



Trinity College Dublin

Coláiste na Tríonóide, Baile Átha Cliath

The University of Dublin

School of Computer Science and Statistics

Automating Enhancement of Blackboard Diagrams for Online Presentations

Caoimhe Mooney



April 29, 2020

A dissertation submitted in partial fulfilment of the
requirements for the degree of
MAI (Computer Engineering)

Declaration

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at <http://www.tcd.ie/calendar>

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at <http://tcd-ie.libguides.com/plagiarism/ready-steady-write>.

Signed: _____



Date: 29/04/2020

Abstract

The goal of this project is to design a system to enhance blackboard writing in video post-production. The use of online lectures has been increasing over time and, with universities moving their studies online over the past few months due to Covid-19, they have become even more prevalent. Making these diagrams easier to view for quick absorption of the information is therefore increasingly important.

The system described in this project breaks the enhancement process into three sections: blackboard location, writing location and writing enhancement.

The blackboard is located by finding the four corners of the board using the intersections of Hough lines. These corners are used to crop and warp the board, so the camera appears directly in front of the writing.

The writing is located using four kernels to find the lines in the image. Excess lines are removed by locating static objects and the lecturer. Static objects are located by finding what does not move for the duration of the video. The lecturer is located by tracking the largest area of motion in the video.

Finally, the writing is enhanced by fitting lines to the writing points in the scene. The lines are fit to extracted groups of points. To prevent only straight edges being added to the scene Hough circles is used to add to the enhanced image.

The developed system is proof of concept that it is a viable area which can be expanded upon. Adding text recognition and curve fitting to the scene will greatly improve the output.

Acknowledgements

I would like to thank my supervisor Kenneth Dawson-Howe for all his help and advice throughout the year, without which this project would not have been completed.

I would also like to thank the Computer Science and Statistics department for their support throughout our final months of college in the face of a global pandemic. They truly stepped up and ensured we knew what was happening at all stages.

Finally, I would like to thank my friends and family for all their support over the past 5 years. I would especially like to thank my Mam who proofread most of my assignments. Also, Andy who encouraged me through the stressful moments with patience and understanding. I wouldn't have been able to do this without all their help.

Table of Contents

Declaration	i
Abstract	ii
Acknowledgements	iii
Table of Figures	vi
Table of Equations	viii
Table of Tables.....	viii
1 Introduction.....	1
1.1 Goal	1
1.2 Motivation	1
1.3 Overview	2
2 Previous work	6
2.1 Indexing online lectures.....	6
2.2 Blackboard lectures	8
2.3 Handwritten diagrams enhancement	8
3 Blackboard Location.....	10
3.1 Mean shift filtering	10
3.2 Edge detection	11
3.3 Line detection	12
3.4 Locating corners	14
3.5 Locating missing corners.....	15
3.6 Frame cropping and warping.....	16
4 Writing Location.....	18
4.1 Defining writing	18
4.2 Line Location.....	18
4.3 Static object location	22
4.4 Lecturer location.....	26

4.5	Verify writing	28
4.6	Removing remaining unwanted lines	30
4.7	Prevent writing disappearing	32
5	Writing Enhancement	34
5.1	Locating line points	34
5.2	Drawing lines.....	35
5.3	Connecting endpoints	36
5.4	Circle location.....	37
5.5	Final output.....	38
6	Evaluation	40
6.1	Blackboard location results	40
6.2	Writing location results	43
6.3	Enhancement results	49
7	Future Work	55
7.1	Accurate circle enhancement	55
7.2	Solving the ground truth conundrum.....	56
7.3	Remove all non-writing lines	56
7.4	Machine learning	57
7.5	Improving efficiency	57
	Bibliography.....	59

Table of Figures

Figure 1.1: Comparison of original frame and board after being located, cropped and warped.	2
Figure 1.2: Original lines image versus final writing image.....	3
Figure 1.3: Writing image versus enhanced writing image	4
Figure 1.4: Final output of enhanced lines on coloured background.....	4
Figure 1.5: Flowchart of the full diagram enhancement process	5
Figure 3.1: Original image versus mean shift filtered image.....	11
Figure 3.2: Mean shift image versus edge image result.....	12
Figure 3.3: Hough lines and probabilistic Hough lines on original image	13
Figure 3.4: All located intersections and final chosen corner points	15
Figure 3.5: Lines from the bottom left and top right corners used to calculate the bottom right corner.....	16
Figure 3.6: Hough transform lines drawn onto the line segments	16
Figure 3.7: Original image versus cropped and warped image.....	17
Figure 4.1: Line kernels	19
Figure 4.2: Convolution with vertical kernel	19
Figure 4.3: Convolution with horizontal kernel	20
Figure 4.4: Convolution with diagonal right kernel.....	20
Figure 4.5: Convolution with diagonal left kernel	20
Figure 4.6: All kernels combined versus skeletonised lines image.....	21
Figure 4.7: Foreground image.....	23
Figure 4.8: Image of all motion in a recording	24
Figure 4.9: Inverted motion image versus dilated inverted image.....	25
Figure 4.10: Original thinned image versus static objects removed	26
Figure 4.11: Dilated lecturer's motion and lecturer's bounding box	27
Figure 4.12: Output from static object removal versus output from lecturer removal.....	28
Figure 4.13: Background image	29
Figure 4.14: Output from lecturer removal versus check against background image.....	30
Figure 4.15: Mask of all pixels in a frame which lie in range of most common colour	32
Figure 4.16: Found writing versus covered writing shown.....	33
Figure 5.1: Results of convolutions with horizontal and vertical kernels	35
Figure 5.2: Results from line fitting	36

Figure 5.3: Result of connecting endpoints.....	37
Figure 5.4: Results of Hough transform for circles.....	38
Figure 5.5: Original cropped frame versus final output	39
Figure 6.1: Result of blackboard location	40
Figure 6.2: Failed blackboard location result.....	41
Figure 6.3: Blackboard location; system output versus ground truth	42
Figure 6.4: Blackboard location; system output versus ground truth	42
Figure 6.5: Comparison of output writing frames with minor line movement	44
Figure 6.6: Original video versus ground truth versus writing output.....	45
Figure 6.7: Original video versus ground truth versus writing output.....	46
Figure 6.8: Original video versus found writing showing incorrectly labelled writing.....	47
Figure 6.9: Original video versus found writing showing lecturer breakthrough.....	48
Figure 6.10: Original video versus found writing showing issues with coloured in sections..	49
Figure 6.11: Ground truth versus enhanced output	50
Figure 6.12: Four stages of processing video; issues with line connections.....	51
Figure 6.13: Enhancement using Hough circles correctly	52
Figure 6.14: Enhancement with Hough circles incorrect output.....	53

Table of Equations

Equation 1: Parametric equation of a line.....	12
Equation 2: Euclidean distance formula	22
Equation 3: Equation of a circle	37

Table of Tables

Table 1: Bitwise or output	20
Table 2: Bitwise and output	23
Table 3: Bitwise not output.....	24

1 Introduction

1.1 Goal

The main aim of this project is to provide post-processing to video lectures to make notes being written on blackboards easier to read. This post-processing will take place before the video is published online, not in real-time. There are two main goals in this project: locate the writing on the blackboard and enhance the located writing to make it easier to understand. The system should be able to locate and enhance writing on the board. It has been primarily designed for diagrams with some writing, as many systems already exist for processing written text. What will be written is not known in advance by the system and cannot be predicted before processing.

1.2 Motivation

Video lectures and online video content within the academic world is a controversial topic in many ways. Ever since online content has been introduced to supplement and enhance the college learning experience, there has been research into how it affects students learning experience and whether it is a hinderance to learning. Many papers have found that when online lectures or video recordings are provided, students will stop attending classes and instead opt to learn from home [1]. However, if just a one to one comparison is used between those who show up to class and those who do not, students who show up to class also usually do better in exams [2]. If all factors are considered, e.g. grades, attendance and usage of online content, then a slightly different group comes out on top. Those who have attended lectures but also use video lectures as a study aid do better overall [2]. It is, however, only beneficial if they are not investing too much time in the online content. The idea of this project is to take the blackboard portion of a standard lecture and make it easier to process so, when students watch the lectures, the concepts are clearer. This would speed up the time the student spends watching the lecture and therefore improve the time spent revising versus grades received.

The use of video lectures has also exponentially increased in the past few months with educational institutes of all levels having to move to online learning as they close their doors due to Covid-19. With people of all backgrounds and abilities having to now make use of these tools, it is important to make them as accessible as possible. Often it can be difficult to read lecturers writing or interpret a diagram. If this writing is enhanced and improved it could make it possible for more people to access the lectures and benefit from them which is especially crucial when all learning is moved online.

There has been a lot of research into image processing handwriting and handwritten diagrams for recognition. Where this project aims to push this research one step further is to apply it to video and use the recognition to improve the quality of the diagrams to make them more accessible. Handwriting is often only analysed as single frame images of the completed piece. However, this project aims to analyse hand drawn diagrams in stages in the video, so the diagram still appears as it is being drawn in sections in real time. The easier the diagrams are to read; the more people will be able to use the lecture videos as an additional resource, as illegible notes are not useful to students.

1.3 Overview

This project can be broken down into three main sections of work. Locating the blackboard, locating writing and finally enhancing the located writing.

First, the blackboard is located. The writing will only be found on the blackboard and removing the external sections of the scene will prevent false positives and reduce the area on which calculations are performed. Locating the blackboard is also important for fixing any warping of the scene. The camera may not be placed directly in front of the board, but at an angle to the board instead. This may cause issues with the enhancement step when trying to find straight lines and circles. Instead the scene can be warped to remove the angle and mimic the view being straight in front of the blackboard. This is achieved by finding the lines along the edges of the board and using their intersections as the corners. These corner points are then used to create a new cropped and warped image of the board. This process will be discussed in detail in chapter 3.



FIGURE 1.1: COMPARISON OF ORIGINAL FRAME AND BOARD AFTER BEING LOCATED, CROPPED AND WARPED.

The second task is to locate the writing in the scene. First a definition of writing was drafted. The details of this definition will be discussed in full in chapter 4. For the purposes of this overview the writing was defined as lines in the image. Using this definition, the writing could then be located. This was done by first locating lines in the newly cropped scene. These lines included unwanted lines such as the podium, monitor and lecturer. The podium and monitor lines were removed by locating all static objects in the scene and removing the corresponding lines. The lecturer's lines were removed by tracking the lecturer's motion and removing all the corresponding lines. The remaining lines are then checked to see if they remain stationary. These final lines are defined as writing. This process is discussed in more detail in chapter 4.

