CLOUD-BASED LECTURE CAPTURING SYSTEM: CASE STUDY

H.M.S.V. Mudalige

IT 15 0210 76

Degree of Bachelor of Science

Department of Information Technology

Sri Lanka Institute of Information Technology

October 2018

CLOUD-BASED LECTURE CAPTURING SYSTEM:

A CASE STUDY

H.M.S.V. Mudalige

IT 15 0210 76

Dissertation submitted in partial fulfillment of the requirements for the degree of Science

Department of Information Technology

Sri Lanka Institute of Information Technology

October 2018

DECLARATION

I declare that this is my own work and this dissertation1 does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

The above candidate has carried out research for the B.Sc. Dissertation under my supervision.

Signature of the supervisor: Date:

ABSTRACT

Today, with the rapid growth of technology and usage of internet services, e-learning has become one of the latest trends in the education sector. As a result, students tend to prefer e-learning than being physically present in a lecture due to multiple reasons. This document examines an innovative approach in the form of smart cloud-based system to enhance the current e-learning procedures, particularly in universities. The "Lecture Capturing System" is a cloud-based web application which uses enhanced techniques to provide an interactive e-learning experience to users of the system. It has the ability to support multiple enterprise customers. It uses a facial recognition-based authentication process to allow remote users to login to the system. A Pan-Tilt-Zoom (PTZ) IP camera captures and tracks the lecturer during the lecture session, moves the camera accordingly and this is streamed live to remotely logged-in students. The lecturer can also share the computer screen if required. Lecturer can share the complete screen or specific application window with webcam view embedded into the screen share if required. The camera intelligently identifies specific gestures performed by the lecturer to rotate towards the audience with the aid of gesture analyzing algorithms. Attendance of remote online students is marked automatically during a live-streaming lecture by using multiple facial recognition processes executing on the server. Offline recording of lectures is also supported after which the video is split into a series of chapters/thumbnails and the audio is converted to text; each chapter representing a presentation slide and the relevant text. Bandwidth and quota are managed intelligently to ensure the best possible transmission rate with minimum data consumption in order to avoid filling the link to capacity which would result in network congestion and poor performance of the network. This is a revolutionary system which is capable of taking e-learning to the next level as it provides a complete classroom experience and makes the whole learning and teaching process efficient and relaxing.

ACKNOWLEDGEMENT

The work described in this report was carried out as my 4th year research project for the subject Comprehensive Design Analysis Project. The completed final project is the result of combining all the hard work of the group members and the encouragement, support and guidance given by many others. Therefore, it is my duty to express my gratitude to all who gave me the support to complete this major task.

I am deeply indebted to our supervisor Dr. Malitha Wijesundara and our Lecturers of Sri Lanka Institute of Information Technology whose suggestions, constant encouragement and support in the development of this research, particularly for the many stimulating and instructive discussions. I am also extremely grateful to our external supervisor Mr. Pramadhi Atapattu, who gave suggestions, feedback and help to carry out this research and for all the encouragement and guidance given.

I also wish to thank all our colleagues and friends for all their help, support, interest and valuable advices. Finally, I would like to thank all others whose names are not listed particularly but have given their support in many ways and encouraged me to make this a success.

Table of Contents

LIST OF FIGURES

1. INTRODUCTION

1.1 Problem to be addressed

Students are still used to physically going to a lecture and listening to the lecturer in real time and making notes of his/her teachings, what s/he sketches on whiteboards or his/her presentation. We have no way of knowing exactly how the current state of the whiteboard came to be after some time.

And of course, we have no way of reviewing what the lecturer has told the students if they forgot to write it down or failed to remember. So, there is a higher probability that students forget what the lecturer said or taught the very next day if they were not properly documented. What if a student gets sick or due to unavoidable circumstances misses a lecture, how is s/he going to learn the missed session? Something like a live streaming with recording session of a lecture would help all the students even if they were present in the lecture itself. The ability to see what they have missed if the student comes to the lecture after it started, a way to refresh their memories before attending the next lecture would result in a huge improvement academically.

Sometimes the students in the back of the classroom cannot properly see the lecturer or the whiteboard resulting poor learning outcomes. Even there is a camera set to show the lecturer it will be permanently mounted and students won't have a proper view most of the time.

1.2 Background Context

Today, communicating with internet and doing things right from our home is becoming more and more practical because it makes day-to-day life very easier. People always try to find a way to use the easiest way to complete their tasks without utilizing too much time and energy.

The art of learning should evolve over time. Even with the latest technology today, we are still used to physically going to a lecture and listening to the lecturer in real-time and making notes of his/her teachings, what the lecturer sketches on whiteboards or his/her presentation.

We have no way of reviving what the lecturer has told the students if not noted down at the same time. But with the speed of the lecturer, it might not be possible in some cases.

As a result, there is a higher chance that students forget what the lecturer said or taught the very next day because it wasn’t documented properly. There can be many reasons for a student to be absent for a lecture because of unavoidable circumstances such as medical trouble or family emergency.

A live streaming with recording sessions of a lecture would help all the students even if they were present in the lecture itself. The ability to revise what they have missed if the student attends the lecture late and the ability to repetitively go through a previous lecture before attending the next lecture would result in a huge academic improvement.

The main purpose of this system is to offer an effective way to help the students to access learning materials and information from anywhere and to quickly recap any forgotten or absent lectures via the earlier recordings of the sessions.

A system and a method for an interactive Internet-based video conferencing multicast operation which uses a video production studio with a live instructor giving lectures in real-time to the participating students. The video conference multicasting permits the students to interact with the instructor and other installations during the course of the lecture and to later browse the recorded session without a hassle.

1.3 Research Gap

There are a countless number of e-learning platforms which cover various aspects of learning such as video streaming, capturing the audience in the lecture hall if necessary, and screen sharing. But, the solution we propose is to handle advanced and enhanced features. Biometric recognition of participants to ensure authentication is proposed so that the attendance can be recorded. In addition to this, every lecture will be maintained as an mp4 video with the set of slides used, and the lecturer’s voice relevant to each slide, which provides an easy way of reference to the students who missed a particular lecture. Another important aspect considered in this research is the way of interaction between the lecturer and the students who have any doubts to be cleared during the lecture. For this purpose, when the students who are physically present in the lecture hall are concerned, a gesture-based system is proposed. Here, when the lecturer notices a specific student with the gesture of asking a question, the lecturer will have to perform a gesture for the camera to turn towards the audience and focus on the specific student who has the doubt by once again detecting the gesture performed by the student. When a remotely logged in user has a question for the lecturer, he/she has to signal the lecturer using a specific command, and then the lecturer has to decide whether to give video and/or audio control over to the specific user. As real-time video streaming consumes a lot of quota and bandwidth, the system has to intelligently manage data usage by ensuring the best possible transmission rate with minimum data consumption.

Taking the above facts into consideration, we propose a smart, cloud-based e-learning platform with advanced and flexible real-time video streaming, while ensuring interactivity between the lecturer and the students.

