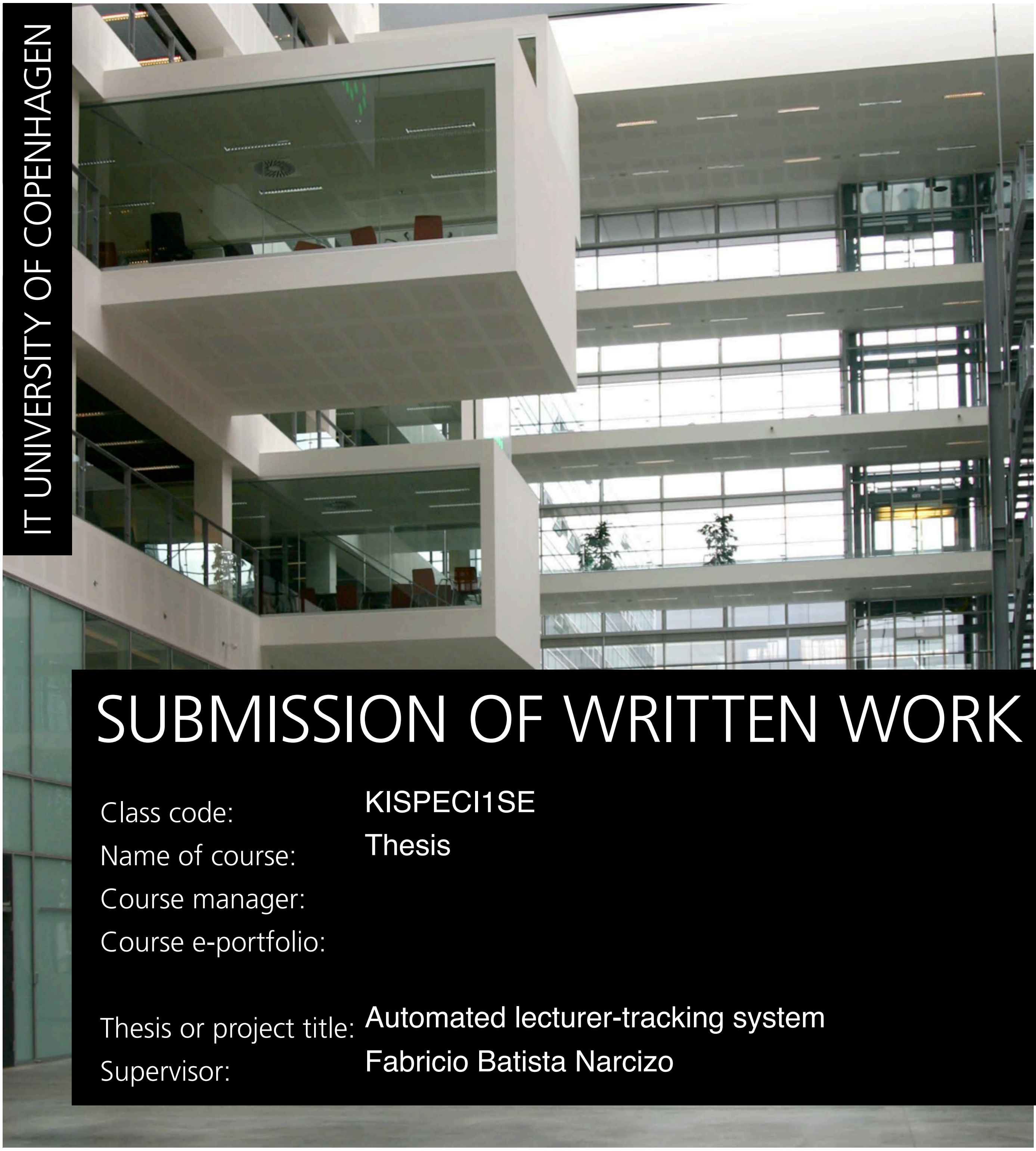
IT UNIVERSITY OF COPENHAGEN



SUBMISSION OF WRITTEN WORK

Class code: KISPECI1SE

Name of course: Thesis

Course manager:

Course e-portfolio:

Thesis or project title: Automated lecturer-tracking system

Supervisor: Fabricio Batista Narcizo

	Full Name:	Birthdate (dd/mm-yyyy):	E-mail:
1.	Andrej Balas	21/05-1992	bala@itu.dk
2.			@itu.dk
3.			@itu.dk
4.			@itu.dk
5.			@itu.dk
6.			@itu.dk
7.			@itu.dk

IT UNIVERSITY OF COPENHAGEN

Master Thesis

Automated Lecturer-Tracking System

Author: Andrej Balas

Supervisor: Fabricio Batista Narcizo, Ph.D.

IT University of Copenhagen
Copenhagen, Denmark

A thesis submitted in fulfillment of the requirements for the degree of Master of Science.

September 2018

Abstract

Development of technology has brought some significant changes to the educational system, resulting in some new means of gathering the knowledge. In this project, we focus on the video lectures that provide significant benefits for all the students. We aim to enhance the way video lessons are recorded by introducing the system for automatic lecturer-tracking.

This thesis introduces the new approach in implementing an automated lecturer-tracking system by using a smartphone as the replacement for a camera device and a processing unit. The proposed solution uses the YOLO real-time object detection system and tracking algorithms from iOS Vision framework to detect and track the lecturer. A motorized pan-tilt head rotates the smartphone based on the input the smartphone sends to it. Experimental results show that the system can perform the desired behavior of lecturer-tracking, eliminating the need for human help in the process of recording.

Keywords: Lecturer-tracking, YOLO, Object detection, Object tracking, Pan-tilt head, Arduino, Bluetooth

Contents

List of Figures

1

Introduction

We live in a time of a vast development of technology. As a result, technology embeds into all different areas of our lives. “Smart” devices everywhere make everyday tasks automated, and we start leaning on them, eventually becoming unaware of their presence. It is impossible to avoid such a firmly integrated paradigm, and why should we, when it is more than obvious that we only benefit from it.

1.1 Background

The effects of technology on education are so evident that nowadays we cannot imagine studying without the help of computers and the Internet. One of the main purposes of a forerunner of today’s Internet - the ARPANET - was to share computer resources, and to encourage the collaboration between the researchers on different locations [1]. The specific act of sharing the resources and knowledge over the network of computers leads us to conclude that the very first designers of the Internet made it so it can facilitate education.

