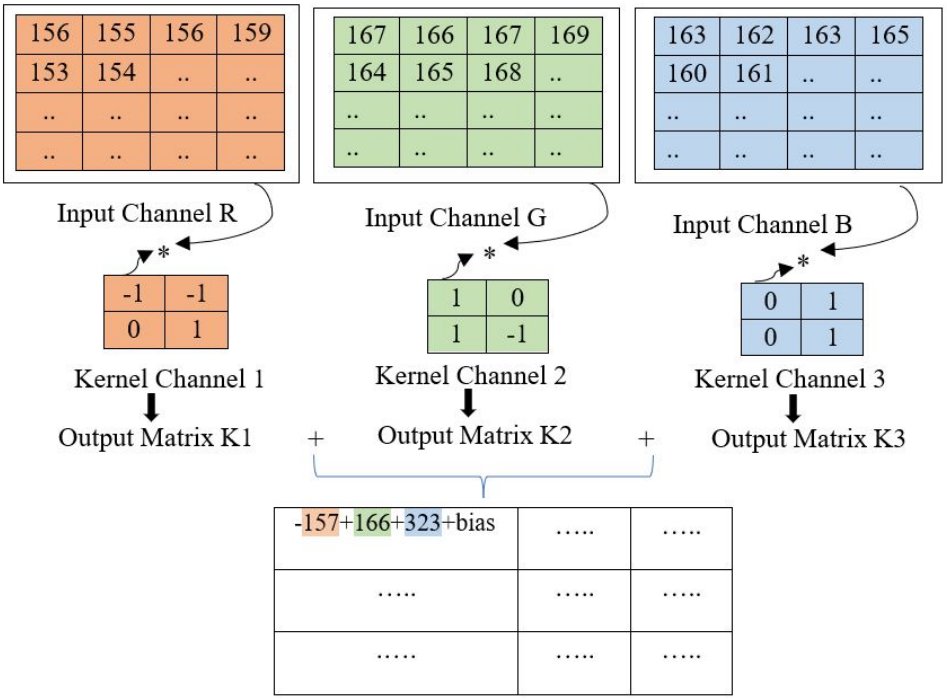

Figure 3.3: 2D convolution operation across channels

feature map with 1 channel. 2D convolution applied on a single input channel will result in a 2-dimensional output feature map if the convolution is not conducted depth-wise.

3.1.6 3D Convolution Across Depth

3D convolution is a comprehensive version of 2D convolution where convolution is applied across multiple dimensions instead of only height and width (Figure 3.4). In 3D convolution, a 3D filter with height, width and depth slides in all 3 directions (height, width and depth/channel). The elementwise matrix multiplication along with addition generates one value at each region where 3D filter is placed on the 3D volumetric data while hovering across its dimensions. All the output values are assembled in a 3D volume as the filter slides through 3D space.

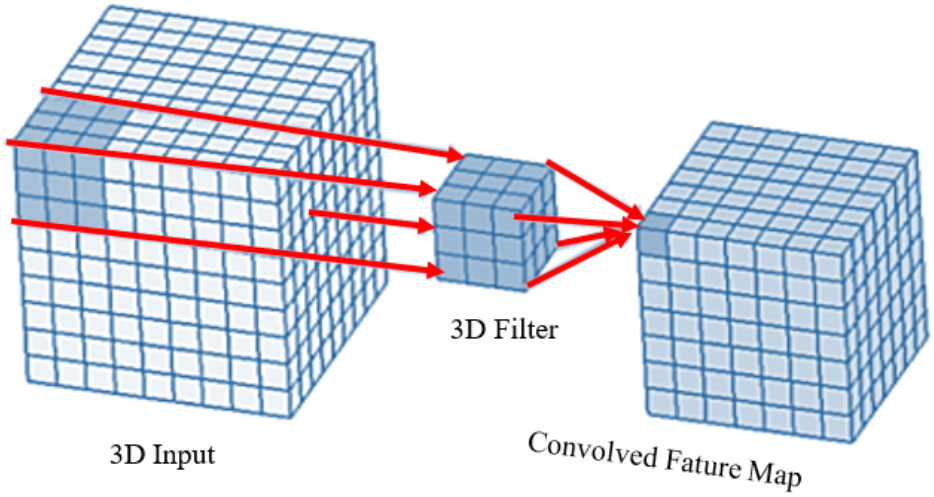

Figure 3.4: 3D Convolution operation

3.1.7 1×1 Convolution

1×1 convolution is a popular approach for performing dimensionality reduction in convolutional neural networks with complex structures. In a 2D space, it can be referred to as convolution with a filter of size 1×1. 1×1 filter slides over an entire image matrix. Every element in the 2-dimensional input image matrix is multiplied by a single number. It is effective in the neural networks that apply larger filters with varying sizes of 7×7, 5×5 and 3×3 on any 3D input data. As the network goes deeper, the number of activation maps increases dramatically and it intensifies the need of computational power. The 7×7, 5×5 and 3×3 filters are expensive as they increase the number of parameters in the network as well as computational requirements. The application of 5×5×32 filter on a 28×28×192 input feature map introduces 120.422 ((28×28×192) × (5×5×32)) million operations in the neural network. 2.4 million operations are required to convolve the same input image with sixteen 1×1 convolution filters. The application of 1×1 filters on the input image produces a feature map of size 28×28×16. The total number of operations gets further reduced to 10 million when the same 5×5×32 filters are applied on the resultant feature map of size 28×28×16. The number of operations decreases from 120.422 to 10 million when a 1×1 convolutional layer with 16 filters is applied before applying 5×5×32 convolution filters on the input image. This way it reduces the number of channels in the feature maps while keeping the prominent features intact. A detailed explanation of the functionality of 1×1 convolution can be learned from the research publication on

Inception V4 [62].

3.1.8 Dropout

During the training period of a neural network, co-dependency among neurons increases which leads to overfitting. The network needs regularization which can prevent overfitting. Dropout is a popular approach for the regularization of a neural network. Regularization penalizes the loss function in the neural network to make it learn a set of weights that are not mutually dependent. In this process, a neuron gets deactivated with a probability p at each training iteration. All inputs and outputs that are connected to the neuron get disconnected. At each training iteration, a neural network revisits the dropped out neurons with a dropout rate of p. A dropped-out neuron might get activated by the neural network at the next training step. Dropout regularizes a neural network by reducing interdependent learning among the neurons. A neural network learns more robust features with the dropout technique.

3.1.9 Batch Normalization

The values of input features might have a wider range starting from a higher number such as -300 to a negative number such as 300. For some other feature maps, the values might range from -10 to 10. The weights related to these inputs will also have a vast range. A neural network might take a longer time to reach a stable state due to this rough distribution of weights. The training of a neural network can be faster if the inputs that it needs to process at each iteration can be normalized. Batch normalization is a regularization algorithm that can improve training speed by normalizing the data in each training batch.

3.2 Recurrent Neural Network

A recurrent neural network is one type of neural network that can process sequences with varying length of inputs with the use of padding.

In an RNN, output in a current step is dependent on the input from its earlier step (Figure 3.5). A simple RNN takes x_t as the first input and the output from previous node as the second input. It performs computation on these two inputs and predicts

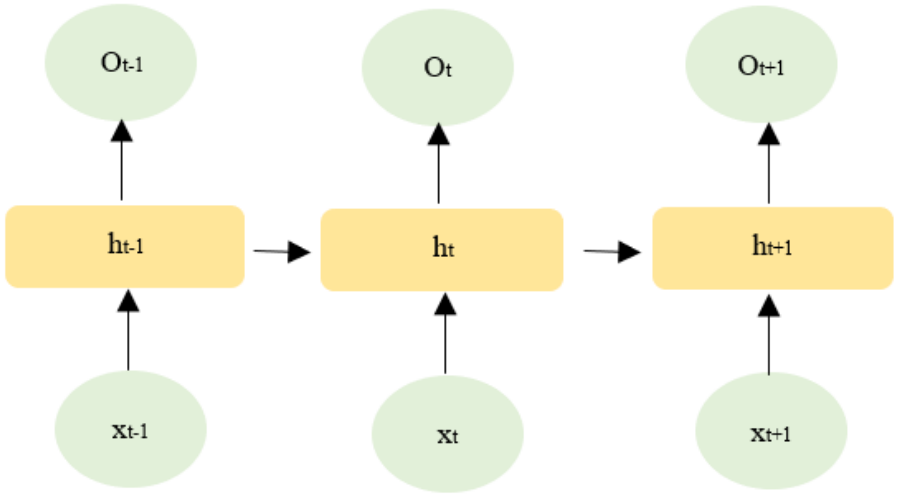

Figure 3.5: Simple RNN

the output. It computes the current hidden state h_t using the previous hidden state h_{t-1} and the current input x_t. The predicted output at any current state depends on all the information that is passed to the network before that state. It remembers some information from the previous inputs in a sequence by computing the hidden states at each time step that enables it to remember the relationship between individual inputs in a sequence.