1.4 Research Questions

Lecture Capturing System is not another web based e-learning applications. It contains unique features which addresses the questions that are there in current systems and uniquely bundle together a well performing e-learning system. Below are some of the main problems that lead to the development of Lecture Capturing System.

To a student who is enrolled in a lecture

- No way of knowing what has happened on a day if he/she unable to attend the lecture physically.
- If the lecture room is too crowded not being able to see the whiteboard or the lecturer clearly.
- No way to revive what was the last lesson the lecturer did if they have not taken down notes.
- No way to view what lecturer did using his computer later on. (eg: typing a partial code, annotating a PowerPoint slide)

- Not being able to interact with the lecturer to solve a question if the lecture room is crowded.

To a lecturer who is teaching a course

- Unable to do a lecture from home and upload it to a server so that all the students can watch it.
- Tracking the real attendance of the students is not fully accurate as friends of the students mark the attendance for absent people.
- Sometimes sharing computer screen reveals all the files in the computer as well as open programs and browser tabs. There is a privacy issue.

1.5 Research Objective

1.5.1 Main Objective

The main objective of this research is to implement a cloud-based smart e-learning platform for university students, and lecturers to facilitate lecture delivering and participating in lectures through real-time video streaming while ensuring that, the maximum possible level of interaction between parties is achieved using image processing and machine learning techniques.

1.5.2 Specific Objectives

1. To track the lecturer’s movements using PTZ cameras to ensure students get a clear picture of what is happening in the classroom.
    - Connect to IP Camera and get the main footage.
    - Detect and Track the movement of the lecturer.
    - Move Camera according to the lecturer’s movement.

2. To intelligently share what the lecture wants to share with the students from lecturer’s laptop screen.
    - Share and Stream live screen of the laptop.
    - Share either screen only or webcam with screen or just webcam footage.
    - Ability to share complete screen or specific application screen only.

3. To provide an interface to handle and manage multiple courses.
    - Add, edit, delete and update courses to main system to support multiple course management.

2. BODY OF THE REPORT

2.1 Addressing the Literature

In recent years, there have been number of research efforts done to address the needs a smart e-learning management system. Below are some of the software functionalities and technologies that has been done prior to our research. Undertaking a Literature Survey helps us on finding and come up with the followings.

Regardless of the enormous growth of e-learning (electronic learning) in education and its perceived benefits, the efficiency of such e-learning systems will not be fully utilized if the students are not inclined to accept and use the system.

As a result, successful implementation of e-learning tools depends on whether the students are willing to adopt and accept the technology. Thus, it has become imperative for e-learning system developers to understand the factors affecting the user acceptance of web-based learning systems in order to enrich the students' learning experience and to create a better product to fulfill the necessary requirements of the student.

"Use of E-Learning", a research was done to find University students' purpose to use e-learning [1]. In this research, Teknologi Malaysia University's students try to apply and use the theory of technology acceptance model (TAM). They have employed structural equation modeling (SEM) approach with a SmartPLS software to investigate students' adoption process. Discoveries indicate that the content of e-learning and self-efficacy have a positive impact and substantially associated with perceived usefulness and student satisfaction, which impact university students' purpose to use e-learning. Although e-learning has expanded acceptance in universities around the world, the study of the intention to use e-learning is still essentially unexplored in Malaysia. The developed model is employed to explain the university student's intention to use e-learning. The study concludes that university students in Malaysia have positive perceptions towards e-learning and intend to practice it for educational purposes.

E-Learning is reflected as an innovative approach to education delivery via electronic forms of information. Multiple researches have been done to find the best way to use the technology and to better fit the students' necessities [1], [2]. The main obstacles that need to be addressed are the insufficient financial support, inadequate training programs, lack of ICT infrastructure, equivocal policies and objectives, and lack of awareness, interest, and motivation toward e-learning technology are considered as the main obstacles to enhance e-learning in Iraqi universities. The lack of training programs and inadequate ICT infrastructure are considered as the key issues which obstruct advancing of the e-learning process in Iraq [2].

Online body tracking by a PTZ camera has been done before to automatically track a single person and focus on that person [3], [4]. Online human body tracking method by an IP PTZ camera based on fuzzy-feature scoring was done. At every frame, candidate targets are detected by extracting moving targets using optical flow, a sampling, and appearance. The target is determined among samples using a fuzzy classifier. Results show that the system has a good target detection precision (> 88%), and the target is almost always localized within 1/4th of the image diagonal from the image center [3]. Autonomous lecture recording with a PTZ camera [4]. This reaches the same viewing experience while watching lectures recorded by an automated system. To accomplish this, they have developed an automatic cameraman (PTZ camera-unit) that is able to, Detect and track a single person/lecturer, Change between different types of shots, Listen to high-level instructions from a virtual or human director, and Take cinematographic rules into account

By tracking the lecturer, he is framed well in the picture at any moment and viewers can't be distracted. Takes cinematographic rules into account, which ensures that the viewer remains focused and the viewing experience is aesthetically more interesting. The action axis is determined by calculating the direction of movement and the gaze orientation. A PID control loop ensures smooth movement of the camera. Because of the speed of the algorithm, it will be easy to downscale for embedded hardware and still perform the calculations real-time.

Remote controlling of the PTZ camera system for lecture rooms [5]. This consist of a simple and inexpensive software solution for remote management of PTZ camera systems. This provides the ability for users to remotely control the PTZ camera system from one place with the simultaneous image capturing ability. Users of this application are able to choose a number of presets and this functionality is provided programmatically by sending queries to CCTV system, which then responds back to the controller. All of the responses are processed immediately into the desired form and stored in a selected row in SQLite database. This method of data storage doesn´t require the installation of SQL database, which makes the solution easier to apply. The program was created using the programming language C#. But this software solution does not support real-time tracking of a person, just several predefined presets so that feature can be improved to real-time operation in Lecture Capturing System. OpenTrack - Automated Camera Control for Lecture Recordings [6] records lecture sessions automatically without the need for a human camera person. A Tabletop Lecture Recording System [7]. This research presents a lecture recording system that employs gestures and digital cameras to facilitate remote distance teaching. Virtual Cameraman [8] uses two PTZ cameras having different utilities. One is named full-shot PTZ camera and the other is movement PTZ camera. To get camera movement information, the system first obtains continuous images from full-shot PTZ camera and theSpn extracts four fuzzified movement features which can represent four characters of audiences' motions respectively. On the other hand, an automatic camera movement model (ACMM) is constructed by recording photographers' habit of CM styles and shot types. The proposed system can select suitable CM styles and shot types by inputting the fuzzified motion features into the ACMM. After that, the system chooses the main target in the input frames obtained from the full-shot PTZ camera by using five aesthetic criteria. Finally, the system operates the movement PTZ camera to finish recording.