The type of digital media that served for spreading the knowledge emerged from simple text files to the entire online universities. The benefits of using digital resources are so big that they induced the whole educational system to readjust according to these new forms of resources.

As the most common way of educating students at universities is through classes, it is natural that their digital forms are video records or streaming. Woolfitt [2] describes the significance and importance of videos in today's educational systems, stating that it is quite likely that this will become more standard over time.

The history of using videos for educational purposes began during the World War II when filmstrips were first studied during as a training tool for soldiers [3]. Cruse [3] also states that "*educators have recognized the power of audio-visual materials to capture the attention of learners, increase their motivation and enhance their learning experience*".

To understand why videos have such an important role in education, we need to take a deeper look into the benefits they provide over traditional educational resources such as classes:

- watch the videos at any time and from any place
- pause, review, slow down, skip and skim through the content [2]
- interact with the lecturer and other students over comments section usually available on platforms for video reproduction
- browse all the previous discussions on the topic
- explore other videos that have the same subject
- lectures can be translated into any language
- lecturers do not have to repeat the classes and can focus on other activities such as exercises or research

Every one of this benefits contributes to the better understanding of the lecture's content which implies that the students who use this approach have an opportunity to gain more knowledge in a better and easier way. It is apparent that students can learn at their own pace and choose the videos that better match their preferences.

Despite the benefits of recording lectures, Cavallaro et al. [4] states that "*because these solutions require the presence of one cameraman for each lecture room, they are not economically viable for many universities.*".

1.2 Objectives

To avoid paying the cameraman for every lecture, an automated recording system should be used instead, where the system itself can recognize and track the lecturer as they move around the classroom. Some ready-to-use products on the market have exactly the same purpose [5, 6]. Since those products are made out of many expensive parts such as 3D sensors or advanced cameras, their price usually goes up to a few thousand dollars.

There are several implementations of such systems [4, 7, 8] that use PTZ (pan-tilt-zoom) camera together with an image processing module. Based on the analysis, a micro-controller sends the signals for pan, tilt and zoom movements of the camera, so it could continue tracking the lecturer. There is also an implementation that next to the camera and the pan-tilt head uses Kinect for 3D human tracking [9].

Research question: How can we improve the performance and the design of automated lecturer-tracking systems concerning the system requirements simplicity and cost?

1.3 Methodology

The idea behind this thesis is to simplify and yet improve the performance of an automated lecture recording system by moving the heart of it to something we already possess - smartphones. The reason smartphones are the ideal piece of technology to use when it comes to recording and image processing is that they have better and better image sensors every year, with a typical smartphone having a resolution with more than ten megapixels [10]. Besides, more and more smartphones come with two integrated cameras, with some having even three of them (Huawei P20 Pro), all of them placed on the back of the device. Such a hardware configuration let us create amazing shots anywhere we go, including the lectures. The sudden increase in the use of machine learning caused manufacturers to implement the new type of processors to smartphones, such as Vision Processing Units (VPUs) in order to run advanced machine learning (ML) algorithms while lowering the energy requirements. Thanks to such a development of smartphones hardware, we can perform advanced video processing task such as *object recognition and tracking* in real time, making our lecturer-tracking system capable of streaming the lectures.

The primary mechanism that would allow a smartphone to move is imagined to be a pan-tilt head, connected to a smartphone over a micro-controller with a Bluetooth module. A smartphone would then send the signals to the head, based on a position of the lecturer in a video frame, without the previous need of the system calibration between the camera and the head.

Why is this system better than ones currently available on the market? For starters, it is easy to find the main hardware requirement - a smartphone - in the pocket of almost every student, and for the implementation of the leading software requirement, we only need to click one button in the mobile-applications stores. Also, one significant advantage is that smartphones are only going to improve over time, which means that by regular switching to better models of smartphones, our automated lecturer-tracking system also improves, which is not the case with other systems.

However, the functional criteria has been set high by the professional systems manufacturers, managing to resist some extra situations that can occur in the classroom, for example tracking the lecturer even if they turn around facing the blackboard, or if other people in the classroom (e.g., students), move in front or behind the lecturer while they are standing still.

Given the recent launches and upgrades of native platforms' frameworks for machine vision tasks, they can be used for accomplishing the desired functional performance of the system.

In chapter 2, we describe all the methods and technology that the system we propose uses to achieve the desired functionality - tracking of a lecturer.

In chapter 3, we present all the results we got from experimental tests of the main methods the system uses for tracking. Here, we also compare the proposed system to the other related systems.

In chapter 4, we propose several improvements that would make the proposed system ready for the real-world usage.

Finally, in chapter 5, we state our conclusion, and we describe the contribution the proposed system makes for the research field of lecturer-tracking systems.

2

System architecture

An automated lecturer-tracking system should be able to perform a “simple” process of rotating the camera towards the lecturer during a video recording of a lecture.

Based on this requirement, we can conclude that our system should include at least a camera, a processing unit, and a motorized rotating head on which we can fix the camera. The processing unit needs to control the motorized head, hence they have to be connected in some way.

To accomplish the desired functionality, our system has several hardware requirements: (1) a smartphone device - we use iPhone X because its operation system provides frameworks that facilitate object detection and tracking by exploiting the neural network hardware called “Neural engine”; (2) a microcontroller - Arduino Uno board is a perfect candidate for our system because it provides application programming interface, which is important for logical connection between the Bluetooth module and the pan-tilt head, but also provides an extensive hardware interface, which we need to physically connect the head and the module; (3) a Bluetooth module - the solution we propose uses Keyes Bluetooth 4.0 module since it supports Bluetooth Low Energy standard used to provide a wireless connection between the Bluetooth module and the smartphone; (4) a pan-tilt head - we decided to use

Maxwell MP-101 motorized pan-tilt head because it provides an interface for controlling the pan and tilt movements of the head over remote socket adjusted for 7-pin DIN cable.

Regarding behavioral structure, the proposed system consists of several functional units that exchange data needed for the overall functionality of the system. The smartphone captures and processes video input, determines the lecturer's position using object detection and tracking, makes a decision about necessary movements based on the lecturer's position, and sends the control signals to the microcontroller over Bluetooth module. Bluetooth module provides interface for both wireless communication with the smartphone and wired communication with the microcontroller. The microcontroller continually read the data from the Bluetooth module, generates the electric impulse, and sends it to the pan-tilt head. The pan-tilt head rotates based on the input it receives from the microcontroller.