3.2.1 Long Short-Term Memory (LSTM)

Long Short Term Memory (LSTM) is one particular type of RNN that has resolved the problem of vanishing gradient descent. RNN fails to remember information if the model has to backpropagate to a much earlier time step. LSTM operates using three gates (Figure 3.6): a forget gate, an update gate and an output gate. LSTMs have a unit called cell state that preserves long-sighted context over distant time steps. The straight line running through the top of the diagram is the cell state.

In the forget gate, information from the previous cell state is filtered out using the input from the current step x_t and hidden state representation for the earlier time step. The value from the concatenation of hidden state h_{t-1} and input x_t is passed to a sigmoid function to squash the output in between 0 and 1. The output from the sigmoid gets multiplied by the cell state C_{t-1} from the earlier time step. If the multiplication result is 0 it gets eliminated from C_{t-1} otherwise it is passed to the

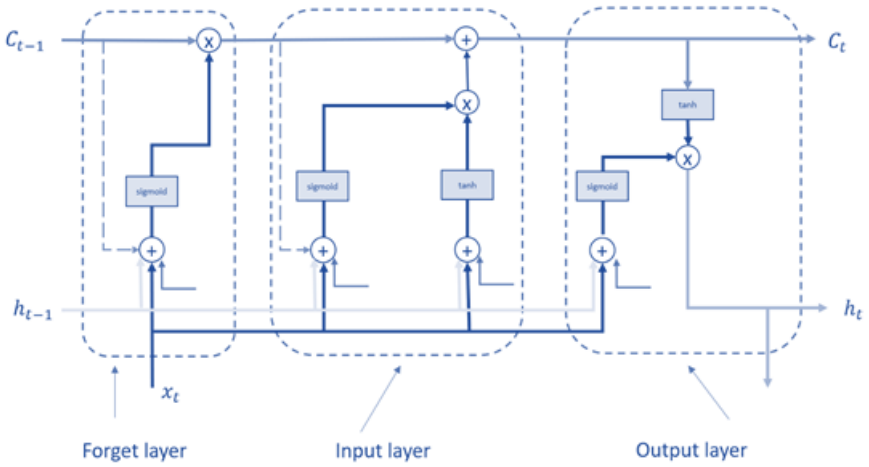

Figure 3.6: Single LSTM layer

next layer.

The concatenation results from the current input x_t and the hidden state h_{t-1} goes through a tanh activation function to select possible candidates for learning context from the sequential data. The same result again is passed to another sigmoid function to identify which candidates or information should be updated to the cell memory C_{t-1}.

In the output gate, another sigmoid function processes the concatenated value from the input x_t and the hidden state h_{t-1} to calculate the output. A tanh function is applied on the cell state C_{t-1} . The result from tanh is multiplied to the calculated outputs to determine the selected outputs that will be passed down as the output produced by the output gate. In an entire LSTM cell, the horizontal line on top is considered as the long-term memory.

3.3 Histogram of Oriented Gradients (HOG)

Histogram of oriented gradients (HOG) is popular as a feature descriptor in image processing and computer vision for object detection. It explores edge orientation in the images. It emphasizes on the shape of any objects in a given image. A HOG computes the gradient of each pixel in the scenery. The magnitude and angle between each pixel in an image are calculated. The gradient of a given image is calculated by

combining the magnitude and angles calculated for each pixel. To compute the HOG features, it produces histograms using the magnitude and orientation of the gradient.

Chapter 4

Methodology

4.1 Model Design

An ensemble of two deep 2D CNN-LSTM models and one 3D CNN model has been employed in this work. A two-dimensional convolutional neural network model with time distributed layer and LSTM is the first model. The second 2D CNN model is developed utilizing the first model as a baseline and adding an inception block in between the time distributed convolutional layers. A simple 3D CNN model is integrated along with the first two models which is designed using 3D Convolutional neural network layers.

The simple 2D CNN has more layers but has less trainable parameters. 2D CNN's are usually not capable of processing a sequence of frames. Although pretrained 2D CNN's such as VGGNet, Inception v3 or AlexNet are not capable of processing consecutive frames together, there is a layer in keras called Time Distributed Layer that allows 2D CNN's to treat a sequence of frames as an input to the 2D keras model.

4.1.1 Basic 2D CNN with LSTM

Videos can be considered as multiple frames recorded at different timesteps. For example, if there are 16 frames extracted from a single input video and if a time distributed convolutional layer is used to extract features from those video frames, the Time distributed layer will treat 16 frames as 16 timesteps and will apply convolution onto each of those 16 frames. The 2D CNN-LSTM model is designed using a Time Distributed Layer and a single LSTM Layer available in Keras. Every convolution

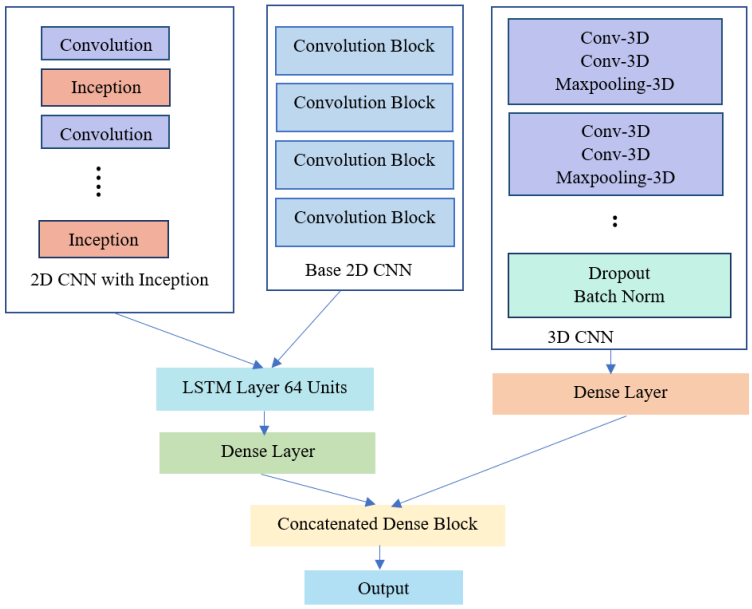

Figure 4.1: Proposed Ensemble Model

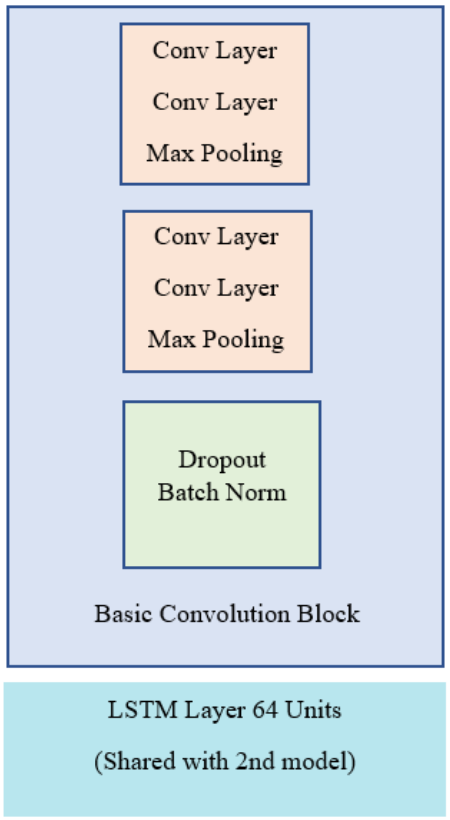

Figure 4.2: 2D CNN-LSTM model with LSTM layer

layer is associated with a Time Distributed layer which can be referred to as Time Distributed Conv layer. A max-pooling layer is included in between the time distributed convolutional layers. Batch normalization has been applied after every two convolutional blocks.

As an advanced regularization approach, a dropout layer is associated with each batch normalization layer in the model. The same convolutional block along with Batch Norm and Dropout layer is repeated a few times to make the CNN structure deeper.

Long Short-Term Memory (LSTM) architecture can detect information from a long-range temporal relationship like standard RNNs. Since videos have both spatial and temporal aspects, the LSTM can extract additional information from the connecting frame sequences which can assist a 2D model in making more accurate predictions. So, an LSTM layer is employed for processing the frame sequences further after features are extracted by the 2D CNN. Features learned from the last convolutional block of this model are passed down to the concatenation layer before it is fed to a shared LSTM layer. Figure 4.2 demonstrates the fundamental structure of the repetitive convolution block.

4.1.2 2D CNN-LSTM with Inception v4 Modules

Inception blocks are incorporated with the Time-Distributed convolutional layers for performing feature extraction at numerous levels of abstraction. An Inception block has been added with each convolutional block to extract features from the video frames. The same convolutional block is used in this model as described in the first base 2D CNN. It is a good strategy to apply convolution with different filters and layers with different configurations for convolving information from the same input data. This approach can make the model highly efficiency in learning objects at different scales and enable it to capture different patterns from the same data source.

This model is built with some modifications to the first 2D CNN model. It has Inception blocks along with each convolutional block which is inspired from Inception V4 [62]. The output layer after the last inception block in Figure 4.3 is concatenated with the output layer from the first model. This concatenated output is then fed to a shared LSTM layer. Figure 4.3 illustrates the structure of the model.