Real-time tracking of a non-rigid target with a moving pan-tilt-zoom (PTZ) camera. The tracking of the object and control of the camera is handled by one computer in real time. The main contribution of the paper [9] is method for target representation, localization and detection, which takes into account both foreground and background properties, and is more discriminative than the common color histogram based back-projection. A Bayesian hypothesis test is used to decide whether each pixel is occupied by the target or not. The target representation is suitable for use with a Continuously Adaptive Mean Shift (CAMSHIFT) tracker. Experiments show that this leads to a tracking system that is efficient and accurate enough to guide a PTZ camera to follow a moving target in real time, despite the presence of background clutter and partial occlusion.

Human tracking applications with a Global Positioning System (GPS) can get the user's position in real time on the open area. But if the user goes into the room or building, then the human tracking application cannot get the user's position as the GPS signal cannot get through the wall. By using RFID, human tracking application can perform tracking in a certain closed area by providing room's or building's information that is entered. The IP camera as part of the system will send the images as visualization inside the room. In this research, human tracking system is built on a specific area that has lots of room or building such as a theme park or sports club. System is designed using an RFID reader, RFID tags, IP Camera, the database server and Android smartphones. The research was done inside Syahdan Campus, Bina Nusantara University – Jakarta. The result shows that the application is working with 100% tapping and mapping accuracy [10].

IP-based physical security products like IP network camera (IPNC) are becoming essential in construction of a modern security system. In order to guarantee interoperability among them, ONVIF has been a de facto standard communication framework. In addition to core specification, ONVIF defines many services. ONVIF Event service is supposed to provide notification messages to registered clients when events happen, which is an essential mechanism to be support to make IPNC intelligent. In this paper, we report our efforts to implement ONVIF Event service for the smart IPNC. First, we design S/W architecture, necessary data structures, and workflow of ONVIF Event service according to ONVIF Event service specification. Then, we implement the design about ONVIF Event service by extending TI’s IPNC reference RDK S/W package [11]. Testing via an Open source ONVIF client verifies our implementation works properly.

Tracking people or moving objects across a PTZ camera and maintaining a track within a camera is a challenging task in applications of video surveillance. In this paper we propose a novel object positioning tracking framework, PTZ camera based position tracking system, which is also known as PCTS, can help estimate the position of an object by using camera parameters, pan and tilt. From object motion vector, the relevant information such as time, background and geographic parameters can be recorded in database as well [12]. In the experiment, the change of a person' position is recorded by the PTZ camera in real time. The PCTS provides a feasible solution for position analysis and security surveillance services in future.

Real-time person tracking has been implemented before, but not quite as what we have planned on doing; real-time broadcasting of the footage without any delay. The complete package of having the lecture capturing along with audience if necessary, screen sharing, Face Recognition based remote login and attendance marking for online participants, viewing the lecture in real-time with added features such as reading what the lecturer has told and intelligently generating chapters on the video according to the lecture slides played alongside with this makes Lecture Capturing System a perfect complete package of e-learning. A thorough research related to e-learning systems has led to the identification of some of the most influential factors used in the field of information systems research. More specifically, characteristics as well as the limitations, weaknesses, and strengths of web-based learning systems. Student variables, such as technical issues and adapting to the new ways are important variables that influence student learning, especially in a collaborative e-learning environment. Understanding these variables is now helpful for developers to design eloquent educational activities to promote student knowledge construction and make learning more effective and appealing. In particular, this research helps to better understand the characteristics of students and to comprehend what the students expect from the learning management systems. This can help the developers achieve the most effective deployment of such systems and also helps them improve their strategic decision making about technology in the future, they can decide on the best approach that fit their students before implementing any new technology.

2.2 Methodology

Figure 1 shows how the system process the feed from the IP Camera to correctly recognize the lecturer and track the movement of the lecturer and turn the camera accordingly.

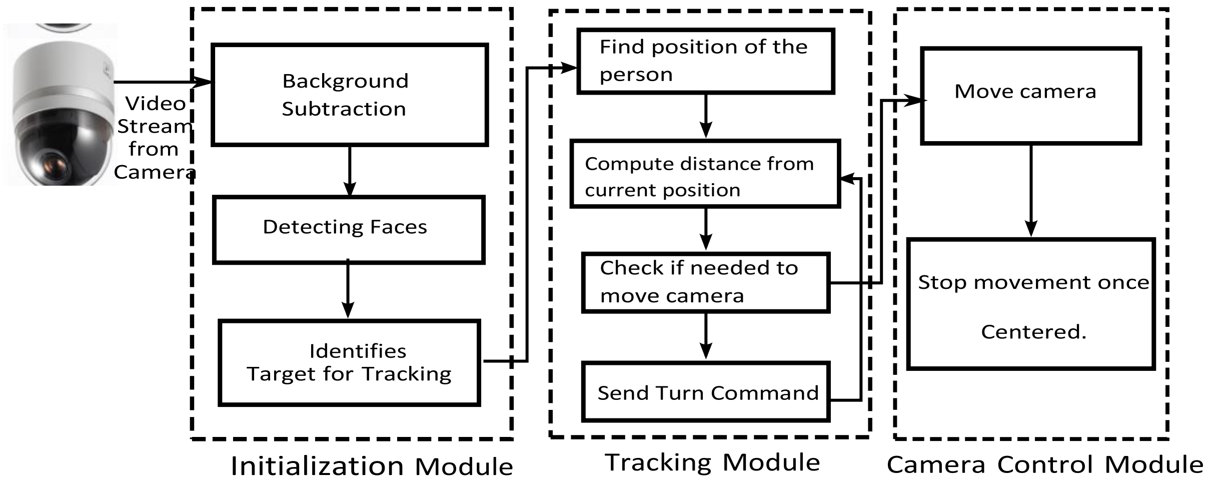

Figure 1 Flow of information, and processing of data for tracking.

Granting multiple platforms support live streaming of a video, what makes Lecture Capturing System unique is its ability to automatically focusing on the lecturer in real-time, lecturer is framed well in the picture at any moment and viewers can't be distracted. Takes cinematographic rules into account, which ensures that the viewer remains focused and the viewing experience is aesthetically more interesting and gives the viewers the best viewing angle all the time without a need for a separate cameraman.

An IP camera will be used to track the lecturer's movement and gestures in front of the camera and produce the necessary PTZ signals to pan, tilt and zoom accordingly thus ensuring that the lecturer's actions are always recorded without missing any detail. This video recording will be immediately compressed 'on -the-fly' in order to reduce its file size, then streamed live and also saved in the database for backup purposes and viewing later. Therefore, the students have the choice of attending the lecture via the live stream or listening to the lecture later. This is very beneficial to students since they can also attend lectures without being physically present (remotely) at the lecture.

2.2.1 Camera Movement with Detection

OpenCV Object Detection using Haar feature-based cascade classifiers is an effective object detection. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images [13].

Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the Figure 2 are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.