Figure 2.1 shows the most important physical components of the proposed system and their behavioral characteristics.

2.1 Method

The biggest challenge of this project was to obtain the lecturer's position in each video frame. Given the available information that we gather through the process of video recording, and a smartphone as the device which acquires that information, it seemed most natural to use the *image analysis* approach for implementing the central algorithm of this process. We use image analysis because the state-of-the-art techniques are fast enough so that we can perform real-time extraction of information from the frames we capture. We need a real-time behavior of the system because the pan-tilt head movements

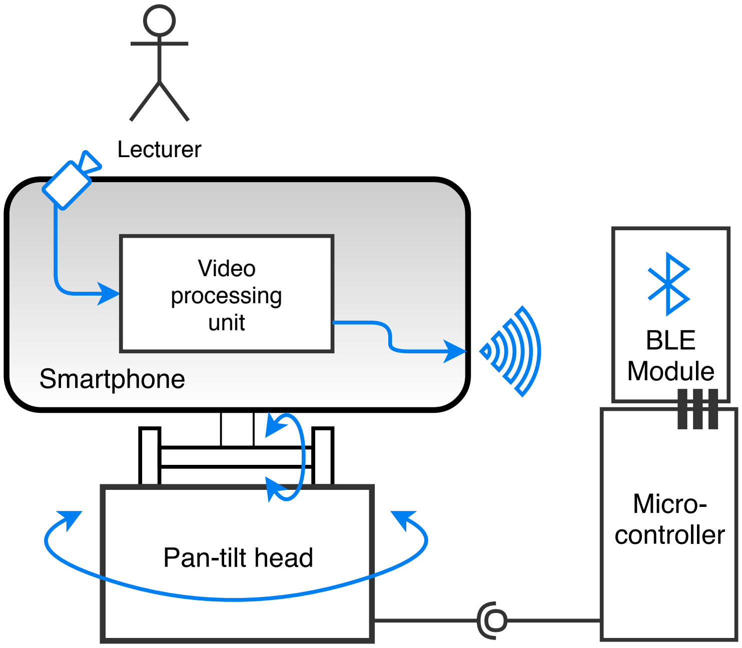


Figure 2.1: System architecture

must not be late, otherwise, the lecturer can escape the frame. Image analysis in the system we propose refers to two major processes: object detection and object tracking.

2.1.1 Object detection

To track the object, we first need to define which object to track. Object detection system recognizes objects in the image, and return their position and class. That is why it is important that the lecturer actually stands in front of the camera at the beginning of the recording session. Otherwise, the object detection system will not be able to return any detected objects and we will not be able to perform tracking. We decided to use TinyYOLO neural network model for object detection, that is adjusted to the Core ML framework.

2.1.1.1 YOLO

YOLO (You Only Look Once) is a state-of-the-art object detection system. It was invented by Redmon et al. [11] and it provides a real-time object detection by changing the entire approach to the problem. At the time, all the methods used a similar approach in which one image would be divided into many sections of different size, and each of those sections would go through a classification system, such as Support Vector Machine (SVM) or Convolutional Neural Network (CNN). Another attempt on improving the speed of object detection system was invention of the Region-based CNN (R-CNN) by Girsgick et al. [12]. R-CNN was based on the similar approach, only classifying proposed regions.

On the other hand, YOLO is based on a different principle in which the whole image only passes once through a CNN, providing as output both classes and positions of the objects.

YOLO first divides a rescaled input image into a grid of 13 × 13 cells. Then, for each cell, YOLO will predict 5 bounding boxes each of which has 5 properties - x and y coordinates of the box center, its width and height, and the confidence score. On top of that, each bounding box can belong to any of 20 object classes, which makes the total number of features per box 25. So each cell will produce 5×25 features, which leads us to the total amount of output features - $13 \times 13 \times 5 \times 25$.

For even bigger increase of the speed, the system uses resized model of YOLO called TinyYOLO. Because of the smaller neural network structure in TinyYOLO, pass through the network is much faster with a slightly decreased but still satisfying accuracy [13].

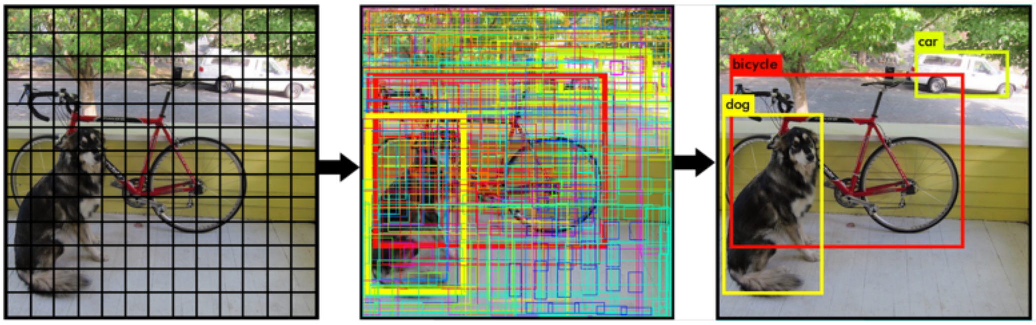


Figure 2.2: Process of detection in YOLO (source: [13])

2.1.1.2 Core ML and Vision frameworks

In order to use a device's capabilities most efficiently, Apple introduced Core ML - the foundation for domain-specific frameworks and functionality [14]. One of those frameworks is Vision, that Apple created to support image analysis tasks on iOS devices.

Core ML and Vision frameworks allow us to implement neural network models into the iOS-based applications. When we say neural network models, we refer to the trained neural networks that already can to predict (calculate) the output based on the input. Since there are different frameworks for training neural networks, their products - the models - often have a different structure than the one we might need. Currently, there already exist many neural network models in the format of *MLModel* - an instance of the iOS class that encapsulates a model's prediction methods, configuration, and model description [15]. However, sometimes the models we want to use are not in the format that conforms to the iOS standards, in which case we need to convert it. Apple itself provides a tool and instructions to convert from one of the well known and widely used models to MLModel [16].