Numerous objects may have a separate distribution of pixels in a scenery. The

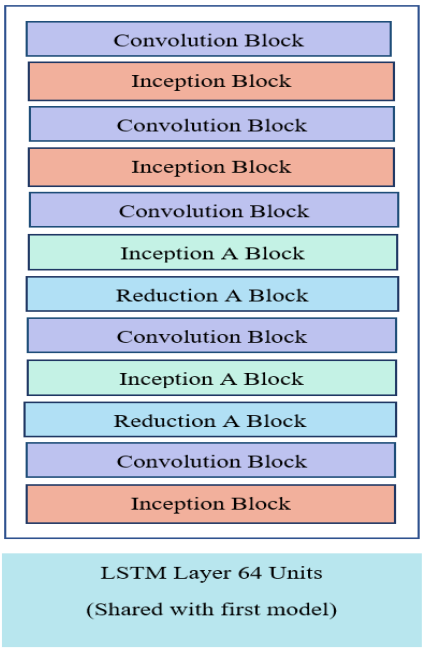

Figure 4.3: 2D CNN LSTM model with modules from Inception V4

same object may appear in an image in a variety of sizes. The use of several filters is great to encapsulate the variations in the object's appearance. Any convolutional neural network may suffer from overfitting with many large filters applied to convolution layers.

Inception networks were built on the idea that a model can handle objects at multiple scales with convolutional filters of different sizes. All the modules from the inception network family make use of 1×1 convolution filters so that the model parameter does not grow promptly with the application of large filters in the convolutional layers. The core motivation behind the use of Inception V4 modules was to make the network wider instead of going deeper while convolving the features with many filters operating on the same input.

This model utilizes the modules from Inception V4 to resolve the issue of computational expenses while efficiently learning patterns from the video frames. All the inception modules are designed to learn from the same input by applying convolution filters parallelly and concatenating the learned features in a single output. Inception V4 also introduces the Reduction modules which are designed to be applied along

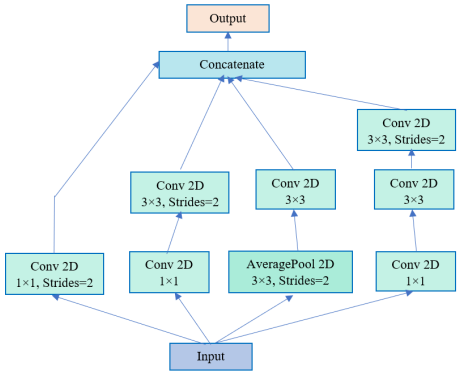

Figure 4.4: Basic Inception module

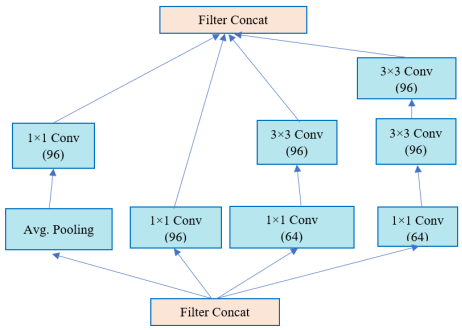

Figure 4.5: Inception A module

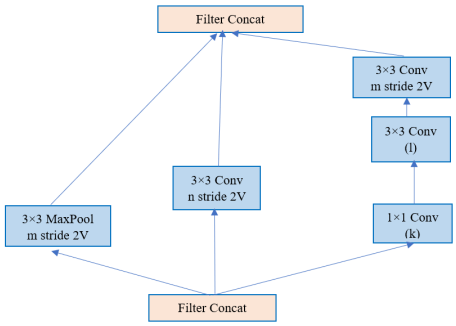

Figure 4.6: Reduction module

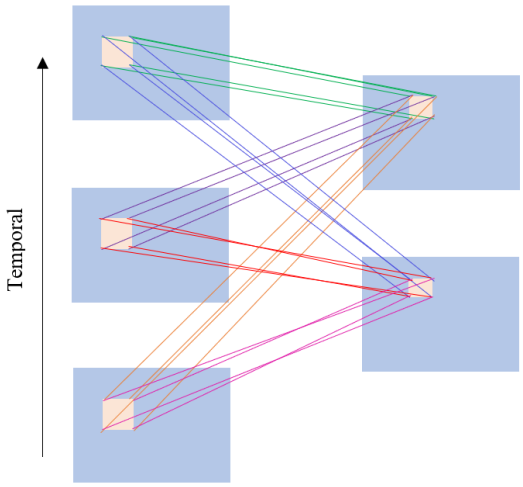

Figure 4.7: Extraction of multiple features from contiguous frames in 3D CNN

with the inception module to manage the size of activation maps [62].

Instead of having a few consecutive transformations through convolutional layers, inception modules comprise of two or three 1×1 convolution layers before the outputs from these layers with the reduced dimensionality can be passed to separate pooling layers and 3×3 convolution layers.

The basic Inception module is depicted in figure 4.4. Figure 4.5 and figure 4.6 shows the inception A module and Reduction module.

4.1.3 Feature Extraction Using 3D Convolutional Neural Network

3D Convolutional Neural Networks are now extensively being used for video classification. 3D CNNs are like the 2D CNN's where features from both spatial and temporal dimensions are extracted by performing 3D convolutions.

The 3D convolution is performed by convolving a 3D kernel to the 3D cube which is formed by stacking multiple contiguous frames together. With 3D convolution, it is possible to capture motion information encoded in multiple contiguous frames. The feature maps in the convolution layer are connected to multiple contiguous frames in the previous layer, thereby capturing motion information. An illustration of feature

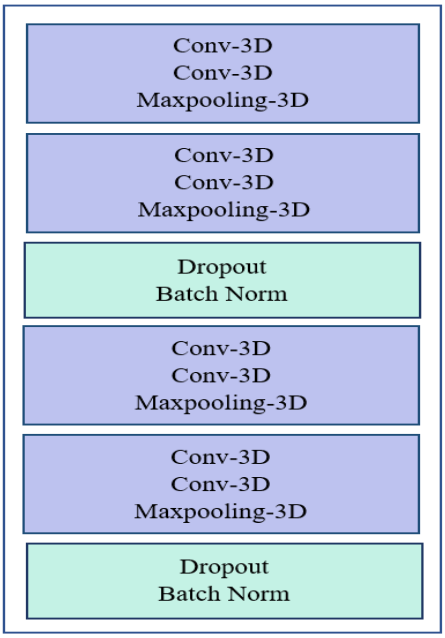

Figure 4.8: 3D CNN

extraction from contiguous frames is given in Figure 4.7.

The 3D convolutional network contains similar convolutional blocks as the first 2D CNN-LSTM model. Instead of 2D max pool layers in 3D CNNs, 3D max pooling layers are used. The structure of the model is demonstrated in Figure 4.8.

4.1.4 Model Ensembling

The complete configuration of the ensemble model is depicted in Figure 4.1. A sequence of 16 frames from each input video is fed to both the 2D CNN-LSTM models and the 3D CNN as a separate input. The same input is processed by each of these models. An earlier work uses two separately trained networks for modeling spatial and temporal features individually [60]. However, this network needs the optical flow vectors to be computed and stored in advance. It also requires the RGB videos to be incorporated with the optical flow vectors as the input to the models. This strategy of processing the input and the design mechanism of the network modules includes excessive memory needs which is impracticable when a large dataset is utilized. Therefore, this work does not utilize separately computed optical flow vectors so that the network can be made end-to-end trainable and efficient.

An advantage of using a Time Distributed layer with the convolution layers is that it learns different aspects such as it detects objects in the frames at first and then learns the orientation of the objects in the next consecutive frames. So, the 2D CNN models can learn spatial features as well as the orientation of objects with the help of Time Distributed Layer and LSTM layer. On the other hand, 3D CNN applies convolution both spatially and temporally. Since 3D CNN can work on temporal features, the LSTM layer is not added anymore. The 2D CNN models apply convolution on different frames such that weights are learned separately but in a temporal manner. In 3D CNN, convolution is applied to all frames together, so the weights are learned in a combined way. So, these two different models learn features in a different manner.

Model ensembling is a powerful technique where predictions or outputs from two or more models are combined together to form a noise invariant and robust model. These models can be different and as diverse as possible. The performance can also be decreased if the two models are highly diverse in nature.

In model ensembling with a basic bagging approach, separate models work on the same training set parallelly. In our case, three models learn from the same input samples parallelly. Thus, three model gets trained simultaneously and the outputs from the last layer of each model are concatenated to produce the final prediction scores of the ensemble model. This is done by concatenating the output from the last dense layer of all 2D and 3D models together. In the ensemble model, all of the 2D and 3D models learn features from the same input in a different way which helps to reduce the variance and improve the performance of video classification to a greater extent.

Chapter 5

Dataset And Preprocessing

5.1 Cataract Dataset

Videos of cataract surgeries were recorded by our medical collaborators [58]. The Cataract dataset consists of real-time and raw surgical videos of varying lengths which were recorded at a frame rate of 29 fps with a resolution of 1920 $\times$ 1080. It contains 197 videos on cataract surgery. It has frame label annotation with the start time and end time of each surgical phase. The annotation for skill levels are based on the appointment status of the surgeon who performed the surgical trial.