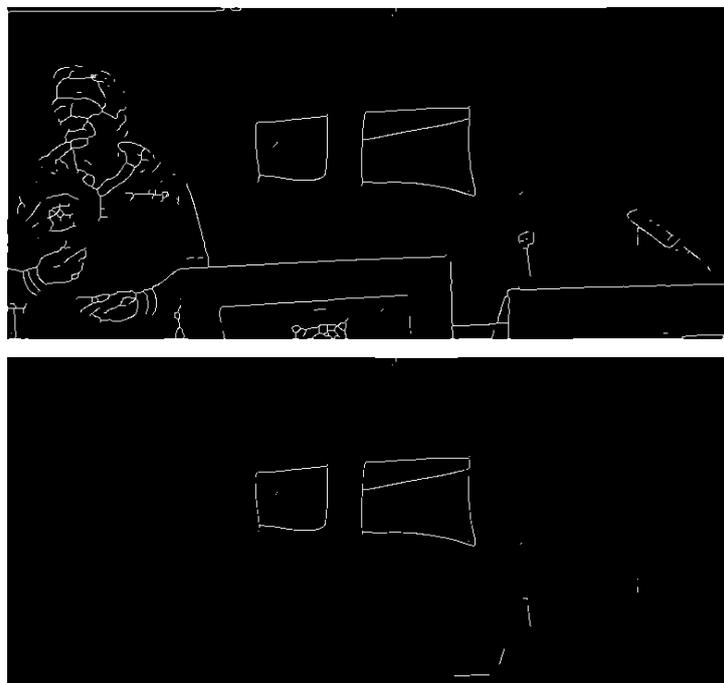


FIGURE 1.2: ORIGINAL LINES IMAGE VERSUS FINAL WRITING IMAGE

The third and final section of this project was to enhance the writing found in the second section. The purpose of this enhancement is to make hand drawn diagrams easier to interpret. When hand drawing a diagram, it is almost impossible to draw perfect straight lines or perfect circles. This enhancement step aims to transform these aspects of the hand drawn diagrams into their perfect versions. The exact steps for this process are discussed in detail in chapter 5. Figure 1.3 below shows a sample of the enhancement which takes place on lines, transforming them from curved lines into perfect straight lines.

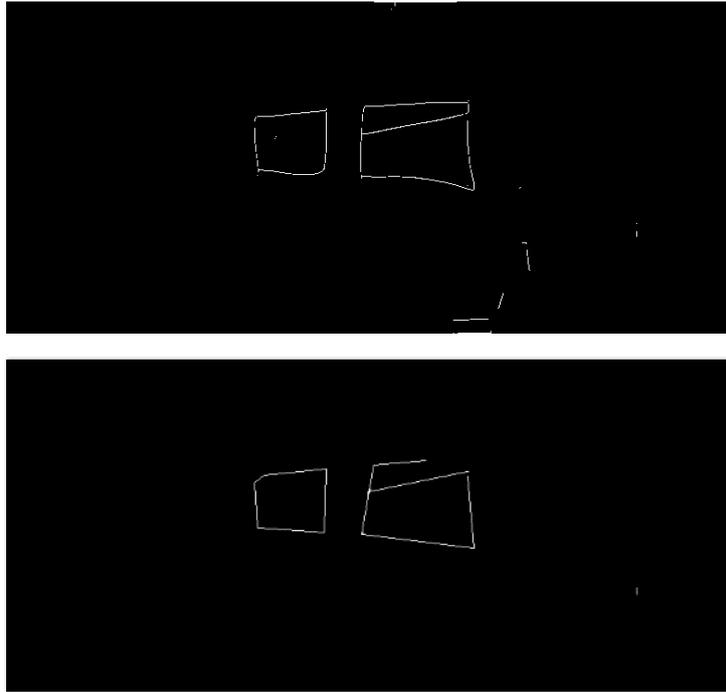


FIGURE 1.3: WRITING IMAGE VERSUS ENHANCED WRITING IMAGE

This final mask of the enhanced writing is then used to produce the output in figure 1.4. This shows the enhanced writing written on a green background matching the board in the video. This background is created by getting the average colour in the image.

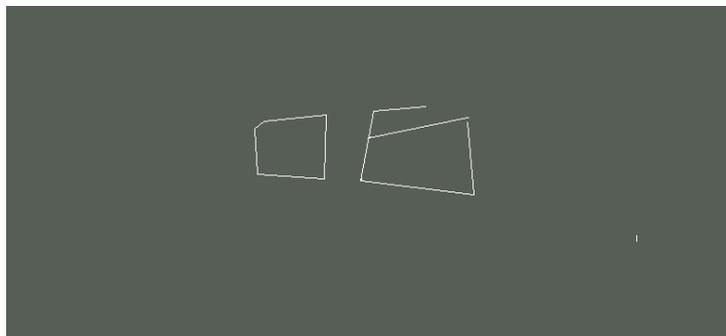


FIGURE 1.4: FINAL OUTPUT OF ENHANCED LINES ON COLOURED BACKGROUND

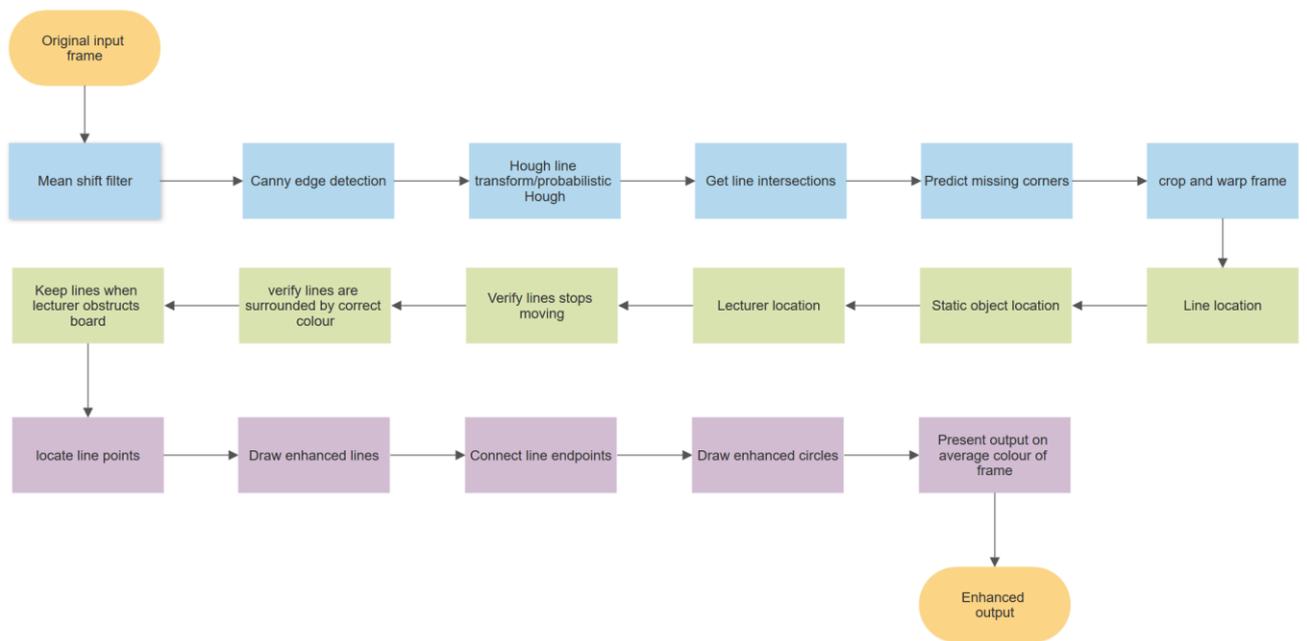


FIGURE 1.5: FLOWCHART OF THE FULL DIAGRAM ENHANCEMENT PROCESS

The code for this project was written in Python using the OpenCV library [3]. OpenCV is an open-source computer vision library originally written for C. It has since been ported to Python where it appears to have more support currently. The reason Python was chosen was because OpenCV has more support in Python and if this project is expanded to integrate machine learning algorithms, Python is more widely used for this and has far more libraries for this purpose.

2 Previous work

Online lectures and online courses have existed for as long as the internet has existed. Many universities, including Trinity [4], use eLearning sites to provide resources to students and assign and receive coursework. Not only are online lectures used as additional resources but there are many online courses, both paid and free. Within the last 3 months most universities have moved to only using online resources due to the Covid-19 pandemic [5]. This has caused a large increase in the number of online lectures being uploaded. The topic of online lectures has been extensively researched due to its prevalence in the academic community. The research in this area has many specific subsections investigating certain aspects. While there has not been research into the specific topic of this project, there are many linked research areas. The largest researched technical aspect of online lectures is the indexing and formatting of content to produce a video which is easy to view. These videos contain multiple resources such as video of the lecturer speaking as well as the slides and what the lecturer is writing. There has also been substantial research into handwriting analysis for both letters and diagrams.

2.1 Indexing online lectures

Indexing lectures is a very important aspect of producing an online lecture which is both easy to view and contains all relevant information. Primarily indexing online lectures is done in order to synchronise multiple resources within the video. Often lecturers will swap between a slide deck and the blackboard, or sometimes an overhead projector. All of these separate elements need to be included in the produced video so that the lecture can be understood in the same way as if the student is sitting in the lecture hall. There are a few different techniques which can be used to do this indexing and synchronisation depending on the type of video recorded in the lecture hall.

The indexing of videos locates key frames in the video. These key frames are frames which mark the progression of the video. They may mark the changing of a slide or the addition of writing to the blackboard. Segmenting key frames, using text on the board or text on the projection screen, is frequently achieved using Optical Character Recognition (OCR) [6]. OCR applies both pattern and feature recognition to recognise characters in an image or frame. Every character in the alphabet has a very specific shape with certain features which make it possible to determine which letters are which. There are also more common patterns of letters used to make up words. Using the two formulae that lettering follows, it is possible to recognise letters and words in a scene in order to create metadata containing the words related to each key frame.

This metadata can then be used to locate key frames in the video, such as when there is a large change in the words in the metadata.

This indexing technique can then be expanded to synchronise slides with video footage of lectures. If the video footage contains the projection screen, OCR can be used to produce metadata of the words visible on the screen [7]. The metadata can be matched with the words on the slides to find when in the video the slides shown are changed. The words on screen will only change when the slide is changed to a new slide or extra information is added to the current slide. These changes are the only points when an index needs to be created in the video as they are the important transitional topic points. This allows for synchronisation of both the slide deck and the video. These key frames can then be used as index cards for navigating the video. The projection screen is not always fully visible in a video taken in a lecture hall and so this idea has been expanded upon to also use speech recognition analysis alongside OCR to index and synchronise the video [8]. It was found that comparing the keywords extracted from using these two techniques, and finding those with the most similar words, allows the footage to be correctly synchronised. This can be further extended to retrieve keywords from the handwritten written information and add another layer of keywords to compare.

Indexing and synchronising online lecture videos is an important topic into which there has been a lot of research. There are now multiple programs available online to do these indexing and synchronising techniques in order to produce better quality online lectures. Trinity lecturers have a product called PanOpto [9] built into the eLearning site used by the university for this purpose. PanOpto allows videos to be recorded straight into the system. It then indexes the lecture and fully synchronises any videos and slides uploaded. It also produces small image tiles of each slide used in the lecture. These can be clicked on to jump to the point in the lecture where that slide is being discussed. This project is proposed as an addition to a system such as PanOpto. It is suggested to be a pre-processing step to upload to PanOpto or ideally it would be implemented as part of the upload process to PanOpto. Any hand drawn diagrams would be enhanced prior to publishing to the portal where students can view the video.

2.2 Blackboard lectures

There has been some specific work surrounding analysing and indexing video footage of lectures with writing on blackboards. Most of the work surrounding blackboards has been to create handouts for students so they do not have to watch the video of the lecture. This is an important area to focus on in terms of background for this project. Locating when the blackboard has been written on in order to create notes comes close to the first goal of this project, which is locating the writing in the video. Garber et al [10] were looking for specific key frames where the writing on the board had changed but was not occluded by the lecturer. Foreground subtraction was used to locate the lecturer and once again OCR was used for the writing. The located text was used to find the best snapshot before the board was erased. The ‘best snapshot’ was defined as the clearest view of the text in the video. Preventing the text from becoming occluded is a very important topic for writing location, whether producing notes or a full video. The issue becomes more complex when analysing a video and all frames must be kept, rather than indexing when frames can be removed.