Given the popularity of Vision framework since it was released, many developers convert existing neural network models to MLModel format and publish them online. This is also the case with the TinyYOLO model that we needed for object detection in our system. Matthijs Hollemans not only converted the TinyYOLO model into MLModel but also made a function which based on $125 \times 13 \times 13$ output returns the result in the form of structure called *Prediction* which consists of: (1) *classIndex*, representing which class the predicted object belongs to; (2) *score*, that tells us the confidence that the predicted object belongs to a particular class; and (3) *rect*, which describes all the properties of a rectangle (coordinates and size). He published his work on the blog [17] and licensed other developers to reuse and modify the code he wrote.

For this project, the system detects only objects of class person. The way to accomplish this is to filter all the predictions that don't have predicted class being *person*. In the method that parses the output of the neural network, and creates predictions, we only need to skip the predictions that belong to any other class but the class person. The next code achieves such filtering:

```
if detectedClass != personIndex {
    continue
}
```

where *personIndex* is just an index of class person in the class labels array:

```
let personIndex: Int = labels.index(of: "person")!
```

This way, for every input video frame, we get the positions of all the persons TinyYOLO network predicted.

2.1.2 Object tracking

Another problem that we have to solve in order to make the proposed system work is to track the person once we detect him. Since this is the lecturer-tracking system, we will refer to that person as a lecturer.

How can we know if the detected person is a lecturer? We cannot know based on the person detection since the lecturer is also just a person. For that reason, our system requires that only one person is detected before we can start tracking him. As soon as one person is detected, we treat him as a lecturer.

In order to track the lecturer after we get their position using the object detector, we use Vision framework's API. Unfortunately, Apple decided not to publish an explanation of how the tracking algorithm works, which makes it a "black box" of this system. However, we still know how to use it: we need to provide the position and the size of the object we want to track, in this case, a lecturer, and the video frame on which object resides, and Vision will for every next frame return predicted object's position.

There are two reasons why we should use object tracking and not object detection throughout the video processing. The first reason is that the algorithm for object tracking loads much less processor power. During a pilot test, we noticed that whenever our system uses object detection, the debugger shows "Very high" consumption of CPU, while object tracking shows "Low". This means that using object detection instead of tracking would drain the battery of the device much faster, which is not useful for recording application. The second reason is that occasionally there may be more people detected, and only one of them is a lecturer, so the system would not know whom to track.

2.1.2.1 Detection in terms of tracking

Even though Vision framework's object tracking algorithm is a black box, Apple gave a hint on their presentation [18] that the tracker tries to follow visually the most similar object that we initially defined. If we have a long sequence of frames, which is the case in recording a lecture, the object that we track will probably change its visual representation relative to the first frame. That is why Apple suggest that we should run the object detector every N frames.

In the proposed solution, we run the object detector every 100 frames, and we restart the tracker providing a newly detected object. This will make sure that we often refresh the object tracker, without exhausting the processor power.

A problem can occur if at the moment of repeated object detection there are more that one person in the frame. The tracker does not know which one of detected people it should track. The solution to this is to dismiss newly detected object position if it the object detector establishes two or more persons in the frame. In that case, the tracker will continue following the previously detected person - the lecturer.

2.1.3 Rotation algorithm

By solving the problem of object tracking, we can finally focus on the primary purpose of the system - rotating a pan-tilt head so that we do not lose the focus on the lecturer. Even though the actual signal exchange between the smartphone and the pan-tilt head triggers the head movements, in this section we will describe when and which signal to send to the head, while leaving the details about the communication to the next section.

Since the camera can be closer or further from the lecturer, and the lecturer can move closer or further from the camera, *bounding box* that surrounds the lecturer can significantly vary in size, which makes it an unreliable attribute for deciding whether we need to rotate the camera or not. However, no matter of distance between the camera and the lecturer, the center of the bounding box should always be the same. For that reason, we only observe the center of the object in each frame and decide about movements accordingly.

There are eight directions in which we can move the camera, shown on the Figure 2.3.

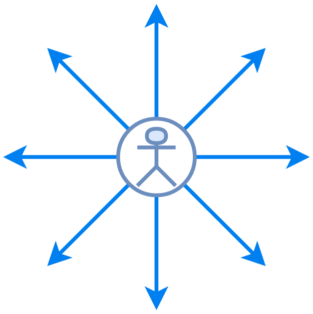
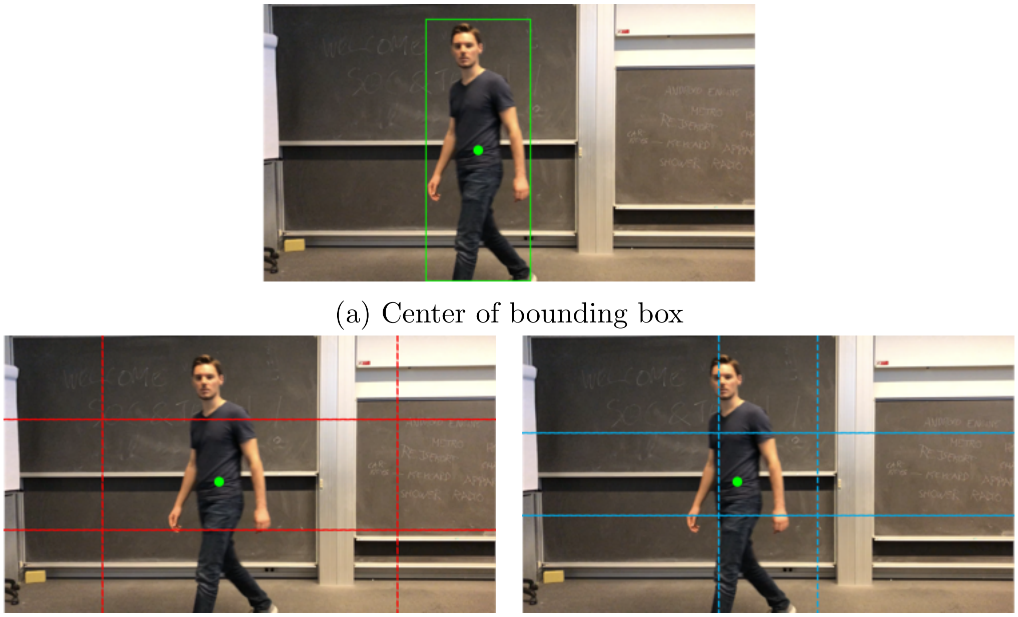


Figure 2.3: Eight directions of pan-tilt head movements

We have arbitrarily chosen the limits of the bounding box center position which crossover will trigger the movement. Given the aspect ratio of the video frames, limits in the right-left direction are a bit more far from the frame center than the ones in the up-down direction. If the center of the bounding box comes into 20 percent of the frame from the left or the right side, or if it comes into the 30 percent of the top or the bottom of the frame, the application will send a Bluetooth signal for a movement to the Arduino board over the Bluetooth module. Arduino microcontroller board will then generate electric impulses that will run the motors in the pan-tilt head, moving it in the desired direction.