The Health Sciences and Affiliated Teaching Hospitals Research Ethics Board at Queen's University cooperated in obtaining the approval of the Institutional Review Board (IRB)/Ethics Committee. Staff and trainee surgeons performed successive cataract surgeries at the Hotel Dieu Hospital located at Kingston Health Sciences Centre, Queen's University, Kingston, Ontario, Canada. The videos of cataract surgery were recorded in the middle of October 2018 and March 2019. Among all residents, only trainee surgeons continuing their last (5th) or second last (4th) year of residency were invited to take part in the event of performing and capturing cataract procedures under the explicit guidance of faculty surgeons.

Informed consent was taken from all patients for cataract surgery and intraocular lens (IOL) implantation. All patients were informed about the involvement of trainee surgeons in the operating room. All staff and trainee surgeons who have taken part in the study were informed about the fact that the surgeries will be video recorded using a Microscope. Informed consent was obtained from all the participating surgeons prior to the commencement of the study. There was no information about the patients available in the microscopic videos that can distinguish any patient exclusively. The

residents responsible for collecting data tracked the surgeons (resident and faculty) to collect identifying data and the complexity of each surgical procedure. This was done to ensure accurate analysis of video data for annotating individual phases and skill levels in the recorded surgeries.

Different patient cases in the study were categorized as either straightforward or complex. The cases that required toric IOL implant, hypermature cataract requiring VisionBlue, Malyugin ring, iris hooks, capsular tension ring (CTR) insertion and posterior capsular rupture (PCR) were identified as Complex Cases. The surgeons reviewed all individual videos to check the video quality and ensure that video recordings were complete and recorded accurately. Recordings where the quality of videos was poor or an incomplete procedure was observed were omitted from the recordings list. Annotation for each selected video was determined by examining the skill level of surgeons who participated in the surgery, surgical technicalities and case-specifics. Video recordings were annotated as either expert-level skills or trainee-level skills. Along with the phase and expertise categories, various operating techniques applied during surgeries and observed in the videos were noted. The steps in the surgeries comprised of the following: clear corneal incisions/Wong incision, dilating cocktail used, continuous curvilinear capsulorhexis (CCC), Hydrodissection, Viscoelasticity, phacoemulsification and nuclear disassembly.

5.2 Dataset Preparation

The data preparation is done following the steps below and depicted in Figure 5.1.

Phase-wise Segmentation:

An annotation file was prepared by the collectors of the dataset where the start time and end time of each surgical phase are marked and noted. A pyhton fucntion is used to trim videos of each surgical phase (Capsulorhexis/Phacoemulsifier/Hydrodissection/Viscoelasticity) from the raw surgery video using the start time and end time of that phase.

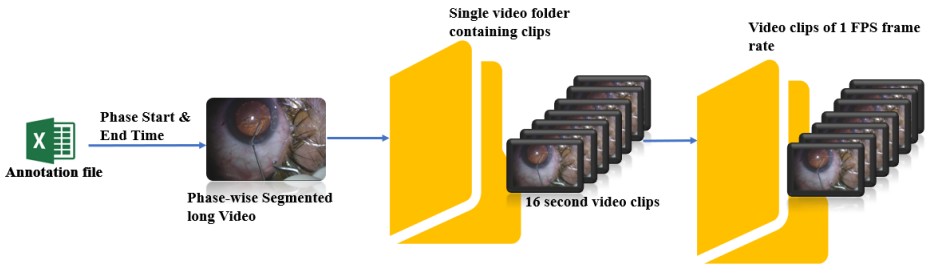

Figure 5.1: Dataset Preparation

Slicing:

Each video is segmented into video clips of equal length with a duration of 16 seconds. The duration of each surgical procedure for different patients was not the same which makes each surgical video related to any specific phase different in length. This is because each patient might have a special condition. Thus, the treatment will be provided based on the patient's situation. The time duration for the completion of cataract surgery may vary from one patient to another. For example, the phacoemulsification procedure for one patient might take 16 minutes whereas for some other patients it could have taken less than 16 minutes. Convolutional neural network requires each input data to be of the same size and shape. The length of the input videos must be equal in order to train a convolutional neural network. The same clip duration is followed for each phase analysis. Each of these video clips representing an entire surgical trial are treated as a single data sample. The level of each of the clips is based on the appointment status of the surgeon who have performed the surgery. If a surgical trail is performed by a staff surgeon, the level of all video clips representing that entire surgical trial will be expert. It will be novice otherwise. The video snippets are labelled as either 0 or 1 where 0 refers to the Novice class and 1 refers to the expert class.

Downsampling:

Each video is downsampled to 1 fps so that there will be only 16 frames in a 16s video. This helps to drop an equal number of redundant frames per second.

Frame Extraction:

A python function is used to extract exactly 16 frames from each video. These 16 frames are then stored in an individual folder for each video with the folders named according to the corresponding video filename.

Temporal Padding:

Temporal padding of the last frame in a video clip is performed so that each video sample can be represented using 16 frames only. There were a few videos for which a sequence of 16 frames was not found. Another python function is used to copy the last frame in each video folder where the total number of frames extracted from a video is below 16 to make up to 16 frames. So, each video sample is represented by 16 frames and can be fed to CNN. Since the original video recordings of surgeries have varying lengths depending on the patient's condition, it was not possible to trim each video related to any phase into video snippets of 16 seconds. Thus, among all the trimmed video snippets extracted from the recording of any phase, there is always one or two snippets that have a duration of fewer than 16 seconds.

5.3 Data Processing Techniques

5.3.1 Up-sampling of minor class

It is always good practice to handle class imbalance between categories in the dataset. Videos of the skill categories have different lengths. The expert category was recognized as a minor class with fewer total snippets than the novice class. Videos from expert data were short. On the other hand, data for novice class was longer than the expert ones which creates a difference between the total number of snippets in expert and novice classes in the training set. Video clips of the minor class were up-sampled by applying a rotation of 5 degrees.

5.3.2 Image Resolution

Further preprocessing is needed before building the models. A resolution of 128×128 is used. Frames are converted to grey scale instead of working with the RGB frame

sequence since RGB frame sequence generates a large number of trainable parameters in the 3D CNN model.

Chapter 6

Implementation Details

The proposed convolutional neural network model is developed in TensorFlow using Keras which is a high-level API that runs on top of the TensorFlow Platform. The implementation is done on Compute Canada Graham cluster using a virtual environment. The model is trained on the tensorflow_gpu package compatible with python version 3.6 with two NVIDIA GPU running parallelly with the environment configuration.

The neural network model is designed by exploring various experimental setups. Changes have been made to the model by observing the differences in the validation accuracy. Initially, four different lengths for sequence were examined to see how the model performance changes with the change in length of frame sequence. A length of 22, 20, 16 and 10 frames were used to find an optimal length to represent the video snippets. It was found that the model performs well with a sequence length of 10-16 frames.

The videos are preprocessed using the OpenCV library. When an image is read by the OpenCV library, it loads and stores the image data in a temporary space using 64-bit floating points. This intensifies excessive memory requirements while the model processes on the data. The data type of the image frames is converted to float 32 to avoid excessive memory consumption by the data while the data generator loads the data into secondary memory during the training of the model. HOG descriptors for image frames are computed using the scikit-image library during data preprocessing.

Keras library provides a layer called Time distributed layer which allows the development of customized time distributed 2D Convolutional layers. It permits applying a layer to every temporal slice of the given input . Time distributed layer is needed to make a Keras 2D CNN process sequence of frames.

ReLU is used as an activation function for the convolutional layers as well as Dense layers. The single LSTM is designed by importing LSTM layer from Keras which has 64 nodes with dropout value of 0.2. The Sigmoid function is used in the output layer as the model predicts binary skill classification score where 0 refers to novice class and 1 refers to expert class.

Keras early stopping was used to monitor the performance on the validation set so that a stable and robust state for the model can be found through training. The ensemble model was trained end-to-end using 100-200 epochs with a different batch sizes of 4, 6 and 8. The batch size of 4 was selected from the observation of model accuracy. The best results were obtained from the model for epochs ranging from 100 to 140. A learning rate of 0.005 and 0.001 were applied which did not work on the dataset. Among different learning rates, 0.0001 was found to be effective for the model so that it learn the patterns from data. The model is trained using RMSprop as an optimizer function with binary cross-entropy being the loss function. A dropout probability of 0.5 was first applied which did not show any notable increase in the model accuracy. The experiment is continued with a dropout probability range of 0.2 - 0.3 to verify if that can make the model learn better. A dropout probability ranging from 0.2 -0.3 is selected for the convolution blocks and the final classification layer after observing the model behavior.

The model has been improved with the use of different data augmentation as well as regularization techniques. The model performance is investigated using regularization techniques at the beginning. There are many mechanisms available for regularizing the network. The basic step starts with increasing the total number of samples in the training set. For large models which are not linear and are deeper, a kernel regularizer is often suggested as the network might get neurons with large weights when different filters and layers are applied.