Writing location on blackboards using the lecturer’s behaviour has also been examined [11]. As lecturers often move across a frame while writing or pacing, but stand still when talking directly to the class, it is difficult to determine which edges in the scene belong to the lecturer and which edges belong to the writing. In Onishi et al the scene is divided into static edges and dynamic edges based on the time they have remained static. Boxes are then drawn around each area of writing added to the board based on their location and the time they appeared. This allows the writing to be accurately located while keeping any static lecturer lines separate.

2.3 Handwritten diagrams enhancement

Handwriting recognition has been an area of research for decades. There has been extensive research into recognising different alphabets and handwritten notes [12][13]. As mentioned previously, it is easy to build a system to recognise characters. The alphabet is static without new symbols being added. This allows for a simple comparison to be done between located writing and known characters. Once these letters have been recognised, it is a simple substitution problem to enhance the writing. It is also possible to train a neural net to learn what an alphabet looks like and then use it as a classification tool [14].

The issue with diagrams in many circumstances is that there is no predicted outcome at the end. There is no result to compare the diagram to and mark it as being correct or most like a specific diagram. Rather than attempting to recognise the diagram and then enhance it, as is

done with writing, it is easier to approach it with a more simplistic view of what diagrams are. Diagrams are usually geometric shapes or interconnected lines. By obtaining geometric information from hand drawn diagrams it is then possible to use this information to reconstruct an easier to view diagram [15].

Song et al suggest geometric information such as centres of circles, endpoints of lines and line interconnects are all important geometric information to retrieve from a diagram. By collecting these points of interest and accurately labelling them, it is then possible to reconstruct the diagram from the information. If it is possible to reconstruct the diagram to its original state, the information can be used to improve the diagrams. Inaccurate circles can be drawn as perfect circles, lines can be drawn as accurate straight lines and other geometric shapes can be made to look how we expect them to look if not hand drawn. There has been no previous work integrating this theory into post-processing, which is where this project aims to build upon.

3 Blackboard Location

This chapter will detail the process used to locate the blackboard in the scene. There are six steps in this process to be discussed in detail; mean shift filtering, edge detection, line detection, locating corners based on line intersections, locating corners not found in the previous step, cropping and warping the frame. The overall aim in this section is to locate the four corners of the board in order to crop and warp the frame.

3.1 Mean shift filtering

The first step taken in the process of locating the blackboard is mean shift filtering. This is the filtering step taken from the mean shift segmentation algorithm [16]. This filtering algorithm is inbuilt to the OpenCV library being used for this project. Mean shift filtering is a clustering algorithm which uses spatial properties to alter the image. The pixels in a defined size area are all replaced by the average colour of the area. Any outliers to the average colour are removed to prevent the colour being skewed by outlier data. The radius of the area considered to average the pixels within is defined by the user. The purpose of this algorithm in this process is to increase the colour contrast along the edges of the board. The edges along the edge of the board vary largely in colour. By using mean shift filtering the variance in the colour is intensified. This is due to the two averages either side of the edge differing so largely from each other. The colours either side of the edge must lie in one cluster or the other as there is very little similarity between the two clusters. This will make it easier in the next step to find those edges. Figure 3.1 shows the difference between the original image and the mean shift image. The differing colours along the blackboard's edge are very pronounced.



FIGURE 3.1: ORIGINAL IMAGE VERSUS MEAN SHIFT FILTERED IMAGE

3.2 Edge detection

The second step in locating the board is to perform edge detection on the mean shift filtered image. The additional contrast along the edges from the mean shift filtering makes it easier to find the edges in the image. This step finds all the edges in the image with the main purpose being to find the edges of the board. For this step Canny edge detection [17] is used. This algorithm was chosen as it is built into OpenCV and is therefore simple to implement while also finding the correct output needed for the next step. Canny edge detection is a multiple step process to find the edges in the image. The first step is Gaussian blurring on the greyscale of the mean shift image. The image must be greyscale as Canny cannot work on a 3-channel image. This convolutes a Gaussian filter with the image in order to reduce noise. This noise reduction is especially important in locating the board as it reduces the occurrence of smaller edges, which break the long edges of the board, and removes edges found from smaller variances in colour such as the different greens in the board.

The intensity gradient of the image is found from the Gaussian blurred image. This involved convoluting filters to find the horizontal, vertical and diagonal gradients. These gradients can be used to then calculate the direction of the edges. This produces thick lines as edges. The next steps are used to thin these lines. First, non-maxima suppression is used. This sets all

values except for the local maximum to 0. This reduces the size of the edges significantly. These values are then passed through a double threshold, both a lower and upper value threshold. This reduces any noise which has survived until this point in the algorithm. Finally, the edges are tracked using hysteresis. Hysteresis removes any weak edges remaining by tracking whether they are connected to stronger edge points. If they are connected to stronger edge points they are kept as they are most likely part of that edge. The result of this edge detection is shown in figure 3.2. Only the strong edges in the image are kept and the four edges of the board are clearly shown.



FIGURE 3.2: MEAN SHIFT IMAGE VERSUS EDGE IMAGE RESULT

3.3 Line detection

Now that an edge image has been created, in order to find the location of the corners, the lines which make up the edges of the board need to be detected. This is done using both Hough lines [18] and probabilistic Hough lines [19]. First, the Hough line transform is applied to the edge image. The Hough transform uses the parametric form to describe a line.

$$x \cos \theta + y \sin \theta = r$$

EQUATION 1: PARAMETRIC EQUATION OF A LINE

For each x and y , the corresponding r and θ are plotted. This created sinusoids in the Hough space. Points that intersect the same r, θ pair are collinear. This allows us to draw lines across the image. Due to taking all points in the image into account, it can only find lines which span across a large proportion of the image. The lines found and drawn lie across the whole image. These lines can be seen in blue in figure 3.3. They span across the whole image and require a large number of points in order to be located.

The probabilistic Hough line transform uses the same mathematical theory as above in order to locate the lines. However, rather than attempting to find lines using all the edge points, it uses a subset of the edge points. This allows for shorter lines in the image to be found. These lines can be seen in green in figure 3.3. Being able to find the smaller lines is important for locating all four edges of the blackboard. In figure 3.3 the bottom edge of the board is occluded by objects such as the podium and computer monitor. This makes it impossible to find the entire edge. By finding the shorter lines in the image the bottom corners can be found using the small amount of the bottom edge which is visible



FIGURE 3.3: HOUGH LINES AND PROBABILISTIC HOUGH LINES ON ORIGINAL IMAGE

3.4 Locating corners

Once the lines which form the edges of the blackboard have been located, the four corners of the board can be found from their intersections. The corners of the board must lie at the four extremes of the board where the found perpendicular lines meet. All the lines are iterated over to find any intersections with other lines. This is done for both full lines and line segments found in the image. Found intersections are added to a list of all intersections and their corresponding lines are stored in an equivalent list of edges. In order to ensure the intersections found are corners and not overlapping lines, any intersections found must have approximately a 90° angle between them. This prevents line segments which lie almost parallel from being added to the list of possible corners.

These located corners are then divided by quadrant in order to group them with corners found in the same area. For each of these divided clusters, outliers are removed. This prevents intersections which were found closer to the centre of the board being added to the next step and skewing the result. For each of the clusters, the point which is closest to the centre of the frame is chosen. Removing the outliers prevents the point chosen being in the centre of the board. The closest point to the centre is chosen as it is the closest point to the board edge left after removing outliers. Figure 3.4 shows all the points which are labelled as corners and the resulting chosen corners from the clusters.



FIGURE 3.4: ALL LOCATED INTERSECTIONS AND FINAL CHOSEN CORNER POINTS

3.5 Locating missing corners

From the previous step there may be corners which have not been detected due to being obscured by objects such as the podium. There may also be corners which have been found in the incorrect location as with the bottom right corner in figure 3.4. This bottom right corner was found as the intersection between the board and the podium when in fact the true corner lies behind the podium. These anomalies in the corner location are found by checking the found location in comparison to its neighbouring corners. If the corner lies over a certain threshold of pixels away from the x or y coordinate of its neighbouring corner, depending on whether the neighbouring corner lies horizontally or vertically to the corner in question, then it is recalculated. In this case the threshold is set as 500 pixels. This threshold is large to allow for the board to slope away from the neighbouring corners. In order to recalculate the position of the missing or miscalculated corner, its neighbouring corners are used. The horizontal edge from the neighbouring corner, which lies along the x axis from the corner being calculated, and the vertical line from the neighbouring corner, which lies on the y axis, are used. These two lines are drawn to a blank image as shown in figure 3.5. Figure 3.5 depicts the lines required to find the bottom right corner.



FIGURE 3.5: LINES FROM THE BOTTOM LEFT AND TOP RIGHT CORNERS USED TO CALCULATE THE BOTTOM RIGHT CORNER

This image with the two lines can then be used to interpolate the missing corner. The Hough line transform is used again to draw the lines. Using the Hough transform, with adjusted parameters to increase the sensitivity, allows it to use the shorter line segments to get lines which span the image. These lines are shown in figure 3.6. The intersection of these lines is found as before and the cluster of intersections is again narrowed down to a single corner point.

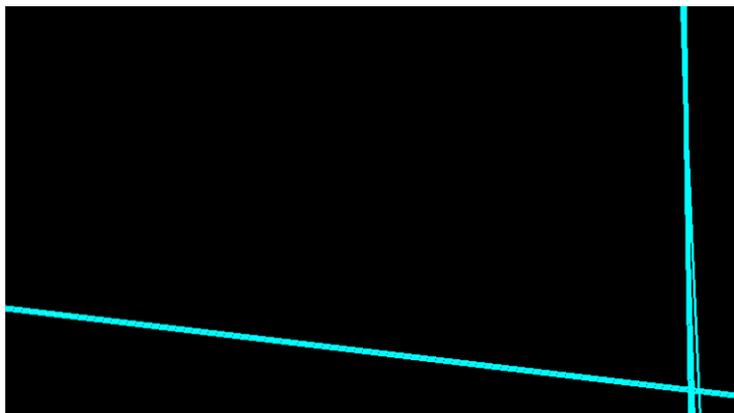


FIGURE 3.6: HOUGH TRANSFORM LINES DRAWN ONTO THE LINE SEGMENTS

If only a single neighbouring corner has been found, the intersection between its corresponding edge and the opposite edge of the frame is used. If there are no neighbouring points or an intersection cannot be found, the corner point of the original frame is used.

3.6 Frame cropping and warping

The final four corners which have been located are then used to crop and warp the frame. The minimum and maximum x values are found to calculate the width of the cropped image. The minimum and maximum y values are found to calculate the height of the cropped image. These two values are then used to create a rectangle the size of the board. Using the four found corners of the board, the image is warped to match the found corners with the corners of the rectangle

the size of the board. This produces the output image as shown in figure 3.7. The cropping ensures only the board is being analysed for the remaining calculations and reduces false positives. Warping the image ensures the camera angle appears to be straight in front of the blackboard. This makes it easier to find straight lines and circles in the image as they no longer appear distorted.