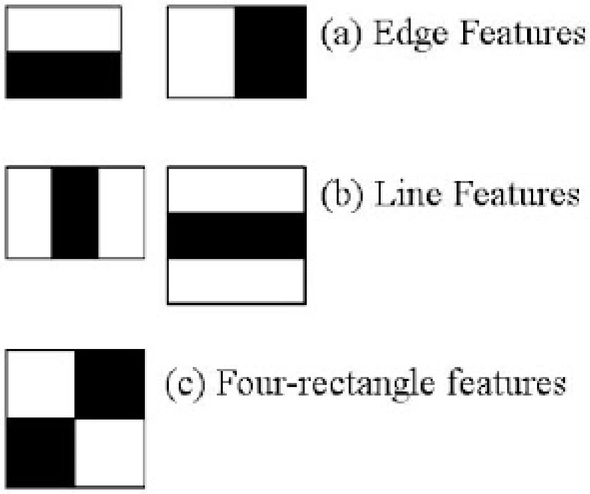

Figure 2 Haar features

Now, all possible sizes and locations of each kernel are used to calculate lots of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find the sum of the pixels under white and black rectangles. To solve this, they introduced the integral image. However large your image, it reduces the calculations for a given pixel to an operation involving just four pixels.

But among all these features we calculated, most of them are irrelevant. For example, consider Figure 3. The top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applied to cheeks or any other place is irrelevant.

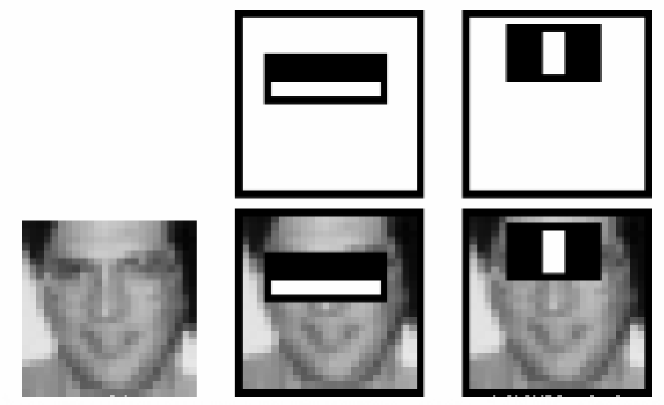

Figure 3 Features Calculated

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. Obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that most accurately classify the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then the same process is done. New error rates are calculated. Also new weights. The process is continued until the required accuracy or error rate is achieved or the required number of features are found).

The final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features. (Imagine a reduction from 160000+ features to 6000 features. That is a big gain).

So now we take an image. Take each 24x24 window. Apply 6000 features to it. Check if it is face or not.

In an image, most of the image is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot, and don't process it again. Instead, focus on regions where there can be a face. This way, we spend more time checking possible face regions.

For this they introduced the concept of **Cascade of Classifiers**. Instead of applying all 6000 features on a window, the features are grouped into different stages of classifiers and applied one-by-one. (Normally the first few stages will contain very many fewer features). If a window fails the first stage, discard it. We don't consider the remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region.

Haar-cascade Detection in OpenCV

OpenCV already contains many pre-trained classifiers for face, eyes, smiles, etc. Those XML files are stored in the opencv/data/haarcascades/ folder. For this detecting lecture purpose we have used Facial Detection Haar cascade [14].

First we need to load the required XML classifiers. Then load our input video stream from IP Camera in grayscale mode.

Now we find the faces in the image. If faces are found, it returns the positions of detected faces as Rect(x,y,w,h). Once we get these locations, we can calculate if the camera needs to turn to make sure the lecturer is detected in the middle.

This lecture tracking script uses OpenCV’s Haar Cascade Classifier to do the person detection task. First it initializes a face cascade using the frontal face Haar cascade provided with the OpenCV library. Then it starts to detect And Track the Largest Face it can find, if not tracking Face or lost the tracked face again it uses Haar cascade detector to detect face and then correlation tracker to follow it using dlib.

Both methods require to scan each the whole frame with a sliding window. The algorithm then tries to find the features of a person in each window position. These methods are too expensive to perform in each frame if we want to run our person tracker on restricted hardware like a budget laptop.

For this reason we combine the person detector with a correlation tracker. The correlation tracker expects a region of interest and starts tracking the pixels inside that region. In subsequent frames it tries to find where the pixels have most likely moved.

This is much faster and more robust than trying to find the person in each and every frame again.

```
def detectAndTrackLargestFace():
    #Open the ip ptz camera
    #capture = cv2.VideoCapture('rtsp://admin:@192.168.1.110:554/1/h264major latency=0')
    capture = cv2.VideoCapture('rtsp://192.168.1.110:554/1/h264major')
    #if need to use web-cam use the 0 option
    #capture = cv2.VideoCapture(0)

    #Create two opencv named windows
    cv2.namedWindow("base-image", cv2.WINDOW_AUTOSIZE)
    cv2.namedWindow("result-image", cv2.WINDOW_AUTOSIZE)

    #Position the windows next to eachother
    cv2.moveWindow("base-image",0,100)
    cv2.moveWindow("result-image",400,100)

    #Start the window thread for the two windows we are using
    cv2.startWindowThread()

    #Create the tracker we will use
    tracker = dlib.correlation_tracker()

    #The variable we use to keep track of the fact whether we are
    #currently using the dlib tracker
    trackingFace = 0

    #The color of the rectangle we draw around the tracking object face or upper body
    rectangleColor = (78,0,137)
```

Figure 4 Detecting largest face in the view to lock in for tracking

```python
#Result image is the image we will show the user, which is a
#combination of the original image from the ip-cam / webcam and the
#overlayed rectangle for the largest face
resultImage = baseImage.copy()


#If we are not tracking a face, then try to detect one
if not trackingFace:

    #For the face detection, we need to make use of a gray
    #colored image so we will convert the baseImage to a
    #gray-based image
    gray = cv2.cvtColor(baseImage, cv2.COLOR_BGR2GRAY)
    #Now use the haar cascade detector to find all faces
    #in the image
    faces = faceCascade.detectMultiScale(gray, 1.3, 5)

    #print("Using the cascade detector to detect face")


    #we are only interested in the 'largest'
    #face in order to detect lecturer, and we determine
    #this based on the largest
    #area of the found rectangle. First initialize the
    #required variables to 0
    maxArea = 0
    x = 0
    y = 0
    w = 0
    h = 0
```

Figure 5 If not tracking face trying to detect face of the lecturer

```python
max_error = 5
vm_cd = int(t_x)
if current_x<180 and foucesd:
    rot_cen = (180-current_x)*22
    turn_left_ptz = execute_js('onvif_movement/turn_left.js',str(rot_cen))
    print(rot_cen)
    if turn_left_ptz:
        print('Focused Success - Turned Left')
        foucesd = False
    else:
        print('failed left')
if current_x>180 and foucesd:
    rot_cen = (current_x-180)*22
    turn_right_ptz = execute_js('onvif_movement/turn_right.js ',str(rot_cen))
    print(rot_cen)
    if turn_right_ptz:
        print('Focused Success - Turned Right')
        foucesd = False
    else:
        print('failed right')
```

Figure 6 Move the camera to center the lecturer

2.2.2 Person Detection - Secondary

Uses YOLO v2 and Deep Sort with TensorFlow to track the lecturer in the stage. Uses a pre trained model to detect humans [15].