Regularization is a technique that changes the way a model learns which makes the model generalize better on the testing data. When there is a noise in the training data, the model suffers from an overfitting issue. L2 is a common regularization technique. It is also known as weight decay. L2 regularizer with a value of 0.001 is applied on the kernel to avoid overfitting. It forces the model weights to move towards zero. Hidden neurons with larger weights cause a network to overfit. L2 as kernel regularizer penalizes the neurons with larger weights and those neurons get negligible which in turn reduces the complexity of the neural network.

When an ensemble of neural network models is trained on a large data set, it becomes necessary to design the network with different layers precisely so that the learning strategy of the model can be effective. To further investigate the model behavior, a dropout layer is added along with the batch normalization layer. The combination of dropout and batch norm is followed in all convolution blocks. Batch Normalization with dropout layer is deployed to improve the learning mechanism of the model following the idea of Chen et al. [16]. The authors concluded that a combination of each layer can augment the performance of any model. It enables the model to better regularize on the validation sets as compared to the common practice of using a batch norm layer inside the convolutional blocks.

The proposed model is built using a dataset of 197 cataract videos. Videos of some phases such as Phacoemulsification are 15 to 17 minutes long which is high for a video input to be processed as a sequence of successive frames by a neural network. As the work is implemented using such a large dataset with videos length being long, input modality with the use of optical flow which also requires RGB videos to be incorporated was not preferred. It was important to augment data in such a way that would not exaggerate the memory requirement further but still make the model learn data patterns from input resourcefully. HOG filter is applied on the image frames which can augment the spatial features in the input data without adding up more to the computation resources. HOG is applied on the frames of image sequences that extracts HOG features from the images in the sequence before the sequences are fed to the model. The HOG filter available in the scikit learn library is applied as the preprocessor file that reads the video frames before they are passed to the model.

The skill classification scores are based on the individual clips from the dataset and are not computed from an entirely long surgical video. Five-fold surgical trail-out cross validation strategy was used with same data partitions for training, validation and testing data split. Video snippets were divided into training, validation and test sets. So, for each data split, the samples are not repeated. The video snippets in the test set are reserved for testing the model only after the training and validation is done. Since videos were down-sampled to 1 fps, there is no chance that the same surgical activities will be observed in the video clips of test set as well validation or train set. The reported accuracies and other performance metrics were an average over all folds.

Chapter 7

Results and Discussion

The model for surgical skill assessment is trained and evaluated using four sets of video data from four phases in the cataract surgery distinctly. The phases are Capsulorrhexis, Phcoemulsification, Hydrodissection and Viscoelasticity. Videos are classified into two skill categories. The data are classified by the model as either expert or as novice surgeons.

The validation accuracy of the model is recorded using Capsulorrhexis data for all the network configuration discussed in the implementation section. Once the model configuration and hyperparameter settings are finalized, the validation and test accuracy is computed for all other three phases using the improved version of the model.

Table 7.1 illustrates the skill classification accuracy for various experimental settings that are described in chapter 6. As seen in Table 7.1, training accuracy is 95% which is higher than the 77.77% validation accuracy. The training and validation accuracy exhibited a generalization gap at the beginning which indicates the overfitting issue where the model learns well on the train set but fails to perform well on the validation data. To eradicate the issue, L2 regularization is deployed. A weight decay of 0.001 worked well on the model kernels.

The validation accuracy improves to 82.14% with the regularizer while reducing the gap between the train score and validation score. Apart from making the model more stable, it also reduces the memory and time requirements during training time for the same model where the total number of model parameters changes from 18.6

Table 7.1: Model performance under various network configuration

Approach	Train Acc.	Validation Acc.
Baseline	0.9500	0.7777
L2 Regularizer (0.001)	0.9681	0.8214
BatchNorm + Dropout	0.9685	0.8400
HOG Filter	0.9999	0.8960

Table 7.2: Skill assessment accuracy on all four phases

Cataract Phase	Training Acc.	Valid Acc.	Test Acc.
Capsulorrhexis	0.9900	0.8960	0.8100
Phacoemulsification	1.0000	0.9400	0.8494
Hydrodissection	0.9999	0.9072	0.8130
Viscoelasticity	1.0000	0.9100	0.8300

Table 7.3: Train accuracy for skill assessment on all four phases across five folds

Cataract Phase	Fold1	Fold 2	Fold 3	Fold 4	Fold 5	Average
Capsulorrhexis	1.0000	1.0000	1.0000	0.9800	0.9700	0.9900
Phacoemulsification	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Hydrodissection	0.9800	0.9600	1.0000	1.0000	1.0000	0.9900
Viscoelasticity	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Table 7.4: Test accuracy for skill assessment on all four phases across five folds

Cataract Phase	Fold1	Fold 2	Fold 3	Fold 4	Fold 5	Average
Capsulorrhexis	0.8700	0.8200	0.8000	0.7900	0.7700	0.8100
Phacoemulsification	0.8100	0.8072	0.8000	0.9600	0.8700	0.8494
Hydrodissection	0.7760	0.7590	0.9000	0.8000	0.8300	0.8130
Viscoelasticity	0.8200	0.8202	0.8000	0.8300	0.8800	0.8300

Table 7.5: Sensitivity and AUC on Test Set

Cataract Phase	Sensitivity	AUC
Capsulorrhexis	0.8340	0.8700
Phacoemulsification	0.8833	0.9000
Hydrodissection	0.8473	0.8820
Viscoelasticity	0.8677	0.8897

million to 12 million. Dropout is usually used in the blocks of the dense layers where classification is made. Interestingly, the combined application of batch normalization and dropout in the convolution blocks improves the model's capability to converge faster and perform well on the testing data. There is an improvement of approximately 2% in the validation accuracy from the previous score. For the final experiment in the nestwork configuration, the classification scores for the same dataset are recorded by applying HOG filters on the image frames. This time the model yields more promising results with a training accuracy of 99.99% while validation accuracy boosted up to 89.00% on the Capsulorrhexis data.

The same network configuration and parameter settings are followed for training the model on the video data of all other phases. The same model gets trained individually for each phase of data. The surgical activities performed in all phases and instruments used during surgeries are unique and they vary from phase to phase. So the videos of each of the phases are different. This is done to see how well the model works on different phase data. The average classification results for all phases are demonstrated in Table 7.2. The scores in Table 7.2 are an average over all cross validation folds. Table 7.3 and 7.4 shows the training and test accuracy for skill assessment across all cross validation folds.

Table 7.5 delineates further analysis of the evaluation metrics that are necessary for quantifying the effectiveness of the model.

The skill evaluation scores on the Capsulorhexis data reveal the proficiency of the proposed ensemble model. To better visualize how the model behavior improved with the experiments, the learning curves before and after the application of regularization techniques have been depicted in Figure 7.1 and Figure 7.2.

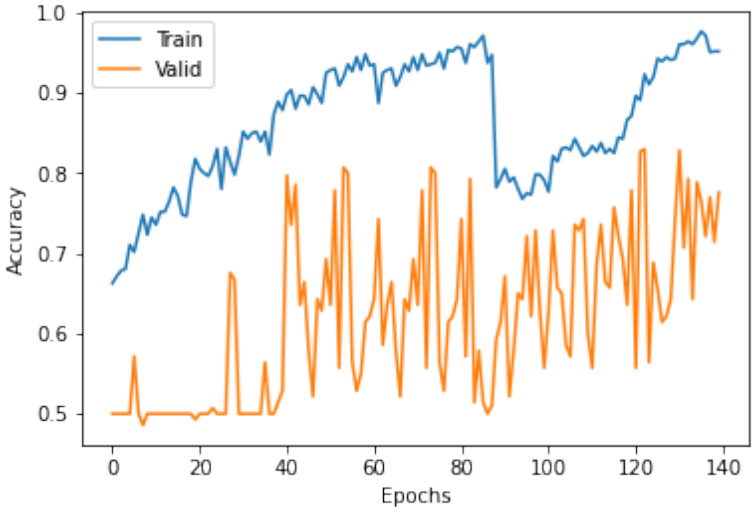

Figure 7.1: Baseline Accuracy

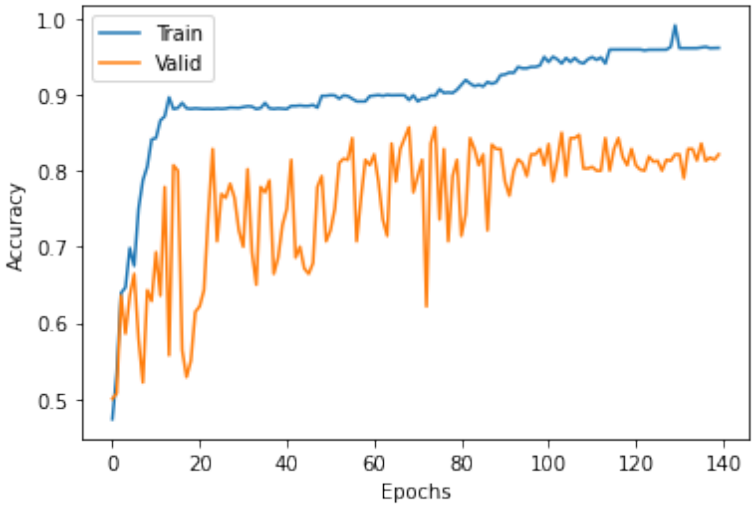

Figure 7.2: Accuracy with L2 Regularizer.