FIGURE 3.7: ORIGINAL IMAGE VERSUS CROPPED AND WARPED IMAGE

The board location is run only on the first frame of the video being processed. This means there is an assumption that the board is visible in the first frame of the video and the corners can be seen. This step is only used for the first frame in an attempt to speed up the run time of the code as it is an intense calculation step. Also, using a fixed four corners rather than recalculating the frame size and positioning, prevents inconsistencies in the output such as the frame size becoming smaller and larger or the corners flickering in position causing a flickering output.

4 Writing Location

This chapter will detail the steps taken to locate the writing in the scene. This process is applied to the video frames once they have been cropped and warped as discussed in the previous chapter.

4.1 Defining writing

In order to locate the writing in the scene, first a definition of what writing is had to be designed. This definition of writing is used to explain what writing is, in a manner that vision techniques can then be used to allow for machines to automatically locate it. There are three sections to the definition of writing used in this project. Writing is lines which vary significantly in colour from its surroundings. In this project these lines are defined specifically as being significantly lighter than their surroundings as they are chalk marks on a blackboard. This could easily be extended to include the inverse, dark lines on a light background, for whiteboards. Secondly, writing moves and then remains stationary. While being written, writing is in motion. Once the writing is completed, it remains stationary until it is erased. Finally, the writing either side of the found lines is approximately the same colour and a similar colour to the most common colour in the image. The colour either side of writing is the colour of the board. It is approximately the same colour, allowing for variations in lighting or erased chalk marks. As the image has been cropped to the size of the board, the most common colour in the scene is the colour of the board. Therefore, the colour either side of the writing should be similar to the most common colour in the image, again allowing for variations in lighting and erased chalk marks.

4.2 Line Location

From the description of writing detailed above, the first stage of the process in writing location is to locate lines in the image. In this step, all lines in the image which are lighter than their surrounding pixels are located. Four kernels are convoluted with the frame to locate these lines. These kernels can be seen in figure 4.1. These kernels are weighted with two centre pixels which are double the outer pixels. This weighting allows the location of areas where the centre pixels differ significantly from the outer pixels. There are zeros used as padding in the kernels as the lines vary in width. The two centre pixels assume the chalk marks are at least two pixels wide. The padding allows for lines wider than two pixels to also be located.

```

horizontal_kernel = np.array([[ -1, -1, -1],
                             [ -1, -1, -1],
                             [  0,  0,  0],
                             [  0,  0,  0],
                             [  2,  2,  2],
                             [  2,  2,  2],
                             [  0,  0,  0],
                             [  0,  0,  0],
                             [ -1, -1, -1],
                             [ -1, -1, -1]])

vertical_kernel = np.array([[ -1, -1,  0,  0,  2,  2,  0,  0, -1, -1],
                            [ -1, -1,  0,  0,  2,  2,  0,  0, -1, -1],
                            [ -1, -1,  0,  0,  2,  2,  0,  0, -1, -1]])

diagonal_right_kernel = np.array([[ -1, -1, -1, -1,  0,  0,  2,  2,  0,  0],
                                   [ -1, -1, -1,  0,  0,  2,  2,  0,  0, -1],
                                   [ -1, -1,  0,  0,  2,  2,  0,  0, -1, -1],
                                   [ -1,  0,  0,  2,  2,  0,  0, -1, -1, -1],
                                   [  0,  0,  2,  2,  0,  0, -1, -1, -1, -1]])

diagonal_left_kernel = np.array([[  0,  0,  2,  2,  0,  0, -1, -1, -1, -1],
                                   [ -1,  0,  0,  2,  2,  0,  0, -1, -1, -1],
                                   [ -1, -1,  0,  0,  2,  2,  0,  0, -1, -1],
                                   [ -1, -1, -1,  0,  0,  2,  2,  0,  0, -1],
                                   [ -1, -1, -1, -1,  0,  0,  2,  2,  0,  0]])

```

FIGURE 4.1: LINE KERNELS

The frame is first converted to HSL space. HSL (hue, saturation, lightness) is an alternate colour space to RGB (red, green, blue). Rather than representing colours as their composition of red, green and blue they are instead represented first by their hue. The hue determines the base colour of the pixel, for example green. The saturation channel then determines how intense the green of the pixel is. Finally, the lightness channel represents how light or dark the green is, with either end of the scale being white and black. The image is then separated into its hue, saturation and luminance channels. Each of these kernels is separately convoluted with the luminance channel of the frame. The luminance channel is used specifically as it shows there are large changes in the brightness in the scene. This allows for the location of the areas where there are light sections surrounded by dark sections. The results of each of these kernels separately being convoluted with a frame can be seen in figures 4.2-4.5.



FIGURE 4.2: CONVOLUTION WITH VERTICAL KERNEL

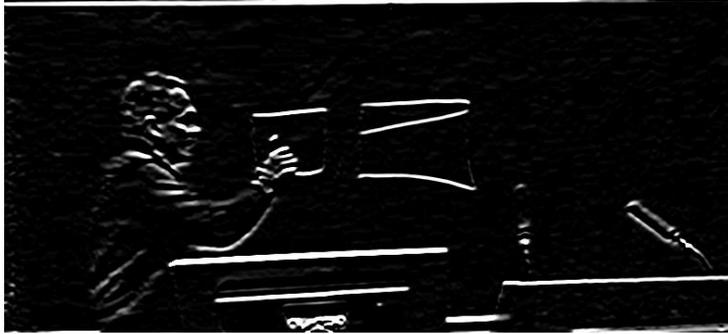


FIGURE 4.3: CONVOLUTION WITH HORIZONTAL KERNEL



FIGURE 4.4: CONVOLUTION WITH DIAGONAL RIGHT KERNEL

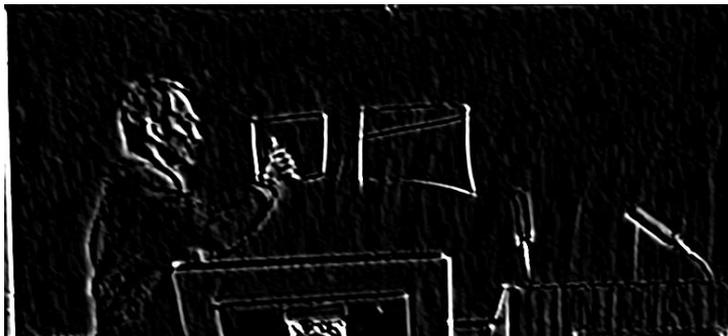


FIGURE 4.5: CONVOLUTION WITH DIAGONAL LEFT KERNEL

The convolution of these four images is added together using a bitwise or function. This applies a binary operation which decides which white and black pixels to keep. As this is a binary operation the input and outputs are represented by binary 0s and 1s. For these binary masks the 0s represent the black in the image and the 1s represent the white in the image. The resulting outputs from this operation can be seen in table 1.

INPUT 1	INPUT 2	OUTPUT
0	0	0
0	1	1
1	0	1
1	1	1

TABLE 1: BITWISE OR OUTPUT

This keeps all the areas of the mask which are highlighted by each kernel. The lines found by these four kernels are very wide due to adding the padding to the kernels. In order to find the thin lines which make up the writing, the image must be skeletonised. This means the image is reduced to the single pixel lines which make up the image. This is done using the thinned function built into the OpenCV library. The thinned function uses binary blob thinning to skeletonise the image. More specifically it uses the Zhang-Suen technique [20]. This technique involves a double pass process. Each pixel in the frame is checked against a list of conditions and is set to 0 or 255, black or white in the mask. On the first pass the conditions checked are; if there are two or more and 6 or less of the neighbouring pixels which are non-zero, if there is a single pattern of 01 in the neighbouring pixels, if the product of neighbouring pixels in positions 2, 4 and 6 is 0 and if the product of neighbouring pixels in positions 4, 6 and 8 is 0. If any of these conditions are not met, the pixel is not deleted from the image. For the second pass, the last two conditions change to: if the product of neighbouring pixels in positions 2, 4 and 8 is 0 and if the product of neighbouring pixels in positions 2, 6 and 8 is 0. This process is reiterated until the image becomes static. This produces the skeletonised image in figure 4.6. This has removed the large thick lines and replaced them with the thin, single pixel, skeletonised version of the image.



FIGURE 4.6: ALL KERNELS COMBINED VERSUS SKELETONISED LINES IMAGE

From figure 4.6 it is obvious that unwanted lines have been located alongside the writing. Namely the lines from the lecturer, the podium and the computer monitor have all been located. In order to locate the writing these extra lines must be removed.

4.3 Static object location

To remove the excess lines from the podium and monitor in figure 4.6, all static objects in the image are located. Objects such as the podium and monitor do not move throughout the video. This is based on the assumption that the camera remains static for the entirety of the recording. If the camera does not remain static, the entire scene changes and therefore the static objects being located have changed location. An initial iteration of the video is carried out in order to locate what objects remain stationary for the whole recording. To locate objects that do not move, all motion in the video is recorded. Motion in the scene is found using k-nearest neighbours (KNN) background subtraction. This is a built in function in OpenCV based on a study by Zivkovic and Van Der Heijden [21]. This background subtraction model uses the first frame of the video as the initial background. As pixels move in the image they are labelled as foreground pixels or motion. The addition of KNN to this algorithm adds the calculation of the Euclidean distance from each segment. This is the straight-line distance between two points in Euclidean space and is defined by equation 2.

$$\text{distance between point } p \text{ and } q = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

EQUATION 2: EUCLIDEAN DISTANCE FORMULA

Segments which are less than a defined threshold distance away from the segments which have been labelled as foreground are added to the foreground. This allows for an object's motion to be tracked as a whole rather than as separate segments of motion. When an object remains stationary for a defined amount of time, it is incorporated into a background image and leaves the foreground. This background image begins as the first frame in the scene and as objects move, they are added or removed from the background image. The background image is a three channel RGB image of the scene, whereas the foreground image is a mask of the motion in the scene. A sample of a foreground mask can be seen in figure 4.7.



FIGURE 4.7: FOREGROUND IMAGE

To locate the areas in which motion occurs, the foreground image is combined, by a bitwise and operation, with the lines found in the scene using the kernels previously discussed. As with the bitwise or operation previously discussed, this is a binary operation carried out on the image to determine the output. This time, both pixels must be white in order to produce an output which is white. Table 2 shows the logical operation's outputs.

INPUT 1	INPUT 2	OUTPUT
0	0	0
0	1	0
1	0	0
1	1	1

TABLE 2: BITWISE AND OUTPUT

The lines are used, rather than just the motion image, as it is the lines that do not move which are of interest, not just the whole section. This calculated frame is added to an image which acts as a running total of areas in which motion occur in the scene. The image produced is white where motion has occurred in the scene and black where objects haven't moved. This image is then saved for the next iteration of the video. There are some issues with using the KNN background subtraction model that can be seen in figure 4.8. The KNN model assumes fast changes in lighting to be motion. This results in areas of the podium and monitor being labelled as moving, as the light reflects off their surface. If the lights are turned off in the scene the motion is no longer added to the image. This is due to the model being inaccurate in the low light and because, when the lights are turned off, the lines on the board are not accurately detected. This means it is pointless to keep data from the low-light sections of the video.



FIGURE 4.8: IMAGE OF ALL MOTION IN A RECORDING

This image of motion is inverted in order to find the areas of the image which remain static. This is done using a bitwise not operation. As with the previous two bitwise operations, this follows a specific logic output. This is a much more simplistic operation. It only takes in a single input and outputs the opposite value to the input.