To avoid too frequent movements, the camera should not stop rotating as soon as the object comes back to the previously defined limits. Instead, we will continue moving the camera until the center of the objects comes inside the center of the image. We defined this center to be middle 20 percent of the width and middle 30 percent of the image height.

To visualize the limits that will trigger the movements and the ones that will stop the movements, Figure 2.4 shows where the center (green dot) of the bounding box (green rectangle) is relative to those limits (red and blue lines).



(a) Center of bounding box

(b) Limits to trigger movement

(c) Limits to stop the movement

Figure 2.4: Bounding box center and its limits

Depending on which of eight areas outside the limits the center of bounding box comes, pan-tilt head will rotate to that same direction in order to keep the lecturer in the focus.

2.1.4 Bluetooth communication

During the lecturer-tracking process, the smartphone needs to send control signals to the pan-tilt head, so that the head performs a rotation in the direction of the lecturer when the lecturer crosses specified limit explained in the last section. Since there is no standard way to connect the smartphone and the pan-tilt head, we propose using a wireless approach - *Bluetooth Low Energy* (BLE) communication standard designed to facilitate energy-efficient wireless communication between devices. By using a wireless approach, the lecturer-tracking system does not interfere with other wired connections that smartphone might need during the lecture recording, such as power supply.

To setup a wireless communication between the smartphone and the pan-tilt head, we need to make a wireless interface of the pan-tilt head using the available remote socket. The type of the pan-tilt head we use for this project comes with a remote that we can use for moving the head in four directions (left, right, up and down), and a slider for adjusting the speed of movements. By connecting the remote socket to Arduino Uno microcontroller, we provide an application programming interface (API) for the pan-tilt head. This means that we can control the board using a set of functions that the Arduino board can compile and execute. Moreover, by an additional connection between the BLE module and the Arduino board, the pan-tilt head becomes fully controllable over BLE wireless interface.

To the pins that are responsible for the direction of movement, the microcontroller has to send a digital signal, 1 or 0. Whenever pin receives 1, it will move the head in the direction defined by a particular pin. Those four pins need to be physically connected to any of Arduino digital outs. Arduino's analog pin will generate an analog signal over digital, using Pulse Width Modulation (PWM). Over the analog pin, Arduino will generate a signal based on values between 0 and 255, denoting the speed at which the pan-tilt head will rotate. A remote socket also provides a power supply for Arduino board, so that extra power supply is not necessary. The connection between the remote socket and the Arduino board is shown on the left side of the Figure 2.5.

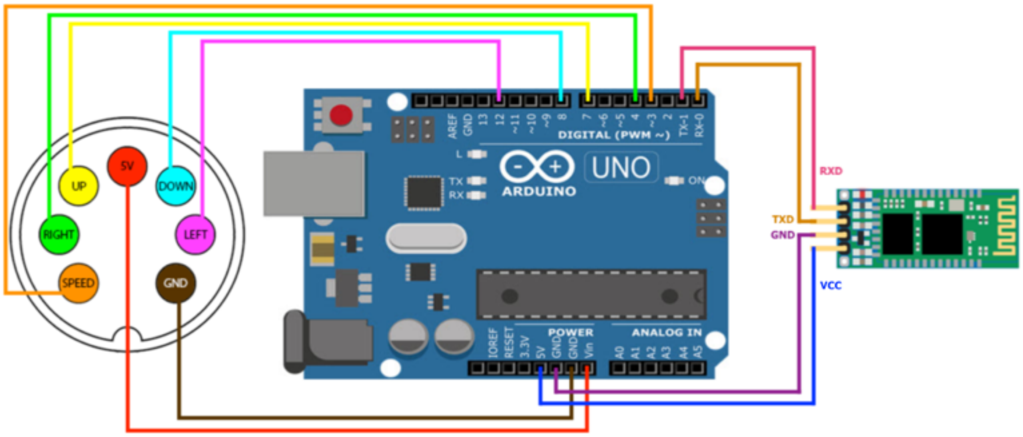


Figure 2.5: Connection between pan-tilt head remote socket (left) and Arduino board (right)

Bluetooth Low Energy module is a *peripheral* device that contains *services* and their *characteristics* for exchanging data over wireless personal network. Characteristics and services conform to Generic Attributes (GATT) hierarchical data structure. Attribute Protocol (ATT) uses GATT data to define the way of sending and receiving standard messages between two Bluetooth Low Energy devices [19].

To exchange the data, the peripheral needs to connect to a central device, in this case, a smartphone. Sending the data to a BLE module is equivalent to writing the data into a characteristic of a service.

Arduino microcontroller continually reads the data using a loop, and based on that data sends the control signals to the pan-tilt head. Arduino can read the data from the BLE module over serial TX/RX pins.

Figure 2.5 shows entire connection of pan-tilt head over the Arduino board with BLE module.

3

Results

The goal of this thesis is to show how to implement an automated lecturer-tracking system using a smartphone and a motorized pan-tilt head. In chapter 2, we introduced all the methods needed for the implementation of the proposed system. In this chapter, we will present the system and the performance of the methods that we used to realize it.

3.1 Object detection

TinyYOLO neural network model successfully performs real-time detection of the lecturer. A lecturer in front of relatively plain background is a "noiseless" environment which is a typical use case that contributes to the better performance of a person detection system. We will say that the system performs well if: (1) person is detected on all frames that contain a person; (2) average confidence score of a detected person is higher than 50%.

Table 3.1 and Figure 3.1 show the test results from which we can determine the performance of the system.