Uses NVIDIA CUDA to use TensorFlow gpu which is extremely fast than using the cpu version.

You only look once (YOLO) is a state-of-the-art, real-time object detection system. On a Pascal Titan X it processes images at 30 FPS.

```python
# Definition of the parameters
max_cosine_distance = 0.3
nn_budget = None
nms_max_overlap = 1.0

# deep_sort
model_filename = 'model_data/mars-small128.pb'
encoder = gdet.create_box_encoder(model_filename,batch_size=1)

metric = nn_matching.NearestNeighborDistanceMetric("cosine", max_cosine_distance, nn_budget)
tracker = Tracker(metric)
```

Figure 7 Load Person Model Data

```python
while True:
    ret, frame = video_capture.read()  # frame shape 640*480*3
    if ret != True:
        break;
    t1 = time.time()

    image = Image.fromarray(frame)
    boxs = yolo.detect_image(image)
   # print("box_num",len(boxs))
    features = encoder(frame,boxs)

    # score to 1.0 here).
    detections = [Detection(bbox, 1.0, feature) for bbox, feature in zip(boxs, features)]

    # Run non-maxima suppression.
    boxes = np.array([d.tlwh for d in detections])
    scores = np.array([d.confidence for d in detections])
    indices = preprocessing.non_max_suppression(boxes, nms_max_overlap, scores)
    detections = [detections[i] for i in indices]

    # Call the tracker
    tracker.predict()
    tracker.update(detections)

    for track in tracker.tracks:
        if track.is_confirmed() and track.time_since_update >1 :
            continue
        bbox = track.to_tlbr()
        cv2.rectangle(frame, (int(bbox[0]), int(bbox[1])), (int(bbox[2]), int(bbox[3])),(255,255,255), 2)
        cv2.putText(frame, str(track.track_id),(int(bbox[0]), int(bbox[1])),0, 5e-3 * 200, (0,255,0),2)
        print("Live Tracking - " + str(bbox[0]) + ' ' +str(bbox[1]))
        #print("Live Tracking2 - " + str(bbox[2]) + ' ' +str(bbox[3]))
        if track.track_id > 1:
          print("Too Many People in Stage " + str(track.track_id))


    for det in detections:
        bbox = det.to_tlbr()
        cv2.rectangle(frame,(int(bbox[0]), int(bbox[1])), (int(bbox[2]), int(bbox[3])),(255,0,0), 2)
        print("Detected in " + str(bbox[0]))
```

Figure 8 Detect and Track Lecturer

2.2.3 PTZ Camera Movement

ONVIF stands for **Open Network Video Interface Forum.** It's an open industry standard that provides interoperability among IP security devices such as security cameras, video recorders, software, and access control systems [16].

Since we are using ONVIF protocols to move the camera it allows the compatibility from different vendor's devices so LCS will have the support by most IP based security devices manufacturers giving it an added benefit of not limiting the system to a specific brand of IP Cameras.

The node-onvif is a Node.js module which allows you to communicate with the network camera which supports the ONVIF specifications.

The ONVIF (Open Network Video Interface) is an open industry forum promoting and developing global standards for interfaces of IP-based physical security products such as network cameras. The ONVIF specifications are available in their web site.

Recently, most of network cameras for business support the ONVIF standard. Furthermore, some network cameras for home support it though the implementation is partial. The node-onvif allows you to control network cameras which implement the ONVIF standard.

With this there are separate commands to move the camera movement left, right, up or down at selected speeds and can freeze movement according to the requirements.

```
new Cam({
    hostname : HOSTNAME,
    username : USERNAME,
    password : PASSWORD,
    port : PORT,
    timeout : 10000
}, function CamFunc(err) {
    if (err) {
        console.log(err);
        return;
    }

    var cam_obj = this;
    var stop_timer;
    var ignore_keypress = false;
    var preset_names = [];
    var preset_tokens = [];

    cam_obj.getStreamUri({
            protocol : 'RTSP'
        }, // Completion callback function
        // This callback is executed once we have a StreamUri
        function (err, stream, xml) {
            if (err) {
                console.log(err);
                return;
            }
```

Figure 9 Get Stream URL of the IP Camera

```
// Import libraries
const express = require('express');
const router = express.Router();
const bodyParser = require('body-parser');

// Import configuration file
const config = require('../configurations/config');

router.use(bodyParser.json());

let HOSTNAME = '192.168.1.110',
    PORT = 8999,
    USERNAME = 'admin',
    PASSWORD = '',
    STOP_DELAY_MS = 500;

let Cam = require('../lib/onvif').Cam;
```

Figure 10 Connect to IP Camera

```
function move(x_speed, y_speed, zoom_speed, msg) {
    // Step 1 - Turn off the keyboard processing (so key-presses do not buffer up)
    // Step 2 - Clear any existing 'stop' timeouts. We will re-schedule a new 'stop' command in this function
    // Step 3 - Send the Pan/Tilt/Zoom 'move' command.
    // Step 4 - In the callback from the PTZ 'move' command we schedule the ONVIF Stop command to be executed after a short delay
    // and re-enable the keyboard

    // Pause keyboard processing
    ignore_keypress = true;

    // Clear any pending 'stop' commands
    if (stop_timer) clearTimeout(stop_timer);

    // Move the camera
    console.log('sending move command ' + msg);
    cam_obj.continuousMove({x : x_speed,
            y : y_speed,
            zoom : zoom_speed } ,
        // completion callback function
        function (err, stream, xml) {
            if (err) {
                console.log(err);
            } else {
                console.log('move command sent '+ msg);
                // schedule a Stop command to run in the future
                stop_timer = setTimeout(stop,STOP_DELAY_MS);
            }
            // Resume keyboard processing
            ignore_keypress = false;
        });
}
```

Figure 11 Camera Movement Function

```
function stop() {
    // send a stop command, stopping Pan/Tilt and stopping zoom
    console.log('sending stop command');
    cam_obj.stop({panTilt: true, zoom: true},
        function (err,stream, xml){
            if (err) {
                console.log(err);
            } else {
                console.log('stop command sent');
            }
        });
}
```

Figure 12 Stop Camera Movement

2.2.4 Easy Screen Share Browser Extension

With the ability to share the screen either completed or selected custom application window right from the web browser and start streaming it along the main live stream makes the lectures task at ease and more efficient. This is helpful because most of the times when the lecturer starts to plug his laptop to the main projector in a normal classroom all of his desktop content open tabs on web browser, everything is visible to the students, privacy is a concern and it’s a hassle to switch sharing the screen on and off all the time to the lecturer. With this Easy Screen Share feature lecturer can stream his webcam footage alongside with the shared screen if required (If in front of the laptop blocking the main camera view).