INPUT	OUTPUT
0	1
1	0

TABLE 3: BITWISE NOT OUTPUT

The errors caused by changes in lighting must be removed. The outer edges of the podium and monitor are also labelled as moving due to the motion of the lecturer occurring along the edge. These lines need to be included in the static mask also. These two issues are removed by dilating the inverted image. Dilating the image removes the areas inside the monitor which have been incorrectly labelled as moving and include the outer edges of the static objects. Dilating the image is done by convoluting the image with a 5x5 kernel of ones. This convolution calculates the maximum value in the area of the kernel and replaces the centre pixel with the maximum value. This causes bright areas in the image to become brighter, or in the case of this binary mask the white region grows. Prior to this dilation a morphological open is also applied to the frame. This applies erosion followed by dilation and removes noise in the image. Erosion works similarly to the dilation process but instead uses the minimum value.

As the writing in the scene is in motion before becoming static it is not counted as a static object. Some other measures are taken to reduce the changes in lighting impacting the mask produced. There is a buffer allowed at the start of the recording where the background is not added to allow the model to correctly initiate parameters. Large changes in motion, such as the majority of the screen being labelled as foreground, are also not added to the motion image as they are caused by lighting changes or the camera auto-focusing.

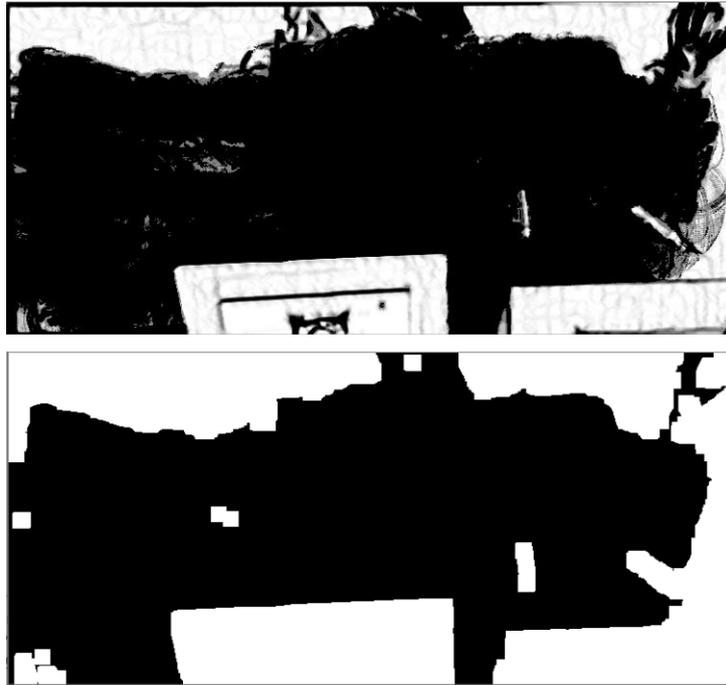


FIGURE 4.9: INVERTED MOTION IMAGE VERSUS DILATED INVERTED IMAGE

This dilated image is then used to find the lines that belong to the static objects. A bitwise and operation is carried out on the dilated image and the thinned lines image. This output is then inverted and another bitwise and operation is carried out to produce the output in figure 4.10. This multistep process is performed, rather than just using the original motion image, as it is more accurate in removing the errors produced by the KNN background model. The KNN model was used rather than a light sensitive algorithm such as codebook as its results on the lecturer's motion proved to be more accurate. Also, due to the low-light sections of the recordings being unnecessary for processing, due to the lines being impossible to process, the KNN model took preference.

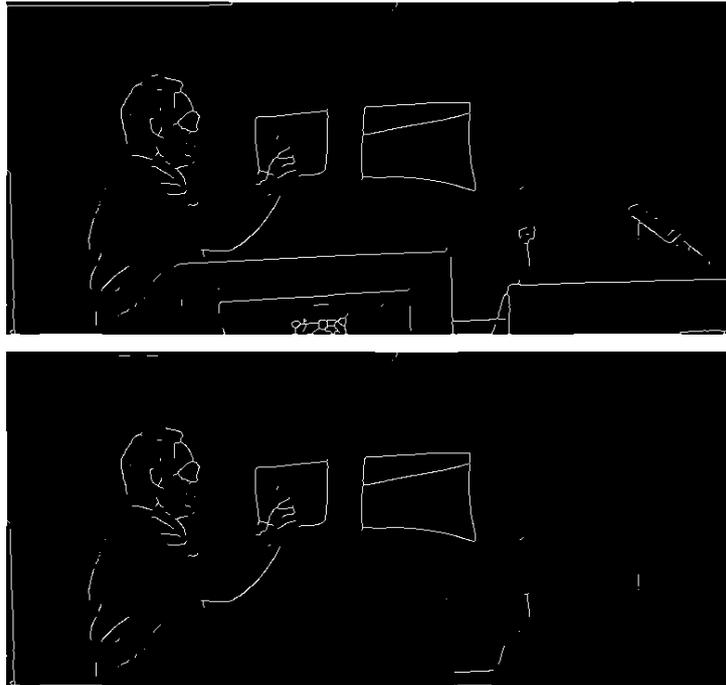


FIGURE 4.10: ORIGINAL THINNED IMAGE VERSUS STATIC OBJECTS REMOVED

4.4 Lecturer location

The lines associated with the static objects in the scene have been removed, leaving the excess lines of the lecturer to be removed from the scene. In order to locate the lecturer, the motion in the scene is analysed again. The lecturer is the largest area of motion in the scene. This means that in the foreground mask, the largest area which is white is the lecturer. Other motion in the scene is most likely noise. The writings motion is tied to the lecturer's motion as the lecturer obscures the view of the writing while it is in motion. When the lecturer moves, the motion is not perfectly surrounding the entire lecturer. Often just the lecturer's hand or head will be moving while the rest of the body is static. This can make it quite difficult to locate the whole of the lecturer as a solid object. In order to accurately locate the whole of the lecturer a two-step process is carried out.

The first step of the process is applied during the first iteration of the video where the motion of the video is analysed to find the static objects. During this iteration, the foreground image is dilated to try and join the areas of motion that belong to the lecturer into the one region. Contours are found in the foreground image. In this case contours are areas of white pixels which are connected. For each contour a bounding box is drawn. A bounding box is the minimum rectangle which can be drawn around the contour and still encompass the entire shape. Non-maxima suppression is applied to the bounding boxes. This application of non-maxima suppression joins any overlapping bounding boxes to form a singular bounding box.

This allows for multiple bounding boxes in the area of the lecturer to be amalgamated into a singular bounding box. The bounding box with the maximum area is labelled as the box which surrounds the lecturer and its area, width and height are added to a running total. This running total is used to calculate the average size bounding box for the lecturer. These numbers are then passed onto the second step in the process of locating the lecturer.

The second step for locating the lecturer takes place on the second iteration of the video. On this iteration the foreground image is found once again using the KNN background subtraction model. The foreground image is dilated, and contours are found in the image. This time the largest contour in the image is located and a bounding box only drawn around this single contour. The area of the bounding box for the contour is then compared to the average area of the lecturer bounding box from step one. If the area is smaller, the width and height of the located bounding box are replaced with the average width and height calculated in step one. This ensures that the correct contour is located, and the correct size bounding box is drawn which will encompass all the lines of the lecturer the majority of the time. The current bounding box and previous bounding box are drawn onto a blank image to create a mask of the lecturer's location as shown in figure 4.11.

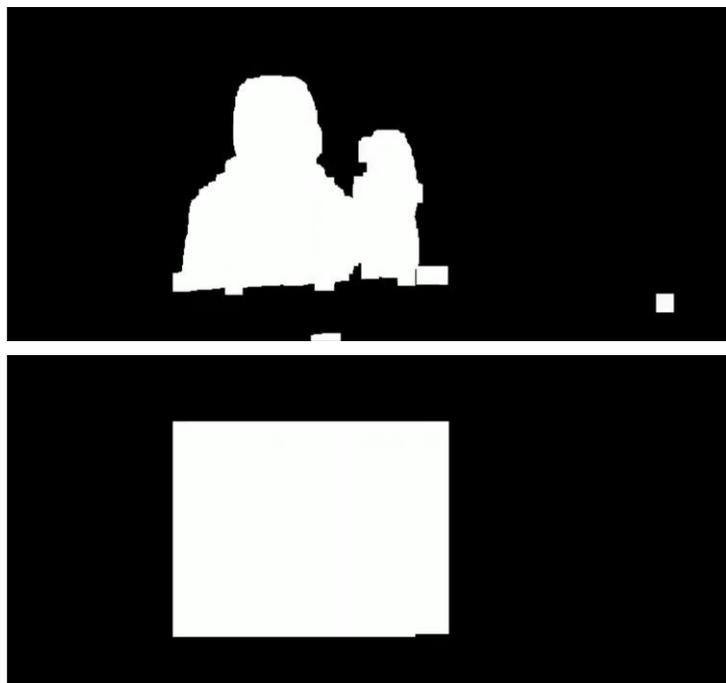


FIGURE 4.11: DILATED LECTURER'S MOTION AND LECTURER'S BOUNDING BOX

This located bounding box can then be used to remove the lecturer's lines from the scene. By inverting the mask of the lecturer's and performing a bitwise and operation this will keep any lines not associated with the lecturer. This operation is performed on the output image produced by removing the static lines in the scene. This significantly reduces the unwanted lines in the scene leaving only the lines associated with the writing and a few other lines.

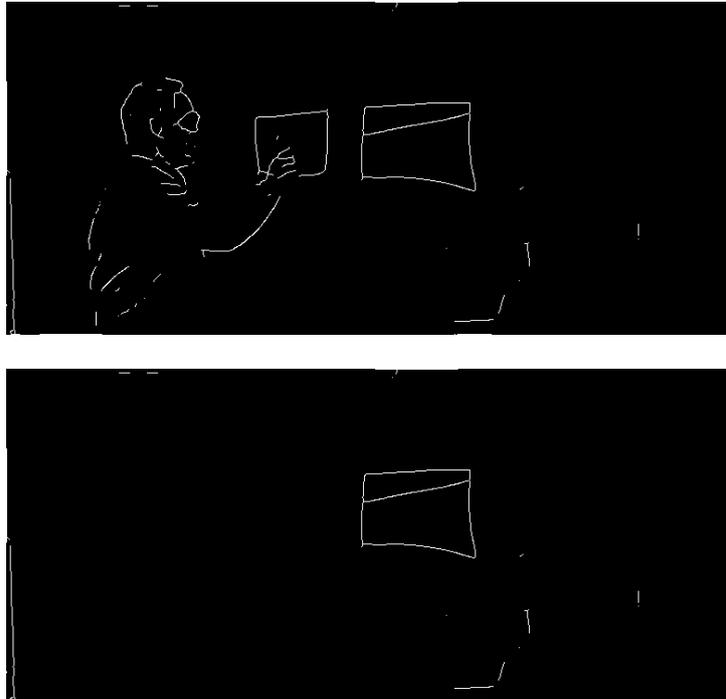


FIGURE 4.12: OUTPUT FROM STATIC OBJECT REMOVAL VERSUS OUTPUT FROM LECTURER REMOVAL

4.5 Verify writing

Once the major areas of excess lines have been found and removed, the remaining lines are tested to see if they pass the second requirement to be classified as writing. This requirement is that writing moves and then remains static. When the writing is moving it is included in the foreground mask. It is also included in the lecturer mask as the movement of writing is attached to the end of the movement of the lecturer's arm. Once the writing becomes static, it integrates into the background image. As explained earlier, the background image updates to include anything in the scene which remains static over a specific length of time. This produces an image as shown in figure 4.13. This background image includes any writing in the scene approximately 100 frames after it has been written. If the lecturer remains stationary for longer than those 100 frames, areas of the lecturer can also break through into the background image.