From figure 7.1, it can be seen that the learning curve shows some unusual spikes around 60-80 epochs. The gap between the train and validation curve discloses that the model needs modifications for converging to a stable state.

The learning curve for the validation set in figure 7.2 shows some improvement over the curve in Figure 7.1 where it starts to converge after 80 epochs. The model learns better after 80 epochs and starts to demonstrate an increasing trend while reducing the generalization gap. The validation and test accuracy for the other 3 phases were consistent to the model accuracy for Capsulorrhexis data(Table 7.2). The model revealed a validation accuracy of 94.00% and a test score of 84.94% on the Phacoemulsification phase. The results for the other two phases showed a consistent result with the test accuracy for Hydrodissection and Viscoelasticity being at 81.30% and 83.00% respectively.

The highest sensitivity of 88.33% and AUC score of 90.00% were observed on the dataset for Phacoemulsification phase. The model showed a notable result with an average AUC score of 88.00 % over all data for all phases which exemplifies a prominent model performance. The model shows an average sensitivity score of 85.80% across datasets for four phases using 16 seconds of video snippets. These scores indicate that the model will be able to distinguish between expert and novice surgeons on the data well.

7.1 Discussion

The model was initially developed for identifying operative skills in Capsulorrhexis data. When a test accuracy of 82.00% was observed on one phase, the project work was extended using the data from Phacoemulsification, Hydrodissection and Viscoelasticity. The same model is further trained and assessed on the other phases to verify how well it generalizes over data from different phases where different surgical actions are performed. The surgical trials from each phase include variation in the data where actions performed in one phase such as Capsulorrhexis are different from the steps performed in Phacoemulsification. For example, the tools used in Capsulorrhexis is different from the tools used in Phacoemulsification, Viscoelasticity and Hydrodissection. The time that it takes a surgeon to perform each of these surgical phase is different. The videos for Capsulorrhexis are about 156.3 (average) seconds long. The data from Phacoemulsification has an average duration of 729.5 seconds

which includes a huge variation in the surgical activities performed on the videos for each unique phase. The duration of surgical procedures for the other two phases is also unique from the first two phases mentioned above. The videos of Viscoelasticity have an average duration of 54.7 seconds. The data from Hydrodissection have an average duration of 73.3 seconds long. The model resulted in a test accuracy of 81.00% on Capsulorrhexis data which was improved to 84.94% when the same network infrastructure was tested using Phacoemulsification data. A test accuracy of 83.00% was recorded on the Viscoelasticity data which was higher than the score observed for Capsulorrhexis data on which the model was first trained. The consistency in the overall results for all phases implies that the model is robust to different types of surgical phases. However, the difference between training and testing accuracy for all phases can be overlooked (Table 7.2) as a generalization gap between from a neural network perspective. Overfitting can also occur if there is any ambiguity in the data which can include a generalization gap. This is because a high heterogeneity in the distribution of training data samples can also cause the problem where a neural network model fails to learn true data patterns. This could be true in our case as the levels for skill assessment were based on the appointment status of the surgeons. A staff surgeon might perform a surgery as like as a novice surgeon instead of revealing some expertise. The same can happen in the surgical videos of novice class. So there is a higher chance there will be noise in the surgical videos that is leading the model to have a generalization gap between training and testing result. We have observed that the human raters fails to agree on the same skill levels in a similar dataset like ours for the 29% of the time. So, the model performance is still sufficient as compared to human-based skill assessment scores.

There were 197 surgical trials in the dataset which also indicates that there is a variation in the data samples. Since the ensemble model yielded significant accuracy on such a dataset with data samples for four types of surgical phases, it can be speculated that it will work well on data from different types of surgeries as well.Five-fold trial-out cross validation strategy was used with same data partitions. This implies that the model can generalize to never before seen data.

Real time prediction was obtained on some videos to check how well the model can predict score. To check how well the model can perform in real-time, the prediction result from a surgical video is depicted in Figure 7.3. A 147-second video that belongs to the Capsulorrhexis phase was loaded and trimmed into 16s clips. The model's

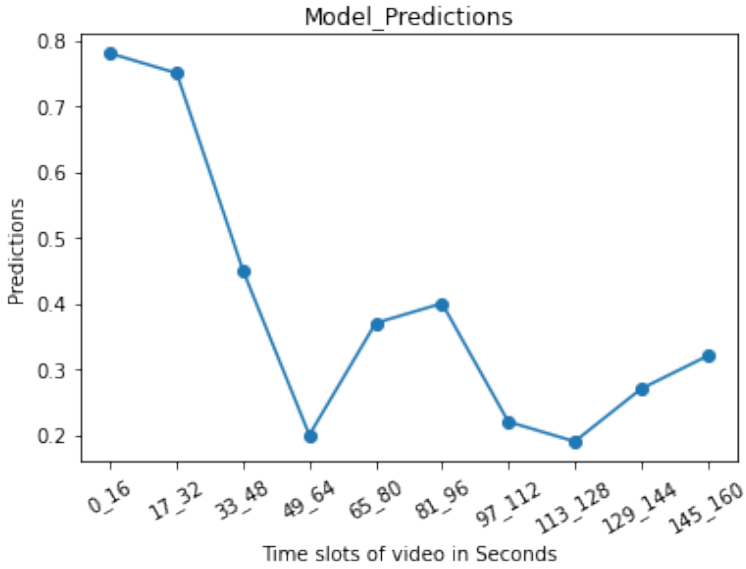

Figure 7.3: Real time prediction of the model on all the 16-second video snippets representing a never seen before video sample (Model identifies novice). Video clip duration (e.g. clip 1 : 0-16 seconds, clip 2: 17-32 seconds) for all 16 second clips are plotted along x axis.

prediction score for each of the 16-second clips representing an entire surgical trial was plotted on the y-axis. In the x-axis, the duration of the video clip's was plotted as video from 0-16 seconds, 16-32 seconds and up to 160 seconds. Since each of the video snippets should be 16 seconds long to be processed by a convolutional neural network, the video length was extended to 160 seconds by temporally padding the first snippet with 13 frames. The skill level for an expert is 1 and 0 indicates a novice level performance. We have considered 0.5 as the threshold for identifying each skill category, since score 1 indicates expert and 0 indicates the novice class. So, when the model score is below 0.5 it can be considered that the video clip duration or video clips for which the scores are obtained can be rated as novice level performance. A model score of above 0.5 indicates that the model is identifying an expert surgeon.

It is visible that for the first two video clips (video clip duration 0-16s and 17-32s) the prediction score is higher than 0.5. So, in those two clips, the model prediction indicates the surgeon performed as like as an expert. However, for the video clips with a duration of 49-64 and 97-160, the score is within 0.3-0.45 which implies that the surgical performance during that duration was identified as a novice level expertise.

Similarly, the model's prediction score for each of the 16-second clips representing

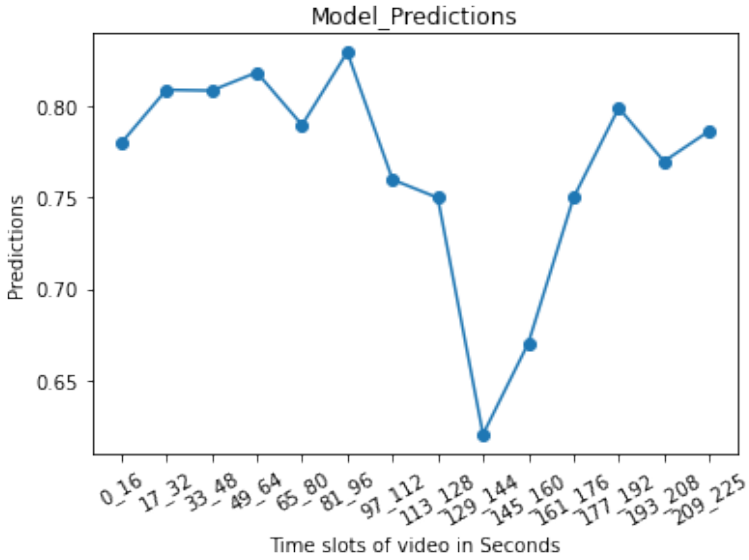

Figure 7.4: Real time prediction of the model on all the 16-second video snippets representing a never seen before video sample (Model identifies expert).

another surgical trial was plotted on the y-axis in figure 7.4 where the model identifies the surgical trial as an expert level performance. It can be seen in figure 7.4 that the skill classification score ranges from 0.77-0.85 for 0-64 seconds which falls below 0.75 at 97 seconds. This downward trend continued to move within 0.62-0.66 till 160 seconds. The skill classification score again surges to 75 and continues to exhibit an upward trend till the end of the procedure. The prediction score lies within a range of 0.62-0.84 which is above 0.5. The scores for all the video clips are above 0.5 and they are close to 1 which is the level for expert class. Thus, the model designates the surgeon's ability as an expert-level surgical performance. The prediction scores on the unseen data reveal that the model can perform well on unknown data.