We can see that the system correctly detects in average $68,5\%$ frames that

Frames	Clip 1	Clip 2	Clip 3
Detected	100	67	70
Total	100	100	100
Percent	**100%**	**67%**	**70%**

Table 3.1: Percentage of correct detections

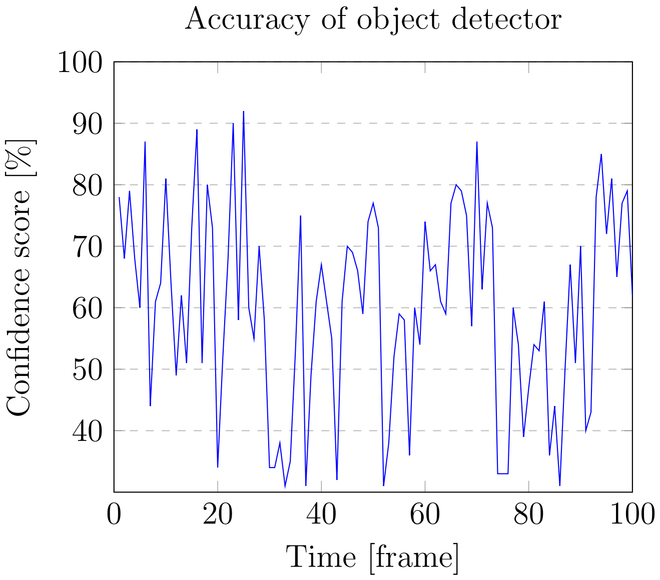


Figure 3.1: Confidence score throughout frames

contain a person in the case the person normally moves during a test. When the person stands still, detection is 100% accurate. The plot in Figure 3.1 shows that the confidence score changes significantly on a frame basis, but has the average of 60.5% - predictions with the confidence score less than 30% are dismissed which is an arbitrarily set threshold.

3.1.1 Object tracking

In order to measure the correctness of the tracking algorithm, we have to define what successful tracking is. “A tracking is successful if the center of the position of a tracked person lies inside the bounding box received from an object detector for that same person in each subsequent frame”.

We performed two different test to show the performance of the object tracker. In the first test, the tracked person makes no fast nor unusual movements. In the second test, the person often changes their position and rotates around themselves.

Two pie charts in Figure 3.2 show the results of object tracking tests. In a normal use case, we can see that the position returned by a tracker matches the position of a detector 69% of the time. However, the second test shows that sudden changes in movements and fast and irregular movements significantly decrease the tracker accuracy.

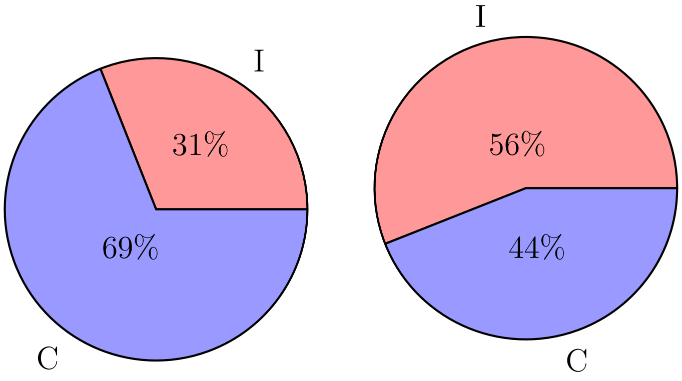


Figure 3.2: Incorrect and correct positions in tracking

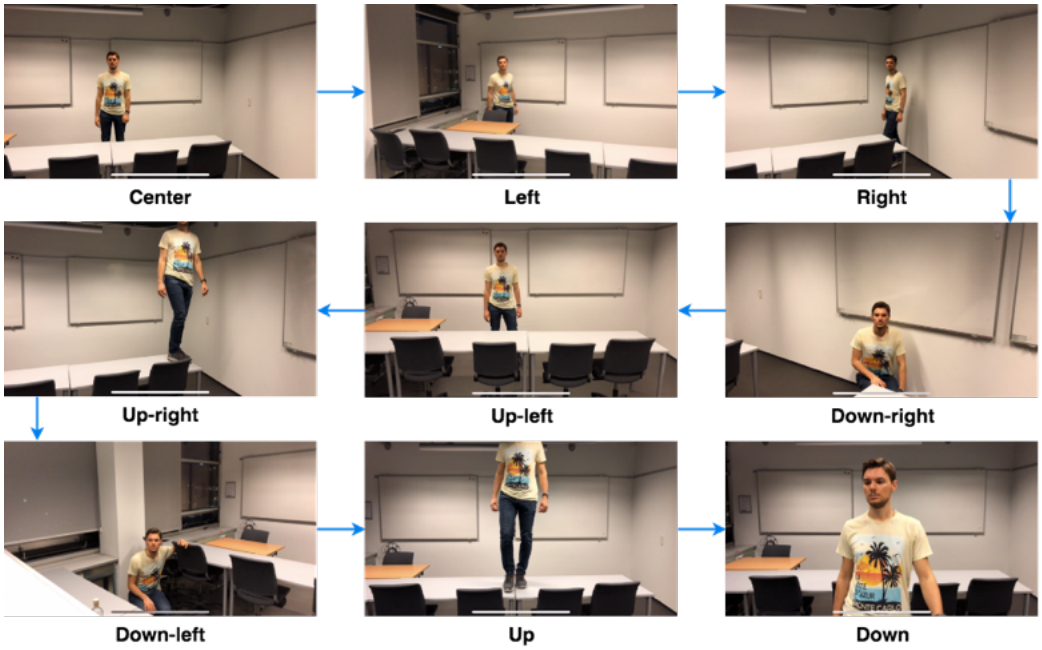


Figure 3.3: Pan-tilt head movements test results

3.2 Pan-tilt head movements

The last test of this project evaluates the performance of the pan-tilt head movements. To consider the performance of the pan-tilt head movements as a good one, the head needs to start rotating on time and must stop once the lecturer is in the center of the frame. Besides, the head must interchangeably use all eight directions of movements without losing the focus on the lecturer.

In this qualitative test, a person will move in each of eight directions at a time, trying to trigger the movement of the head. Afterward, the subject should continuously move, going in each of eight directions, and then return to the center, expecting the head to follow all the time.

Figure 3.3 shows the results of the pan-tilt head test: after moving in each direction, the tracked person has to remain in the center of the frame.

This experiment aimed to evaluate the behavior of the entire system - all object detection and tracking, and the Bluetooth connection to the pan-tilt head have to function correctly in order for the system to automatically track the lecturer. Given that the pan-tilt head rotates in the correct directions, continually following the lecturer, we can say that the system is useful.

3.3 Related work

There have been several attempts at making an automated lecturer-tracking system. However, none of them includes using a smartphone as a part of the system, which makes this project a novel approach in this field.