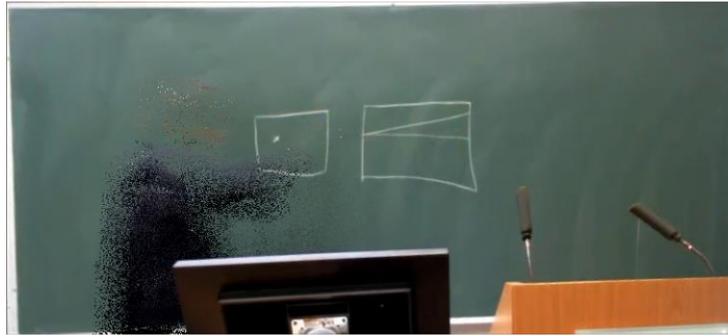


FIGURE 4.13: BACKGROUND IMAGE

As it takes 100 frames for the background image to fully take all changes on board, the current frame being processed is compared to a background 100 frames in the future. This ensures that the lines which are left in the frame are added to the background image and therefore remain static. This does not solve issues with extra lines which remain from static objects as they also are added to the background image. Using this process leaves an excess of 100 frames at the end of the video without a future background to compare to. These 100 frames are left over as the future video the current frames are being compared to has ended. To solve this issue, the last 100 frames of the video are compared to the final background in the video. The background cannot change once the video is over, therefore nothing new can be added to the background. This verification step only removes minor excess lines but is mainly an assurance that the second requirement in the definition has been met. This is obvious in figure 4.14 which shows the output from the lecturer removal and the comparison of static lines. Only small lines in the bottom right and left corners have been removed. Any excess lines remaining are left from the static objects in the scene.

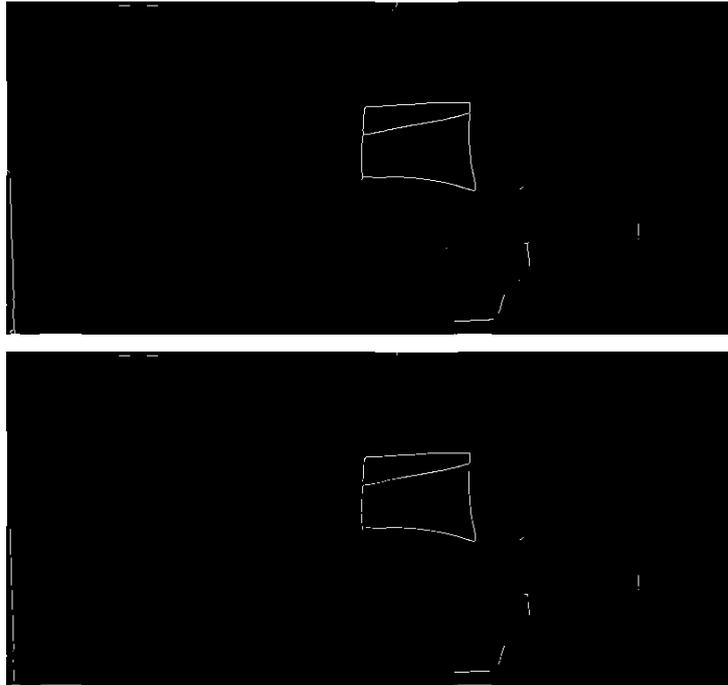


FIGURE 4.14: OUTPUT FROM LECTURER REMOVAL VERSUS CHECK AGAINST BACKGROUND IMAGE

4.6 Removing remaining unwanted lines

Even with the last verification step in the process there are still some unwanted lines which breakthrough in the footage. These lines are mostly surrounding the lecturer when the detection fails or the static objects whose lines have not yet been removed. The lecturer's lines still breakthrough after the last verification step if they have stood still for long enough. Standing still for long periods of time is also what causes the lecturer detection to fail as the motion is no longer there to analyse. To remove these lines the final specification in the writing definition is implemented. This specification is that the colour either side of lines defined as writing should be roughly the same colour and should be similar to the average colour in the frame. The first part of this can be verified by checking the values of the image either side of the line. Each pixel contains 3 values, one for each channel. The line image is currently a mask which stores the location of all the pixels in the frame which are lines. These pixel locations can be used to check the colours of the pixels either side of the line in the original frame. If the pixels either side of the lines are similar in colour they are kept as lines for the next stage of the process. These pixels are not assumed to be the exact same colour as there may be variances in the green on the board due to the lighting or previous chalk marks being erased leaving behind marks. A threshold difference is instead allowed between the two pixels to allow for these slight differences. This threshold allows for a variance in any of the channels as slight changes in any of the RGB channels can cause the colour to significantly change. The threshold was chosen

by examining the variance in the green on the board and allowing for the same amount of variance between pixels either side of a line.

Next, in order to compare the colours either side of the line to the most common colour in the scene, the most common colour must be calculated. This calculation is done in HSV space as it allows for more accurate calculation of the colour. HSV (hue, saturation, value) colour space is similar to the HSL space previously discussed. The hue channel for both of these representations is the same. The saturation and value channels are where there are differences in these two colour space models. In the HSV colour space, pure colours, colours which are not mixed to be created, have a saturation value of 1, while in HSL colour space they have a saturation of 0.5. Adding white to a colour in HSV colour space does not change the tone of the colour, only adding black does this, whereas adding either white or black changes the tone in HSL. This means that the top and bottom of the lightness scale are white and black in HSL whereas for HSV black is at the bottom of the value scale but white is in the centre of the saturation scale. The most common colour is located using k-means clustering [22]. K-means clustering finds the centres of k number of clusters. The pixels are then labelled according to their proximity in colour to the clusters. The clusters grow in size as corresponding pixels are added. In order to locate the most common colour in the image the largest cluster is chosen as it is the cluster with the most corresponding pixels of similar colour values. A range is then chosen either side of this value in order to locate the entire board. The colour of the cluster is the average of all the pixels added to the cluster and it is therefore not accurate to use that single colour to locate the board. Using the HSV colour space is very useful when choosing the range. The hue on the board does not vary much as they are all shades of green. However, the saturation and value vary significantly. This is mainly due to lighting and shadows cast by the lecturer. The range chosen reflects these variances allowing for small fluctuations in the hue and larger fluctuation in the saturation and value. This range is used to create a mask of everywhere in the frame which is inside this range.

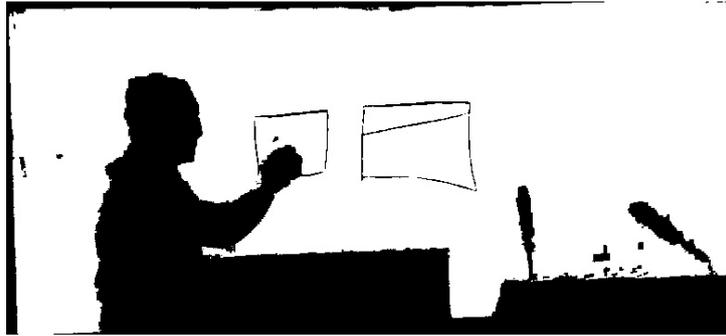


FIGURE 4.15: MASK OF ALL PIXELS IN A FRAME WHICH LIE IN RANGE OF MOST COMMON COLOUR

The values either side of each line which has been defined as writing are checked against this mask. If the values on both sides of the line is white (255) in the mask the line is kept. The values either side of the line are checked three pixels away, rather than the directly adjacent pixels, as this reduces the static lines at the edges of objects appearing. This prevents breakthrough of lines from static objects, which still appear, and lecturer breakthrough, which has passed all other stages of verification.

4.7 Prevent writing disappearing

Once the writing has been located, there are some occasions during the video in which the writing should remain static and not change. The first of these is when the lights are turned off. When the lights are off it is not possible to analyse the writing in the scene. As mentioned above, whether the lights are on or off is determined by the average luminance in the scene. When the luminance drops below the threshold for the lights being on, whatever the last output frame was is kept as the current frame until the lights are turned back on again. Once the lights are turned back on the scene starts to be processed again. If anything was written on the board while the lights were turned off, it is added to the output once the lights are back on.

The second situation where the output on the board must remain static is when the lecturer moves in front of the writing. If the lecturer moves in front of the writing, to add to it or to stand and speak, the writing should remain static. This prevents the diagram appearing and reappearing whenever the lecturer moves in front of it. This is done by again using the bounding box for the lecturer. Wherever the lecturer is in the scene according to the mask is kept as the lines from the previous frame. On each iteration the previous frame is stored in order to refer to for this step. Anywhere in the current writing output which has the same location as a white pixel in the lecturer mask is replaced with the previous frame in the same location. This ensures continuity between the frames and prevents the lecturer's movements

from causing the lines to disappear. Figure 4.16 shows the output once the covered lines have been added back into the frame.

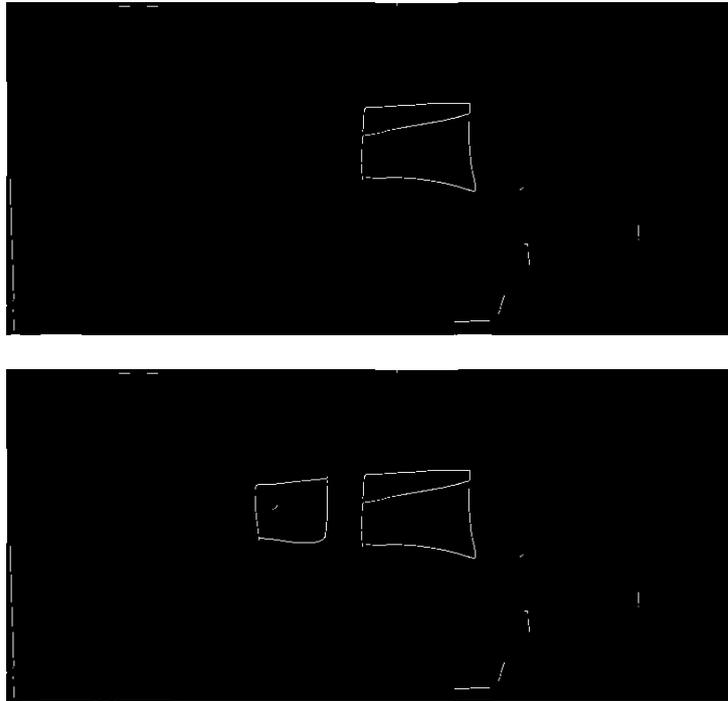


FIGURE 4.16: FOUND WRITING VERSUS COVERED WRITING SHOWN

5 Writing Enhancement

The final step in this project was enhancing the writing. The aim of this step was to transform the writing located in the last process into an easier to view version. This process involved enhancing lines and enhancing circles. The line enhancement process is a multistep process taken to fit the written lines with straight lines, and connect those lines, to create the correct diagrams. The circles were enhanced using the Hough transform for circles.