This extension simply initializes socket.io and configures it in a way that single audio/video/screen stream can be shared/relayed over users without any bandwidth/CPU usage issues. This uses RTCMultiConnection is a WebRTC library that is used for WebRTC streaming [17].

```
function captureDesktop() {
    if (connection && connection.attachStreams[0]) {
        setDefaults();

        connection && connection.attachStreams.forEach(function(stream) {
            stream.getTracks().forEach(function(track) {
                track.stop();
            });
        });

        chrome.storage.sync.set({
            enableTabCaptureAPI: 'false',
            enableMicrophone: 'false',
            enableCamera: 'false',
            enableScreen: 'false',
            isSharingOn: 'false',
            enableSpeakers: 'false'
        });
        return;
    }

    chrome.browserAction.setTitle({
        title: 'Capturing Desktop'
    });
```

Figure 13 Desktop Capture Initialization

2.2.5 Course Management

The administrator can manage courses conducted at the university. When a new course is introduced to the university, the administrator can add it to the system. If any changes are proposed to an existing course (e.g. change the number of credits in the course), the administrator is able to update its details. There would be occasions when an existing course's learning material gets outdated; in this case, the administrator can delete this course from the system.

2.3 Research Findings

2.3.1 Findings

The most striking feature of LCS is the automatic lecture detection and camera rotation accordingly. Even though there several e-learning video streaming systems none has approach an intelligent way of using such tracking with the most basic hardware and system performance requirements. This is purely achieved using from scratch using OpenCV Haar Cascade and Node ONVIF Protocol to control the camera.

In order to build this lecture tracking script, we use OpenCV's Haar Cascade Classifier to do the person detection task. First it initializes a face cascade using the frontal face Haar cascade provided with the OpenCV library. Then it starts to detect And Track the Largest Face it can find, if not tracking Face or lost the tracked face again it uses Haar cascade detector to detect face and then correlation tracker to follow it using dlib.

Both methods require to scan each the whole frame with a sliding window. The algorithm then tries to find the features of a person in each window position. These methods are too expensive to perform in each frame if we want to run our person tracker on restricted hardware like a budget laptop.

For this reason we combine the person detector with a correlation tracker. The correlation tracker expects a region of interest and starts tracking the pixels inside that region. In subsequent frames it tries to find where the pixels have most likely moved. This is much faster and more robust than trying to find the person in each and every frame again.

Below are the other tools and technologies used.

Tools

- Pycharm
- IntelliJ IDEA
- Robo 3T
- WebStorm
- ONVIF Device Manager
- OBS Studio
- Google Chrome Dev

Technologies

- React
- Node.JS
- MongoDB

- Tensorflow GPU
- OpenCV
- HTML
- JavaScript

2.3.2 Performance Requirements

It is expected that the proposed system will perform all the requirements stated under the functional requirements section. Some performance requirements identified is listed below:

- The system should be able to accommodate the traffic of streaming live.
- Dashboards should be refreshed in real time.
- This is a web application, so simultaneous access is required, an average response time for any action should not exceed more than 5 seconds.

2.3.3 Maintainability

The system may need regular corrective and preventive maintenance to make certain that any hindrances and limitations are removed at the earliest time of detection. This is importance to ensure a smoothly functioning system.

2.3.4 Other Requirements

Additional requirements in order to maintain consistency and global standards are mentioned below.

- Usage of IEEE standards for imagery and typography to maintain consistency [3].
- Usage of open-source software tools and technologies.
- Convenient navigation through the system.
- Positioning of buttons, tabs, labels, and other components.

2.4 Testing

After the code is developed it is tested against the requirements to make sure that the product actually solves the needs addressed and gathered during the requirements phase. During this phase, all types of functional testing and non-functional testing would be carried out.

Functional testing involves:

- Unit testing - Each individual unit will be tested by the respective team member to see if it functions correctly.
- Component testing - Similar to unit testing but with a higher level of integration. The big difference here is that the testing is done in the context of the application instead of just directly testing the method in question.
- Integration testing – This involves integrating modules one at a time until the whole system is set up. At the end of each iteration, the system is tested to ensure there are no errors during integration.
- System testing – After the entire system is set up, system testing is carried out to make sure all the components have been integrated successfully and the system is fully functional.

In non-functional testing, certain attributes of the system such as memory leaks, performance, and robustness of the system are verified. Some of these test phases are performance testing, security testing, load testing, and recovery testing.

In order to test our tracking of the lecturer, we use ONVIF Device Manager V2.2.25 provided by Synesis [15], which is an open source desktop application used to manage ONVIF IP devices.

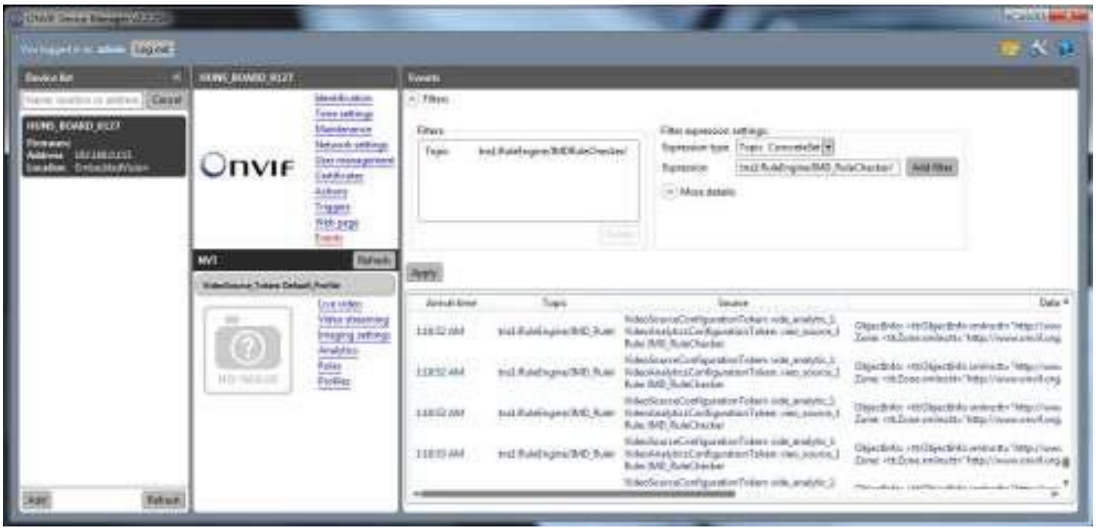

Figure 14 ONVIF Device Manager

3. RESULT AND DISCUSSION

3.1 Evidence

This section covers the results achieved from the research project and the new methodologies found to address further research in the undergraduate context.