In the figure for real-time prediction (Figure 7.3), it can be seen that a high score is observed at the beginning indicating an expert level performance. It seems difficult to conclude whether the surgeon is an expert or novice until a lower score of just above 0.2 is observed from 49 seconds to 64 seconds. As the procedure continues, low skill classification score is predicted by the model for a time duration in between 97 seconds to 160. The changes in the score at the end reveal the true skill level of the surgeon where the score ranges in between 0.2 - 0.3 indicating a novice level proficiency. Since the skill level for novice class was set to 0, the model scores ranging from 0.2

- 0.3 indicates a novice class which also matches the ground truth for the surgical trial in figure 7.3. This prediction can be utilized to identify exactly at what time duration the surgeon showed a novice level expertise in the surgery so that a trainee can practice more on that specific part of the surgery. This can add a significant value to the surgical training curriculum in the operating room considering the time requirement. The model can save the unnecessary time spent after discovering when a surgeon's performance went wrong in an entirely long surgery apart from the fact that the neural network only takes time during training which is quite low in the inference time as opposed to manual grading.

7.2 Interpretation

We compare our results to some other related work who also performed skill assessment using different strategies.

In a different study, expert ophthalmologists tried to grade operative competency in a similar cataract dataset like ours based on 4 assessment criteria in order to come up with the skill levels for the surgeries [58]. According to the analysis of the expert ratings, human graders agreed on the same skill levels as the levels we have on the annotation file of our dataset for 71% of the time. So, human graders failed to agree on the same skill levels for almost 30% of the time. Therefore, an average testing accuracy of 82.00% is adequate to consider the fact that the proposed model can be even better than the human rating consistency.

Earlier research work reported a similar accuracy for skill classification by utilizing motion trajectories obtained from tool tip positions in the surgery videos [41]. Their work was tested using approximately 100 Capsulorrhexis data where tool tip information was considered for representing skill levels. Our model yields a reasonable accuracy on a vast dataset comprising of almost 200 surgical trials. Our ensemble model was tested using data from four procedures in cataract surgery. It provided a higher accuracy of 84.94% and 83.00% on two other datasets from the procedure recordings of Phacoemulsification and Viscoelasticity. It does not need tool trajectory computations from raw surgeries for encoding operative expertise.

Another skill analysis was conducted on our dataset in an earlier experiment using a late supervision approach [58]. This work was not included in the literature review since it is still under review and not published in any conference or journal yet. The

highest skill classification accuracy achieved from the earlier work was 63.30%. The outcome from our work indicates a substantial improvement over the earlier results for skill assessment.

7.3 Limitation

One of the limitations of this project is that there could be an inconsistency in the ground truth levels for skill classification. There were seven faculty surgeons and five trainee surgeons who participated in the study for recording the surgical procedures. For the annotation of skill levels, trials operated by staff or faculty surgeons were annotated as an expert-level performance. The rest of the video recordings were marked as novice-level performances as they were conducted by the resident or trainee surgeon. It is possible that some of the staff or faculty surgeons might not be experts. Similarly, it can also happen that a resident or trainee surgeon might have an expert level of operative skill. Thus, an expert surgeon might conduct a surgery using novice-level technical skills. A novice surgeon might perform as like as an expert surgeon. The annotation of the videos with skill levels based on the appointment status of the surgeons might not be precise. This indicates that our ground truths for skill levels are prone to error.

The videos were recorded in the same hospital set up but it includes many surgical trials. The recordings were obtained using the same microscope with the same configuration. The dataset does not include trials from many hospitals with different hospital set up. The tools that are used by the surgeons were from several manufacturing companies which also poses a limitation in the dataset. Other hospitals might have tools from companies that were not used by the surgeons in our dataset.

This work is implemented on Compute Canada as the dataset size was large and was not portable to be stored in any other repository apart from cloud storage. The model can not be trained on a large dataset using a GPU with memory lower than 8GB. To train the model on a large dataset like ours with an image sequence longer than 22 frames and a resolution of a minimum 250 by 250 pixels, a gpu with more than 16 gb memory capacity is needed. Having said that a 16 GB memory requirement for a large dataset is still not expensive as a gpu of 16 gb memory is available as opposed to 32 or 64 GB GPU. The model might not work well on a dataset with a small number of training samples. The model is not designed to predict skill classification

scores on an entire surgery video as it will require excessive memory to train a model on an entirely long frame sequence where the sequence will consist of more than 22 frames.

We have performed trial-out cross validation where the surgical trials are performed by 7 staff and 5 resident surgeons. So there are video clips in all data split where the same surgeon performed the surgical activities in the videos of training , validation and test set. A user out or surgeon out cross validation strategy would have been better than the trial out strategy. Since we have few surgeons who have performed the surgical trials, it was not possible to follow surgeon out cross validation strategy.

Chapter 8

Conclusion and Future Work

The accuracy scores of the ensembled model reveal that the model is effective in capturing intraoperative skills in cataract Surgeries. An ensemble of 2D and 3D convolutional networks with different sizes of convolution filters can successfully study technical skills in cataract surgeries without using tool motion trajectories for evaluating expertise in surgeries. Histogram descriptors turned out to be effective for augmenting visual appearance in the data that improved the model's ability to assess surgical proficiency with an increase in the accuracy. One of the noteworthy achievements of this work is obtaining consistent results from the four procedures of cataract surgery which demonstrate how well the model can predict scores on different phases.

For future work, there is inconsistency in skill annotation in the dataset since the skill levels are based on the appointment status. Furthermore, the surgical trials were not performed under the supervision of any expert surgeon. Thus, the skill level annotations were not performed by seeking any expert opinion who are already working in the field. So the skill levels can be revised and annotated by assigning expert surgeons to verify the ground truths. A manual skill assessment can be done to further validate the ground truths.

Data from more than one hospital can be included in the dataset to formulate a more generalized algorithm for surgical skill recognition. The surgery videos can be recorded with a more zoomed-in configuration in the microscope which can dramatically boost up model accuracy.

An ablation study can be performed to see how each individual model in the ensemble model work. The data from all phases are different and so it is unreasonable to build a model that will be trained on one phase and tested on a totally different phase. However, a model can be designed where the model will identify each phases

at the beginning and then will select which model to utilize for performing skill assessment on each detected phase.

We would like to include the standard deviation for all the results in the five cross validation for future work. We did not use any RGB image frames and thus an RGB image sequence can be utilized to see if that has any added value. Our model can be trained on a 32 or 64 GB gpu with a higher resolution which might increase the performance of the model. A recurrent network-based model can be studied to see if it outperforms the ensemble model. A video classifier can be experimented to predict on an entirely long surgery video which is possible with high computation resources.

List of References

[1] "http://hyperphysics.phy-astr.gsu.edu/hbase/vision/eye.html."

[2] "http://hyperphysics.phy-astr.gsu.edu/hbase/vision/eye.html."

[3] "National grouping system categories report, canada, 2003-2004 .National Physician Database."

[4] "Canadian ophthalmological society cataract surgery clinical practice guideline expert committee.canadian ophthalmological society evidence-based clinical practice guidelines for cataract surgery in the adult eye." vol. 43, 2008, pp. S7–S33. [Online]. Available: http://www.supereyecare.com/residents/COSCataractGuidelines.pdf

[5] S. W. A. Raju and J. Huang, "M2cai surgical tool detection challenge report," 2016. [Online]. Available: http://camma.u-strasbg.fr/m2cai2016/reports/Raju-Tool.pdf

[6] N. Ahmidi, L. Tao, S. Sefati, Y. Gao, C. Lea, B. B. Haro, L. Zappella, S. Khudanpur, R. Vidal, and G. D. Hager, "A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery," vol. 64, no. 9, 2017, pp. 2025–2041.

[7] H. Al Hajj, M. Lamard, K. Charrière, B. Cochener, and G. Quellec, 2016. [Online]. Available: https://cataracts.grand-challenge.org/

[8] ——, "Surgical tool detection in cataract surgery videos through multi-image fusion inside a convolutional neural network," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2017, pp. 2002–2005.

[9] J. Alvarez-Valle, "https://www.microsoft.com/en-us/research/project/medical-image-analysis/."

[10] D. P. Azari, L. L. Frasier, S. R. P. Quamme, C. C. Greenberg, C. M. Pugh, J. A. Greenberg, and R. G. Radwin, "Modeling surgical technical skill using expert assessment for automated computer rating," vol. 269, 2019, p. 574–581.

[11] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *European Conference on Computer Vision (ECCV)*, 2006.

[12] S. Bhardwaj, M. Srinivasan, and M. M. Khapra, "Efficient video classification using fewer frames," 2019, pp. 354–363.