5.1 Locating line points

In order to fit lines to the hand drawn lines, the points which needed to have a line fit to them had to be located. The first stage of this process was to separate the horizontal and vertical lines from each other. This was done in order to prevent the corners where lines meet having the incorrect lines fitted. If the lines were not separated by horizontal and vertical, diagonal lines would be fitted across any joined lines. As the aim of this section is to still accurately portray the diagram while enhancing it, this risk had to be mitigated by separating the lines. This line separation was done by using morphological operations. For both horizontal and vertical, a rectangular kernel was used. The shape of the kernel is what separates the lines. For the horizontal lines a kernel of 3×1 was used. This removes the vertical lines as they do not have neighbouring pixels on the horizontal axis. For the vertical pixels the inverse kernel of 1×3 was used. This preserves only the vertical lines as the horizontal lines do not have neighbouring pixels on the vertical axis. The horizontal kernel preserves the diagonal lines in the image. The two kernels are convoluted with the writing image to produce the two separate images in figure 5.1.

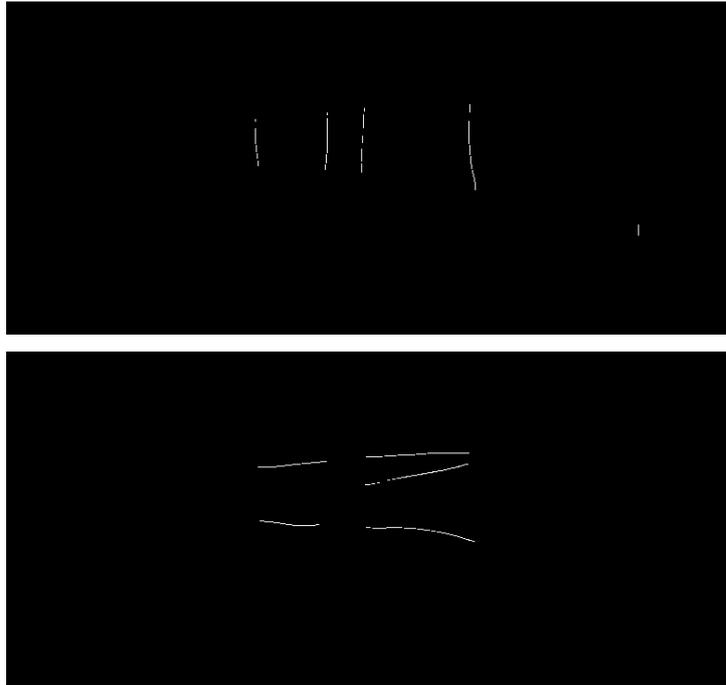


FIGURE 5.1: RESULTS OF CONVOLUTIONS WITH HORIZONTAL AND VERTICAL KERNELS

5.2 Drawing lines

As can be seen in figure 5.1, the output from the convolution produces broken lines. In order to fit a line to the entire line and not small sections of each of the lines, the image is dilated to close the gaps in the lines. Contours can then be found from the dilated image. The contours from the dilated image are more reliable as they rely on the pixels being connected to the neighbouring pixels. For each contour, a line is fitted to the original points in the contour, not the dilated points. This is done by drawing a single contour on a blank image. By using a bitwise and operation between the single drawn contour and the vertical or horizontal image, depending on which image is being processed, the set of points which a line can be fit to is found. Using the two points which are furthest apart as the ends of the line, a straight line is drawn. This process is used, rather than using a line fit function, as it achieves better results. The OpenCV library has a built-in function which fits a line to a set of points. This line fit function finds the line which crosses the highest number of points in the set of points provided. However, this function did not prove to achieve better results than simply using the two most distant points in the set. This is most likely due to the fact the line fit to the set of points is a whole line which continues across the entirety of the image rather than a subsection of the image. Using the contour to find the endpoints of these lines produced an inaccurate result. This fit line function also finds it difficult to deal with the slope of vertical lines which are an important aspect of the diagrams to locate. The lines found from using the two furthest line

points result in accurate line sections but do not accurately find the intersection points such as corners. This is clear in figure 5.2 where few of the corners are correctly connected.

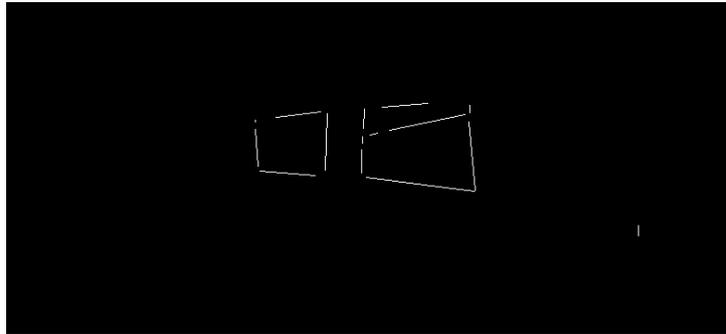


FIGURE 5.2: RESULTS FROM LINE FITTING

5.3 Connecting endpoints

To fix the issues surrounding the intersection areas the endpoints of the lines are connected. In order to do this the distance between each line end is calculated. The closest line end, which is not the other end of the same line, is then connected to that line end provided it is under a specific distance threshold. This distance threshold is set to prevent neighbouring diagrams becoming amalgamated into a singular diagram. The ends of the lines are joined in two different ways to try and correctly represent the diagrams without distorting them. If the two end points of the lines are both from vertical lines or both from horizontal lines the two nearest end points are replaced by the average of the points. If, from the two end points, one is from a horizontal line and one is from a vertical line, the two points are replaced with the x coordinate of the vertical line and the y coordinate of the horizontal line. This is to prevent the diagram from becoming distorted and to attempt to keep right angles in the diagrams. While this technique works very well for diagrams with elements such as squares and rectangles, it does cause some issues in shapes such as triangles. This is because the angles in triangles should be kept as acute angles rather than straightened out into a right angle to match squares and rectangles. This is a compromise which must be made in order to attempt to preserve some of the diagrams.

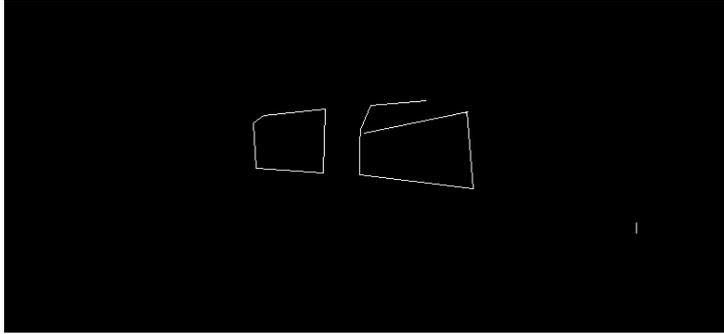


FIGURE 5.3: RESULT OF CONNECTING ENDPOINTS

The lines found in each frame can flicker due to finding a slightly different contour, resulting in a different angled line, or a different line end may be the closest in the next frame, resulting in different line ends being connected. In order to reduce the changes in the diagram which cause flickering and can cause issues viewing the diagram, the overlap of the lines with the original writing output is checked. If there is a large addition to the lines which do not overlap with the located writing, the frame is not updated and the previous found lines are kept instead. This reduces the flickering issues and issues with viewing the diagram.

5.4 Circle location

For circles in the video the line location can cause issues as it creates perfect straight lines out of the curved sections of the circle. To try and compensate for this issue, the Hough transform for circles is used [23]. This works similarly to the Hough transform for lines previously described. However, rather than using the parametric line equation, the equation of a circle is used instead.

$$(x - a)^2 + (y - b)^2 = r^2$$

EQUATION 3: EQUATION OF A CIRCLE

Using the points from the writing which have not had lines fitted, the Hough transform for circles is applied. This searches for circles within a range of radii which have enough crossings to be defined as a circle, in the same way the Hough for lines searched for points with multiple crossings. Hough for circles accuracy is not as good as the line detection method used. It often struggles to find circles of the correct radius or cannot consistently find the same circle frame to frame. Some methods are put in place to attempt to mitigate these issues. First, circles are checked for their overlap with lines already located. If they have too few overlaps, they are most likely not an accurate circle. If they have too many, they have likely been located within a filled shape. Secondly, the image must be improved by the circle for it to be added. The same test used to reduce the flickering in the lines is run on the circles. This test is run for each

individual circle found in the image by drawing the circle onto a blank image and testing it against the original found writing image. If the addition of the circle does not cause the number of pixels which do not overlap to go up over a certain threshold, then the circle can be added to the image. Finally, the option to have the circles added to the video can be set using a flag when running the code. This allows the user to determine whether they need the circles to be added. These are all added to attempt to prevent the many false positives which Hough for circles brings. Figure 5.4 shows the output of Hough for circles on a frame which includes a circle.

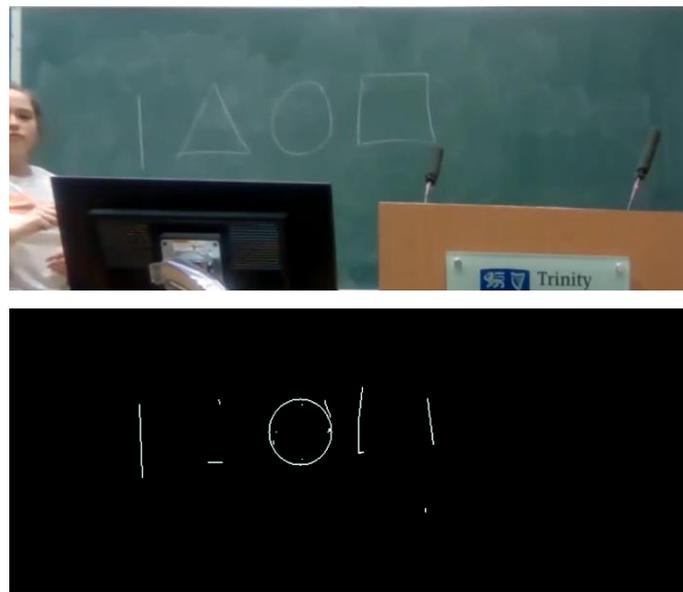


FIGURE 5.4: RESULTS OF HOUGH TRANSFORM FOR CIRCLES

5.5 Final output

The final output of the video is presented on a green background as shown in figure 5.5. This background colour is achieved by getting the average colour across the whole image. The average, rather than the most common, is used for efficiency. Using k-means clustering is not highly efficient in terms of run-time. As the k-mean clustering used has been run in HSV space it cannot be used for the output colour. Converting the single colour from HSV to RGB produces an inaccurate colour. Getting the average colour, while it does include the static pixels and lecturer pixels, still produces a green colour similar to the board due to the board being much more prevalent in the image. Once the average colour has been calculated, an image the size of the original cropped frame is filled with the average colour. To convert the mask of the writing to output on the board, a method comparable to compositing [24] is used. First the mask of the enhanced writing is combined with a blank white image using a bitwise and operation. The mask is then inverted using a bitwise not operation and combined with our background

average image using a bitwise and operation. The two images are combined using a bitwise or operation. This produces an output image with the enhanced writing in white and the background of the average colour.