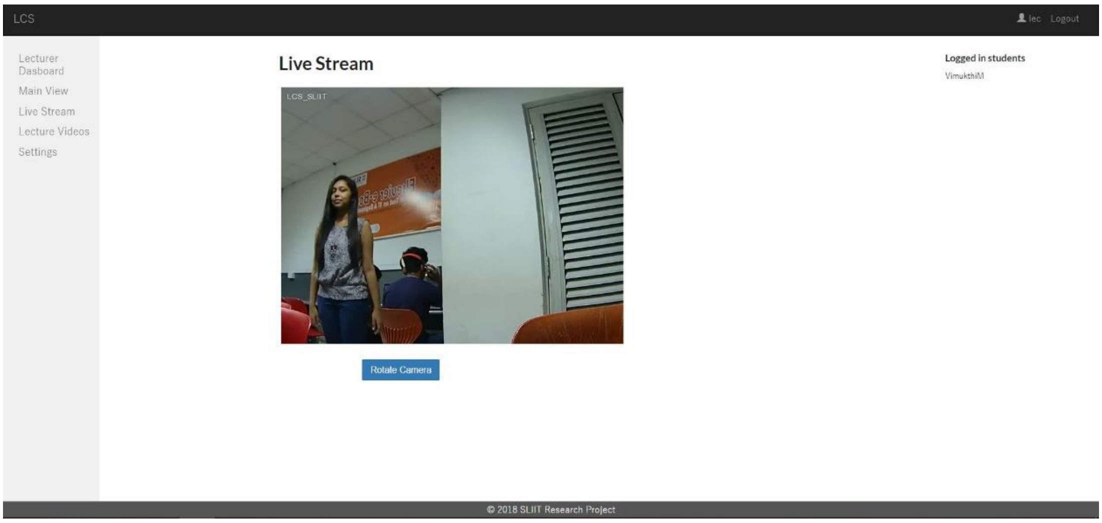

Figure 15 Live Stream Main View

LCS consist of a web application for the lecturer to log in and start live streaming process. Figure 12 illustrates the main view from the IP camera as in real-time. In figure 13 it shows that if the lecturer chooses he/she can disable the auto tracking features and control everything manually right from the web application if the need arises. This advanced controls view enables lecturer to take command of the camera and do the movement manually if required. Move camera to any direction and to zoom in and zoom out as required. Also lecturer has the ability to stop any scripts to disable automatic tracking of the lecturer.

Example: A special Question and Answers session with another colleague at a specific location or a scientific experiment on stage.

Figure 16 Advanced Controls UI

Figure 14 illustrates the main view of the Screen Sharing extension which enables lecturer to choose whether to share the screen or screen and webcam or just the webcam footage.

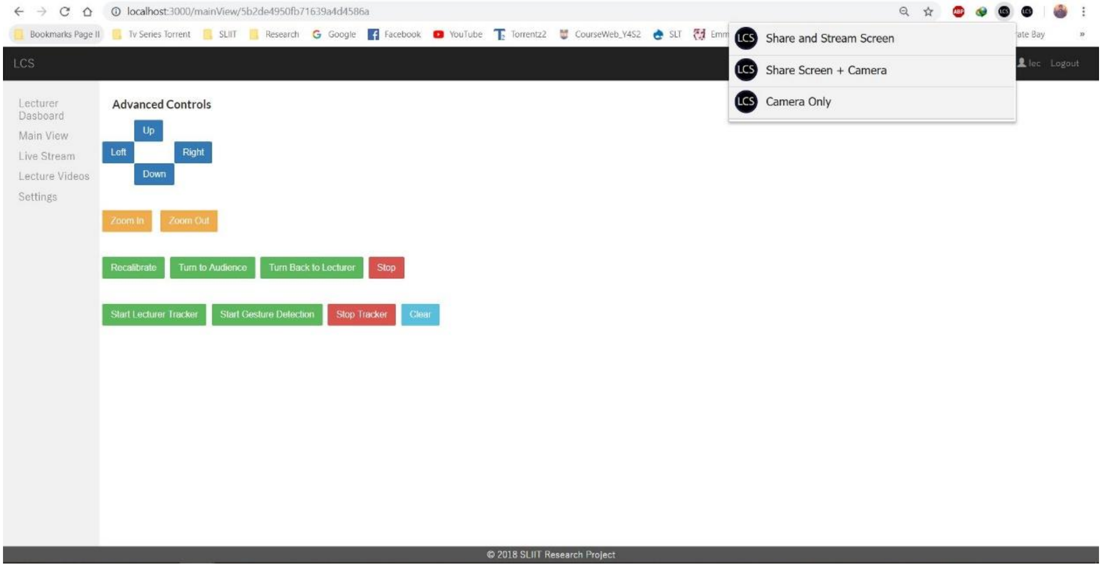

Figure 17 Main View of the extension

With Figure 15 view, lecturer should select the sharing screen, either full screen or one of the specific application window had he/she selected screen sharing. If the lecturer selected full screen sharing everything will be shared from the lecturer's laptop screen including taskbar and all open programs. If selected an Application Window only the selected application will stream to the students, this is really helpful if the lecturer is using his/her own laptop with private files and programs open in the background. Security and Privacy vice this gives lecturer a peace of mind.

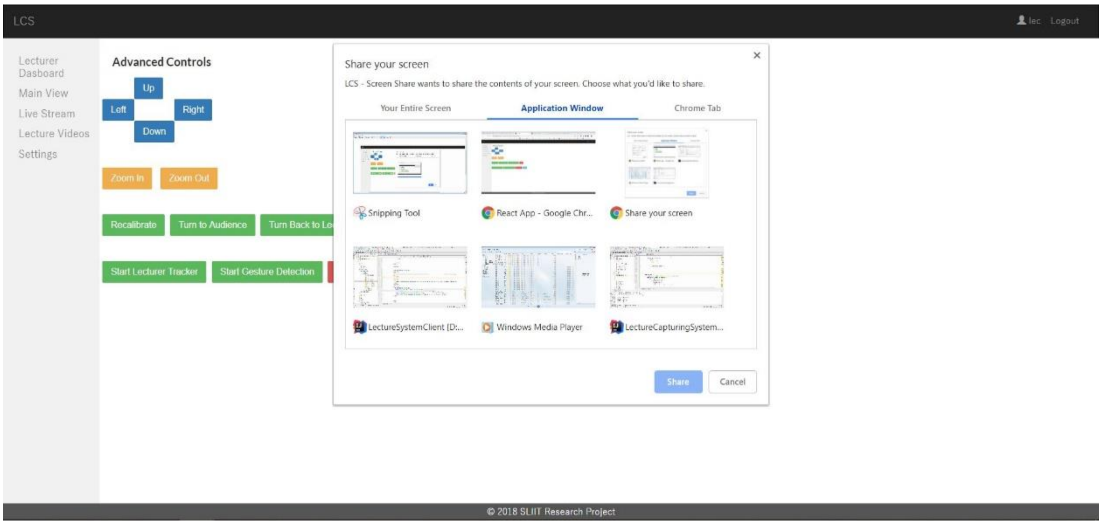

Figure 18 Share Screen Window

3.2 Discussion

Obviously, many methods have been introduced as e-learning platforms. Most of the approaches focus on just one aspect of the e-learning experience. For example most of the time the platforms only support to viewers to view the main static view of a stage remotely. But this research has experimented taking an entirely different path by replicating a class-room like experience for the remotely logged in students and even helps the students who participated get even more out from the lecture.