[13] S. Bodenstedt, D. Rivoir, A. Jenke, M. Wagner, S. T. Mees, J. Weitz, and S. Speidel, "Active learning using deep bayesian networks for surgical workflow analysis," vol. 14, 2019, pp. 1079–1087.

[14] I. L. C. Schuldt and B. Caputo, "Recognizing human actions: a local SVM approach." International Conference on Pattern Recognition (ICPR), 2004, p. III: 32–36.

[15] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4724–4733.

[16] G. Chen, P. Chen, Y. Shi, C.-Y. Hsieh, B. Liao, and S. Zhang, "Rethinking the usage of batch normalization and dropout in the training of deep neural networks," vol. abs/1905.05928, 2019.

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 886–893 vol. 1.

[18] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proceedings of the 9th European Conference on Computer Vision - Volume Part II*, ser. European Conference on Computer Vision (ECCV). Berlin, Heidelberg: Springer-Verlag, 2006, p. 428–441. [Online]. Available: https://doi.org/10.1007/11744047_33

[19] G. Davis, "The evolution of cataract surgery." *Missouri medicine*, vol. 113(1), pp. 58–62, 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6139750/

[20] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, and L. V. Gool, "Temporal 3d convnets: New architecture and transfer learning for video classification," vol. abs/1711.08200, 2017.

[21] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005, pp. 65–72.

[22] J. Donahue, L. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," 2015, pp. 2625–2634.

[23] C. E. Glarner, Y.-Y. Hu, C.-H. Chen, R. G. Radwin, Q. Zhao, M. W. Craven, D. A. Wiegmann, C. M. Pugh, M. J. Carty, and C. C. Greenberg, "Quantifying technical skills during open operations using video-based motion analysis.surgery. 2014 sep;156(3):729-34." PubMed, 2014. [Online]. Available: https://doi.org/10.1016/j.surg.2014.04.054

[24] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1933–1941.

[25] I. Funke, S. Mees, J. Weitz, and S. Speidel, "Video-based surgical skill assessment using 3d convolutional neural networks," vol. 14, 05 2019.

[26] W. F¨orstner and E. G¨ulch, "A fast operator for detection and precise location of distinct points, corners and centers of circular features." Intercommission Workshop of the Int. Soc. for Photogrammetry and Remote Sensing, Interlaken, Switzerland, 1987.

[27] K. Golnik, H. Beaver, V. Gauba, A. Lee, E. Mayorga, G. Palis, and G. Saleh, "Cataract surgical skill assessment," vol. 118, no. 2. Elsevier, Jan. 2011.

[28] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse sequential subset selection for supervised video summarization," ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 2069–2077.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[30] Y. Guo, G. Zhao, and M. Pietikäinen, "Texture classification using a linear configuration model based descriptor," in *BMVC*, 2011.

[31] H. Hajj, M. Lamard, P.-H. Conze, S. Roychowdhury, X. Hu, G. Marsalkaite, O. Zisimopoulos, M. Dedmari, F. Zhao, J. Prellberg, M. Sahu, A. Galdran, T. Araújo, D. Vo, C. Panda, N. Dahiya, S. Kondo, Z. Bian, A. Vahdat, and G. Quellec, "Cataracts: Challenge on automatic tool annotation for cataract surgery," vol. 52, 11 2018.

[32] C. G. Harris and M. J. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[34] G. Huang, Z. Liu, and K. Weinberger, "Densely connected convolutional networks," 08 2016, p. 12.

[35] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," 2017, pp. 2261–2269.

[36] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim, "TGIF-QA: Toward spatio-temporal reasoning in visual question answering," 2017, pp. 1359–1367.

[37] A. Jin, S. Yeung, J. Jopling, J. Krause, D. Azagury, A. Milstein, and L. Fei-Fei, "Tool detection and operative skill assessment in surgical videos using region-based convolutional neural networks," 2018.

[38] A. Z. Karen Simonyan, "Very deep convolutional networks for large-scale image recognition," vol. abs/1409.1556, 2015.

[39] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," 2014, pp. 1725–1732.

[40] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, A. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," vol. abs/1705.06950, 2017.

[41] T. S. Kim, M. O'Brien, S. Zafar, G. Hager, S. Sikder, and S. S. Vedula, "Objective assessment of intraoperative technical skill in capsulorhexis using videos of cataract surgery," vol. 14, 2019, pp. 1097–1105.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," vol. abs/1412.6980, 2015.

[43] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *International Conference on Computer Vision*, 2011, pp. 2556–2563.

[44] I. Laptev, "On space-time interest points," vol. 64, 2005, pp. 107–123.

[45] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," vol. 86, no. 11, 1998, pp. 2278–2324.

[46] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," 2017, pp. 936–944.

[47] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007.

[48] T. Lindeberg and L. Bretzner, "Real-time scale selection in hybrid multi-scale representations," in *Scale Space Methods in Computer Vision*, L. D. Griffin and M. Lillholm, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 148–163.

[49] A. S. M. Sahu, A. Mukhopadhyay and S. Zachow, "Tool and phase recognition using contextual cnn features," 2016.

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

[51] M. Mishra, 2020. [Online]. Available: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

[52] N. Padoy and A. P. Twinanda. [Online]. Available: http://camma.u-strasbg.fr/m2cai2016/

[53] O. Poquet, L. Lim, N. Mirriahi, and S. Dawson, "Video and learning: A systematic review (2007–2017)," in *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, ser. LAK '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 151–160. [Online]. Available: https://doi.org/10.1145/3170358.3170376

[54] J. Prellberg and O. Kramer, "Multi-label classification of surgical tools with convolutional neural networks," in *International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.

[55] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, p. 91–99.

[56] S. Resnikoff, D. Pascolini, D. Etya'ale, I. Kocur, R. Pararajasegaram, G. Pokharel, and S. Mariotti, "Global data on visual impairment in the year 2002," vol. 82, 12 2004, pp. 844–51.

[57] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," vol. 115, no. 3. USA: Kluwer Academic Publishers, dec 2015, p. 211–252. [Online]. Available: https://doi.org/10.1007/s11263-015-0816-y

[58] J. Ruzicki, M. Holden, S. Cheon, T. Ungi, R. Egan, and C. Law, "Use of machine learning to assess cataract surgery skill level with tool detection."

[59] Y. Sharma, T. Ploetz, N. Hammerla, S. Mellor, R. McNaney, P. Olivier, S. Deshmukh, A. McCaskie, and I. Essa, "Automated surgical osats prediction from videos," 04, pp. 461–464.

[60] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014.

[61] N. Sokolova, K. Schoeffmann, M. Taschwer, D. Putzgruber-Adamitsch, and Y. El-Shabrawi, "Evaluating the generalization performance of instrument classification in cataract surgery videos," 12 2019, pp. 626–636.

[62] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *AAAI*, 2017.

[63] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[64] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[65] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4489–4497.

[66] A. P. Twinanda, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy, "Single- and multi-task architectures for tool presence detection challenge at m2cai 2016," vol. abs/1610.08851, 2016.

[67] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy, "Endonet: A deep architecture for recognition tasks on laparoscopic videos," vol. 36, 2017, pp. 86–97.

[68] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," 2011, pp. 3169–3176.

[69] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.

[70] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014.

[71] B. Zhang, S. Wang, L. Dong, and P. Chen, "Surgical tools detection based on modulated anchoring network in laparoscopic videos," vol. 8, 2020, pp. 23 748–23 758.

[72] J. Zhu, J. Luo, J. M. Soh, and Y. M. Khalifa, "A computer vision-based approach to grade simulated cataract surgeries," vol. 26, 2014, pp. 115–125.

[73] Y. Zhu, Z. Lan, S. D. Newsam, and A. G. Hauptmann, "Hidden two-stream convolutional networks for action recognition," vol. abs/1704.00389, 2017. [Online]. Available: http://arxiv.org/abs/1704.00389

[74] A. Zia, L. Guo, L. Zhou, I. Essa, and A. M. Jarc, "Novel evaluation of surgical activity recognition models using task-based efficiency metrics," vol. 14, 2019, pp. 2155 – 2163.

[75] A. Zia, A. Hung, I. A. Essa, and A. M. Jarc, "Surgical activity recognition in robot-assisted radical prostatectomy using deep learning," vol. abs/1806.00466, 2018. [Online]. Available: http://arxiv.org/abs/1806.00466

[76] A. Zia, Y. Sharma, V. Bettadapura, E. L. Sarin, M. A. Clements, and I. Essa, "Automated assessment of surgical skills using frequency analysis," in *Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 - Volume 9349*. Berlin, Heidelberg: Springer-Verlag, 2015, p. 430–438. [Online]. Available: https://doi.org/10.1007/978-3-319-24553-9_53

[77] A. Zia, Y. Sharma, V. Bettadapura, E. L. Sarin, and I. Essa, "Video and accelerometer-based motion analysis for automated surgical skills assessment," vol. 13, 2018, pp. 443–455.

[78] A. Zia, Y. Sharma, V. Bettadapura, E. L. Sarin, T. Ploetz, M. A. Clements, and I. Essa, "Automated video-based assessment of surgical skills for training and evaluation in medical schools," vol. 11, 2016, pp. 1623–1636.