Winkler et al. [20] proposed using depth information of the image to improve the tracking algorithm. In their solution, they use a Microsoft Kinect device that has several sensors to provide motion and depth information about the space it observes. They claim that depth information can solve problems like occlusion and bad illumination: "*the distinction between tracked and not tracked people can be done reliably and robustly using the distance information*" [20]. However, their system required one more PTZ camera in order to capture the video.

The system proposed in this thesis uses a smartphone device with two cameras, which allows it to also obtain depth information of the image without requiring any extra sensors. A depth map is a map which consists of distance values for each pixel of the frame. Operation system in the proposed system has the API for receiving a depth map from the camera input. In Chapter 4, there is more information about the usage of depth information in our system.

Lopes et al. [9] proposed the same solution using Kinect as a crucial sensor for human detection, also introducing different lecturer gestures as triggers for controlling the camera state.

Lee and Xiong [21] proposed a real-time face tracking system based on a single PTZ camera that targets to reduce the required cost with minimal and universal hardware components and which eliminates the need for a separate operator. Even though they claim mobility as one of the main contributions of their paper, their system uses "*video capture card, embedded with a TW6816 chip, installed within the computer unit*".

The advantage of the system proposed in this thesis is that the computer is not a requirement. All the processing happens on the processing units inside the smartphone (CPU, GPU, VPU). Replacing a computer unit and a camera with a smartphone significantly improves the mobility of the system.

Cavallaro et al. [4] proposed an automated lecture cameraman based on face-detection, which "*integrates audio, video and presentation slides from a live lecture*". This system uses a PTZ camera for recording and a computer as a processing unit.

A system based on the face-detection method is error-prone to begin with because we cannot expect the lecturer not to turn away from the camera. It is a widespread case during a lecture that the lecturer turns their back on the students while writing on a blackboard. Without the presence of their face, a system like this is not able to operate.

Chou et al. [7] also proposed a system which has a face-detection as the primary lecturer detection method.

In the system proposed in this thesis, a human detection algorithm is used as a method for the lecturer detection process. This system will detect the lecturer no matter of their orientation relative to the camera.

Wulff and Rolf [8] propose using a background subtraction method for detecting the lecturer. Blob analysis that follows this method seems like a problematic and inaccurate mechanism for not only a lecturer but any human detection system. Author themselves state that this method cannot replace a human force in recording a lecture: "*The goal of OpenTrack is not to completely replace human camera operators, but to allow universities to produce lecture recordings, with a video that contains all important information in enough detail, in an efficient way.*".

4

Further improvements

An important characteristic of the proposed system is its potential for improvements. Not every system can be improved because of its software or hardware limitations. When a system has an ability to improve, it means that it can possibly perform better than it does in its current state.

The proposed system can already perform basic operations that it is made for, but with some improvements, it can become a real-world product that is simple to use and costs much less than already available systems with the same purpose.

In this chapter, we propose some improvements that would for sure significantly improve the performance of the system, and some that would solve some other problems that we did not cover in the scope of this project.

4.1 Zoom functionality

Almost all automated lecturer-tracking systems use a pan-tilt-zoom (PTZ) camera for video recording. A significant advantage of a PTZ camera is that it provides the zoom option next to a pan and tilt movement options. Having a zoom allows such set-up in which camera can be arbitrarily far from the lecturer. Moreover, when a system has a zoom function, it can handle the case in which the lecturer moves much more towards and backward from the camera, then what the system we propose can handle.

Even though all the smartphones have cameras, not all of them have such camera configurations that would, regarding quality, resist recording lectures intensively using the zoom function. Most smartphones still provide only a digital zoom function, which compromises the quality of the recorded material. However, almost all the new innovative smartphones come with enriched configurations that provide two lenses set-up and optical zoom function. We can safely conclude that such configurations will become a standard in just a few years because the smartphone development in the market is swift.

Unlike related systems, the proposed system would have the zoom function wholly disengaged from the functionality of pan-tilt head. All the zoom function controls would be used through a smartphone's API.

We expect that implementing a zoom function would also contribute to the object detection and tracking algorithms because the lecturer would have more-less similar dimensions no matter where they move.

4.2 Speed variations

The proposed system currently operates using only one constant speed of motorized pan-tilt head. We made this choice based on the pilot test of the head - the head makes very smooth movements in all directions. Sharp movements would distract the observer, compromising the quality of the material.

However, there are some cases in which we could use improvement of pan-tilt movements speed. When the lecturer comes too close to the camera and quickly move from one side to another, pan-tilt head cannot manage to follow the lecturer. Increasing the speed of the pan-tilt head movements leaving it constant would produce those sharp movements we mentioned in the last paragraph.

The only way to avoid the sharp movements, but still increase the speed, is to use acceleration and deceleration of movements. The system would have to notice that the lecturer is about to escape the frame, and gently accelerate the movement. After the lecturer stops moving, the system would detect that the center of the frame is approaching the lecturer and it would decrease the speed eventually stopping the movement.

This improvement depends on the physical capabilities of the pan-tilt head - maximum speed of pan and tilt motors.

4.3 Depth information

Using depth information of the frame is the improvement that would increase the performance of the proposed system more than any other. We state so based on solutions that used Microsoft Kinect depth sensor. As mentioned in Chapter 3, Winkler et al. [20] say that depth information facilitates a reliable and robust distinction between the lecturer and the other detected objects.

In order to acquire depth information, the system needs to have at least two cameras. We already mentioned that lead smartphone devices come in a configuration with two rare cameras. Such configuration for sure will not be discontinued unless the concept of the smartphone as we know it today changes.

Depth information of the object would allow the proposed system to handle the cases where next to the lecturer we have other people in the frame, possibly covering the lecturer when passing by the camera. If we know the distance of both the lecturer and the other person passing by, we can treat the passing person differently even if there is occlusion between those individuals. Based on depth information, we can unambiguously say that passing person is not the lecturer we track. We extract this information from sudden changes of distance in the position of the lecturer. The only gradual change means that the lecturer or another person is moving further or closer to the camera.

4.4 Initial state

The proposed system currently does not cover the case when the lecturer escapes the recording area. If by any chance there are no people that the system detects, the camera will stay in the same position until someone enters the recording area again.