A live streaming with recording sessions of a lecture would help all the students even if they were present in the lecture itself. The ability to revise what they have missed if the student attends the lecture late and the ability to repetitively go through a previous lecture before attending the next lecture would result in a huge academic improvement.

The main purpose of this system is to offer an effective way to help the students to access learning materials and information from anywhere and to quickly recap any forgotten or absent lectures via the earlier recordings of the sessions.
A system and a method for an interactive Internet-based video conferencing multicast operation which uses a video production studio with a live instructor giving lectures in real-time to the participating students. The video conference multicasting permits the students to interact with the instructor and other installations during the course of the lecture and to later browse the recorded session without a hassle.

Key Features of Lecture Capturing System is,

- Automatically focusing on the lecturer in real-time to give the viewers the best viewing angle all the time.
- Sharing lecturer’s computer screen with the students so that they can see what the lecturer is doing on his/her computer such as coding, annotating a PowerPoint slide or any media file the lecturer is going to show.
- Lecturer communicating with students either by voice/video on request of the student so that everyone can see the conversation and clear any doubts regarding the question of the student.
- Generating readable text content after the lecture by intelligently converting the lecturer’s voice into text is a helpful feature for students that are having difficulty hearing and it helps the students revise the lecture more efficiently.
- On the recorded offline video having thumbnails of the chapter makes it easier for students to watch the relevant part they needed without going through the whole video.
- Intelligent bandwidth management to make sure that we use the least possible data bandwidth to transfer the videos to the students.
- Capturing the audience and focusing on a guest speaker in real-time helps the remote logged in students to experience the live classroom environment.
- Biometric authentication of the remotely logged in users via facial recognition to act as a secondary layer of security and to mark the attendance of remote students.

LCS stands itself unique form existing product being as a comprehensive product that includes Live Stream with tracking, Screen Sharing, Smart Gesture Recognition, Biometric authentication with attendance, advanced thumbnail generator for recorded videos, lecture in text, multiple course and user management with advanced quota and bandwidth management mechanism all in one. Even though there are many e-learning applications in the market, there isn't a system which handles all the necessary requirements like LCS. Therefore LCS can be reckoned as one of the most effective systems when it comes to E-Learning systems.

4. CONCLUSION

In this research, while existing approaches to e-learning platforms were reviewed, new better way of e-learning with implementing a much organized solution is focused.
This report examines an innovative approach that is best suited to develop a lecture capturing system that provides a complete classroom experience and much more to remotely logged in students. This system stands itself unique from other existing products and being as a comprehensive product that includes biometric authentication, gesture detection, live streaming of lectures, automated attendance marking, offline recording of lectures, bandwidth management and desktop screen capturing all in one.

This research work has been developed mainly for addressing the problems in Sri Lankan universities, specifically addressing the lack of interactivity between the lecturer and the students. Even though this research focuses on universities, it definitely has the potential to be used in other fields such as business conferencing. In next stage, in one hand, research team will be focusing on improving the accuracy of the face recognition and gesture detection models by testing other algorithms effectively. Also, research team will focus on minimizing bandwidth costs by testing out bandwidth optimization techniques. It is hoped that for any person who expects to build a similar system or any other real-time system, results of this research will be an aid and will provide insight on the performance, accuracy and reliability level that can be expected with the combination of tools, technologies, programming approach considered in this paper.

LCS stands itself unique form existing product being as a comprehensive product that includes variety of features that make the remote classroom experience as possible as close to the real experience. Even though there are many e-learning applications in the market, there isn't a system which handles all the necessary requirements like LCS. Therefore LCS can be reckoned as one of the most effective systems when it comes to E-Learning systems.

5. REFERENCES

[1] W., *"Use of E-Learning," Universiti Teknologi*. Malaysia, Johor, Malaysia, 2018.

[2] M. A. Mahmod, "E-learning in Iraqi Universities," in *International Conference on Computing, Engineering, and Design (ICCED), Selangor*, Malaysia, 2017.

[3] P. D. Z. Varcheie, "Online Body Tracking by a PTZ Camera in IP Surveillance System," Department of Computer Engineering and Software Engineering, Station Centre-ville, Montr´eal, (Qu´ebec), Canada, 2009.

[4] T. G, "Dries Hulens, 'Autonomous lecture recording with a PTZ camera,'" in *Canadian Conference on Computer and Robot Vision*, Sint-Katelijne-Waver, Belgium, 2014.

[5] M. M. M. H. R. Jacko, "Remote control of the PTZ camera system for lecture rooms," *Dep. Comput. Inform*, 2015.

[6] B. Wulff, "OpenTrack - Automated Camera Control for Lecture Recordings," *IEEE Int. Symp. Multimed*, 2011.

[7] C.-F. C. a. P.-C. S. Yong-Quan CHEN, "A Tabletop Lecture Recording System," in *International Conference on Consumer Electronics-Taiwan*, Taiwan, 2015.

[8] Y.-T. T. S. C. a. S.-W. C, "Chiung-Yao Fang, 'Chiung-Yao Fang, You-Ting Tsai, Shuan Chu, and Sei-Wang Chen,'" Department of Computer Science and Information Engineering, Taiwan, 2015.

[9] P. Kumar and T. S. Sheng, "Real Time Target Tracking with Pan Tilt Zoom Camera," Digital Image Computing, Adelaide, 2009.

[10] D. P. Hutabarat, D. Patria, S. Budijono and R. Saleh, "Human Tracking Application in a Certain Closed Area Using RFID Sensors and IP," Information Tech, Jakarta, 2016.

[11] C. C. Phan, B. T. Nguyen and S.-T. Chung, "Design and Implementation of ONVIF-based Event Service for DM 814x Camera," IEEE, DaNang, 2015.

[12] C.-S. Yang, R.-H. Chen, C.-Y. Lee and S.-J. Lin, "PTZ Camera Based Position Tracking in IP - Surveillance System," IEEE, Tainan, 2008.

[13] "OpenCV library." [Online]. Available: https://opencv.org/. [Accessed: 11-Aug-2018].

[14] “OpenCV: Face Detection using Haar Cascades.” [Online]. Available: https://docs.opencv.org/3.4.2/d7/d8b/tutorial_py_face_detection.html. [Accessed: 11-Aug -2018].

[15] “YOLO: Real-Time Object Detection.” [Online]. Available: https://pjreddie.com/darknet/yolo/. [Accessed: 08- Aug -2018].

[16] “Onvif.” [Online]. Available: https://www.onvif.org/onvif/ver20/util/operationIndex.html. [Accessed: 07- Aug -2018].

[17] “WebRTC Home | WebRTC.” [Online]. Available: https://webrtc.org/. [Accessed: 11- Aug -2018].

[18] Synesis, ONVIF Device Manager (Onvifdm), April 2015. [Online].

Available: http://sourceforge.net/projects/onvifdm