It would be a useful improvement if the system would know its initial state - the position of pan-tilt head such that the camera focuses on the blackboard. It is more of a use for the observer to look at the blackboard while the system is waiting for the lecturer to come back, rather than hang over some non-useful view, for example, wall or the door.

4.5 Use of gyroscope

In the last section, we proposed the initial state of the system where the camera is facing the blackboard. The advantage of PTZ cameras is that some of them have an API for turning the camera into the desired position.

The pan-tilt head in the proposed system does not have such API, which means we need to find another way of position awareness.

Luckily, most smartphones today are equipped with a gyroscope - a device that provides the information about the smartphone's position in space. Once we set-up and run the proposed system, the gyroscope can tell if the smartphone is moving, how much, and in which direction. When the system is aware of its current position relative to the initial one, it can easily perform the sequence of movements that will turn the camera into the initial state.

Turning the camera into the initial state is not only use of gyroscope we can exploit. Since the pan-tilt head rotates only 340 degrees horizontally and ±15 degrees vertically, it can happen that the lecturer will escape the frame because the pan-tilt head cannot rotate any further. Even though the head will seem to hang, the smartphone will still send the control signals and the head will still receive them trying to move a bit more.

By using a gyroscope we can check if we came to the limit of the pan-tilt head, and we can stop sending control signals, allowing the system to recover.

5

Conclusion

In this thesis, we proposed the automated lecturer-tracking system that has a novel approach of using a smartphone both as a camera and a central processing unit. A pan-tilt head holds the smartphone and enables its rotation which results in an increased area that smartphone camera covers.

Thanks to the state-of-the-art approach in object detection - YOLO, the proposed system supports the real-time object detection. Object tracking algorithm is provided by the smartphone's software development kit framework. By combining the object detection and tracking, the system successfully performs lecturer-tracking in a simple environment.

The main contribution of this project is introducing a new, modular approach, where almost everyone already possesses the main module - the smartphone. Acquiring only missing parts makes the system cheaper to set-up, while upgrading the smartphone with the newer model necessarily causes the upgrade of the whole system, bringing the better performance.

Programming the system on a popular smartphone platform gives a lot of space for improvements and use-case adjustments.

The system can be used not only for lecture recording but also for any kind of individual recording where the help of a cameraman is missing.

Bibliography

[1] Janet Ellen Abbate. “From ARPANET to Internet: A history of ARPA-sponsored computer networks, 1966–1988”. In: (1994).

[2] Zac Woolfitt. “The effective use of video in higher education”. In: *Lectoraat Teaching, Learning and Technology. Inholland University of Applied Sciences. Rotterdam* (2015).

[3] Emily Cruse. “Using Educational Video in the Classroom: Theory, Research and Practice Multimodal Learning Styles Dual-channel Learning Motivation and Affective Learning”. In: 2007.

[4] A. Cavallaro, R. Chandrasekera, and M. Taj. “Hands-On Experience in Image Processing: The Automated Lecture Cameraman”. In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP ’07*. Vol. 3. Apr. 2007, pp. III-721-III-724. DOI: `10.1109/ICASSP.2007.366781`.

[5] Hangzhou iSmart Video Tech Co. *IP Based Lecturer Tracking and Board Writing Detecting System*. `http://www.ismart-video.com/products/Recording/LTC_Series_IP_SDI_Lecturer_Tracking_Syst141.html`. (accessed: 12.08.2018).

[6] Bolin Technology. *Pro AV / Broadcast - Bolin Technology*. `https://www.bolintechnology.com/pro-av/`. (accessed: 12.08.2018).

[7] Han-Ping Chou et al. “Automated lecture recording system”. In: *2010 International Conference on System Science and Engineering*. July 2010, pp. 167–172. DOI: `10.1109/ICSSE.2010.5551811`.

[8] B. Wulff and R. Rolf. “OpenTrack - Automated Camera Control for Lecture Recordings”. In: *2011 IEEE International Symposium on Multimedia*. Dec. 2011, pp. 549–552. DOI: `10.1109/ISM.2011.97`.

[9] Edson Lopes et al. “iReclass-An automatic system for recording classes”. In: *arXiv preprint arXiv:1501.00149* (2014).

[10] H. Nejati et al. “Smartphone and Mobile Image Processing for Assisted Living: Health-monitoring apps powered by advanced mobile imaging algorithms”. In: *IEEE Signal Processing Magazine* 33.4 (July 2016), pp. 30–48. ISSN: 1053-5888. DOI: 10.1109/MSP.2016.2549996.

[11] Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: *CoRR* abs/1506.02640 (2015). arXiv: 1506.02640. URL: http://arxiv.org/abs/1506.02640.

[12] Ross B. Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *CoRR* abs/1311.2524 (2013). arXiv: 1311.2524. URL: http://arxiv.org/abs/1311.2524.

[13] Joseph Chet Redmon. *YOLO: Real-Time Object Detection*. https://pjreddie.com/darknet/yolov2/. (Accessed on 08/30/2018).

[14] *Core ML — Apple Developer Documentation*. https://developer.apple.com/documentation/coreml. (Accessed on 08/30/2018).

[15] *MLModel - Core ML — Apple Developer Documentation*. https://developer.apple.com/documentation/coreml/mlmodel. (Accessed on 08/30/2018).

[16] *Converting Trained Models to Core ML — Apple Developer Documentation*. https://developer.apple.com/documentation/coreml/converting_trained_models_to_core_ml. (Accessed on 08/30/2018).

[17] *YOLO: Core ML versus MPSNNGraph*. http://machinethink.net/blog/yolo-coreml-versus-mps-graph/. (Accessed on 08/30/2018).

[18] *Object Tracking in Vision - WWDC 2018 - Videos - Apple Developer*. `https://developer.apple.com/videos/play/wwdc2018/716/`. (Accessed on 08/31/2018).

[19] *GATT Overview — Bluetooth Technology Website*. `https://www.bluetooth.com/specifications/gatt/generic-attributes-overview`. (Accessed on 09/01/2018).

[20] M. B. Winkler et al. "Automatic Camera Control for Tracking a Presenter during a Talk". In: *2012 IEEE International Symposium on Multimedia*. Dec. 2012, pp. 471–476. DOI: `10.1109/ISM.2012.96`.

[21] S. Lee and Z. Xiong. "A real-time face tracking system based on a single PTZ camera". In: *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*. July 2015, pp. 568–572. DOI: `10.1109/ChinaSIP.2015.7230467`.