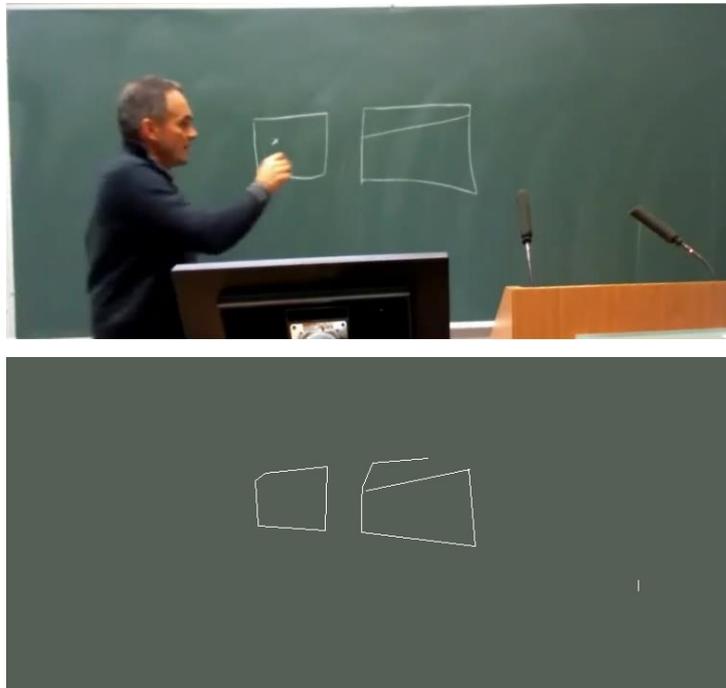


FIGURE 5.5: ORIGINAL CROPPED FRAME VERSUS FINAL OUTPUT

6 Evaluation

Evaluating this system is a difficult process. The purpose of the system is to make the written diagrams easier for students to interpret. However, the diagrams still need to remain accurate. For the evaluation of this system, five different videos were used: three 50-minute lecture videos and two shorter videos designed specifically for testing. The two testing videos have multiple different shapes and diagram types in order to push the system as far as possible. They also use different camera angles to test the board location. The three 50-minute videos are recorded live lectures. These three videos test whether the system is able to work in real-life circumstances. They have less diagrams written but include more of the slow lecturer movement and lighting changes which test the system in a different way.

6.1 Blackboard location results

The first step in the process of this system is to locate the blackboard and therefore will be the first section of the video looked at for results. The system finds the corners of the blackboard and then crops and warps the frame. In most situations the blackboard is located close to its actual size.



FIGURE 6.1: RESULT OF BLACKBOARD LOCATION

Figure 6.1 shows the location, cropping and warping of the blackboard in the frame. The top left corner of the blackboard is not easily detected due to multiple edges in close proximity. The bottom of the projection screen and the top edge of the blackboard become difficult to

separate. The bottom left of the cropped image shows the interpolation step in action. A sliver of the image is shown as black. This is due to the corner of the blackboard being out of frame. When the corner is located and the frame warped it adds extra black pixels where there are no matching pixels from the original frame.

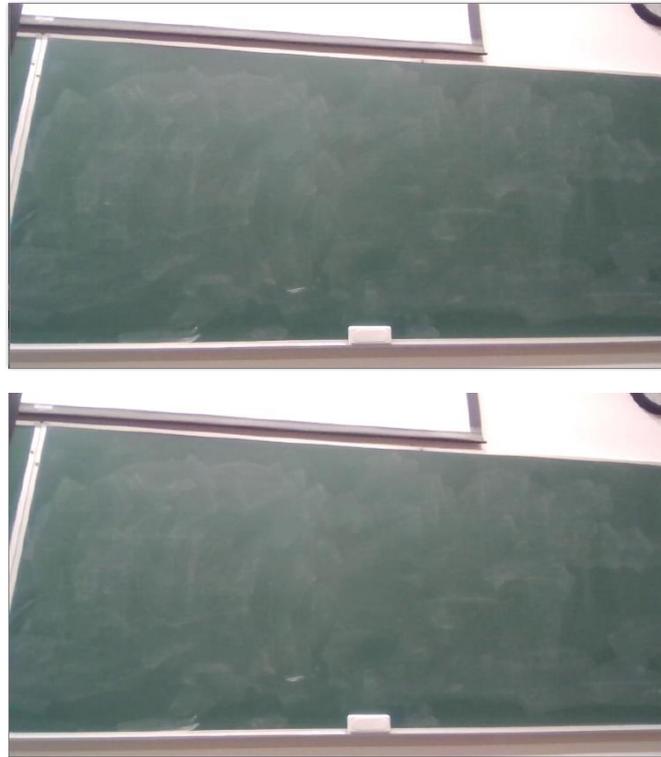


FIGURE 6.2: FAILED BLACKBOARD LOCATION RESULT

Figure 6.2 shows the original and cropped and warped images of a failed attempt at locating the blackboard. Due to the location and angle of the camera in relation to the board it is impossible for the board to be correctly located. Only a single corner is visible and due to the angle and lighting in the scene it is not located. When this happens, the corners are set to the original image's corners resulting in the outputted frame being the same as the original.

Numerically evaluating the system as a whole is quite difficult and thus has been broken down into the evaluation of smaller sections. In order to evaluate any of the system, ground truth must be made with which to compare the output. For locating the blackboard, the ground truth was created by using the correct corners of the board to crop and warp the image. This then created an ideal image of what the output should look like that the systems output could be compared to.



FIGURE 6.3: BLACKBOARD LOCATION; SYSTEM OUTPUT VERSUS GROUND TRUTH



FIGURE 6.4: BLACKBOARD LOCATION; SYSTEM OUTPUT VERSUS GROUND TRUTH

Figure 6.3 and 6.4 compare the correctly located and warped images used as ground truth to outputs taken from the system. These ground truth images are used to calculate the differences between the board located by the system and what should have been located as the board. As the images are not the same size, they cannot be compared exactly pixel by pixel. Even if they were compared pixel by pixel, because the image is warped, the location of each of the objects can change and would therefore show the entire image as different. To accurately compare the

images with their ground truth, the distance between the located corners and the actual corners and a comparison of histograms between the images was used.

For each of the located corners, the distance was calculated between it and the real corner from the ground truth. On average, across the 5 videos, there was a distance of 18.5 pixels between the correct corner and the located corner. This number appears to be quite high when the output videos still appear quite similar. This is due to some of the corners having a much higher distance from the correct corner which skews the mean. The median value for the distance is 7 pixels which seems more reasonable when looking at the output. Even the videos with a large distance, such as the image in figure 6.3, still appear as easy to interpret and are still possible to work with to continue the rest of the process.

In order to take into consideration how the output actually looks, and not just the accuracy of the corner location, a comparison of histograms [22] was carried out on the two images. For this a histogram of the hue channel of each image was created. This stores distribution of the different hues in the images. The histogram for each image is then compared to see how similar the hues in the images are. This is a good comparison of the images as, if the board is not correctly located, the background surrounding the board, which is an entirely different hue, is added to the image. The different hues from the background being added to the histogram result in a large change in the distribution. The ground truth and output image's histograms are compared to get a percentage similarity. This percentage similarity accounts for both the hues in the histogram and the amount of that hue present. On average, across the 5 tested videos the histograms are 90.5% similar. In the image in figure 6.3, where the board has not been located, the histograms are 88.6% similar. There is still a high similarity due to the board being the majority of the image.

6.2 Writing location results

Assessing the success and failure of writing location is a difficult task. The first issue arises when attempting to determine the ground truth of the writing. The question of who should be creating the ground truth appears. Should the lecturer who originally drew the diagrams determine the ground truth? Or is it acceptable for someone else to use their own interpretation of the diagrams to create the ground truth? If someone else has interpreted the diagrams, can it really be called accurate ground truth since it may not match the original idea of the diagram? These are all very important questions which are not easily answerable. For the purposes of assessing this system the ground truth was not drawn by the lecturer. This was done for ease

of assessment but there are definite points that can be argued as to whether it can be used as highly accurate ground truth due to not being drawn by the original illustrator of the diagrams. Once who can draw the ground truth is determined, another issue arises. The writing found by the system is a single pixel in width. However, due to the way the lines are thinned, the pixels which are kept can change from frame to frame without the image shown changing dramatically to a person viewing the scene. When drawing ground truth for the writing, this then raises the question as to whether the ground truth should also only be a single pixel wide. If it is only a single pixel wide, how is the correct line on which to draw the pixels determined? Or should the ground truth line be thicker in order to incorporate slight changes in the chosen line? Both options were analysed to attempt to find the best way to capture these results.

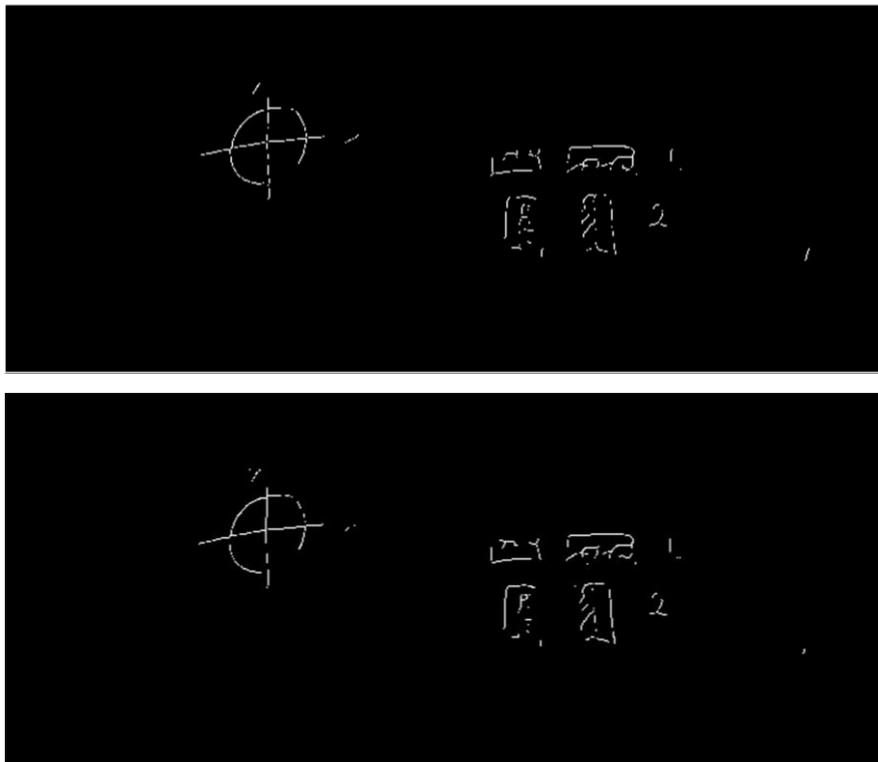


FIGURE 6.5: COMPARISON OF OUTPUT WRITING FRAMES WITH MINOR LINE MOVEMENT

Figure 6.5 shows two frames which at first glance look very similar. However multiple lines in the image have moved by a single pixel. This demonstrates why choosing the correct ground truth is so important for evaluating this system. A single pixel in the difference when comparing to the ground truth may mark that section as incorrect when the image as a whole looks almost the exact same to the average person looking at the output.

The ground truth images were drawn for any frame with major changes. Ideally there would be ground truth for every frame of the video but that is impractical in reality, especially when assessing 50-minute long videos. The ground truth was drawn using a single pixel line. For any comparisons with wider lines, the ground truth images were dilated rather than drawing new lines. The writing ground truth was drawn using the ground truth measurements to locate the board. In order to correctly compare the ground truth and output frame they were translated back into the original image space. This means the images were warped and cropped in reverse to have the correct location of the writing points.

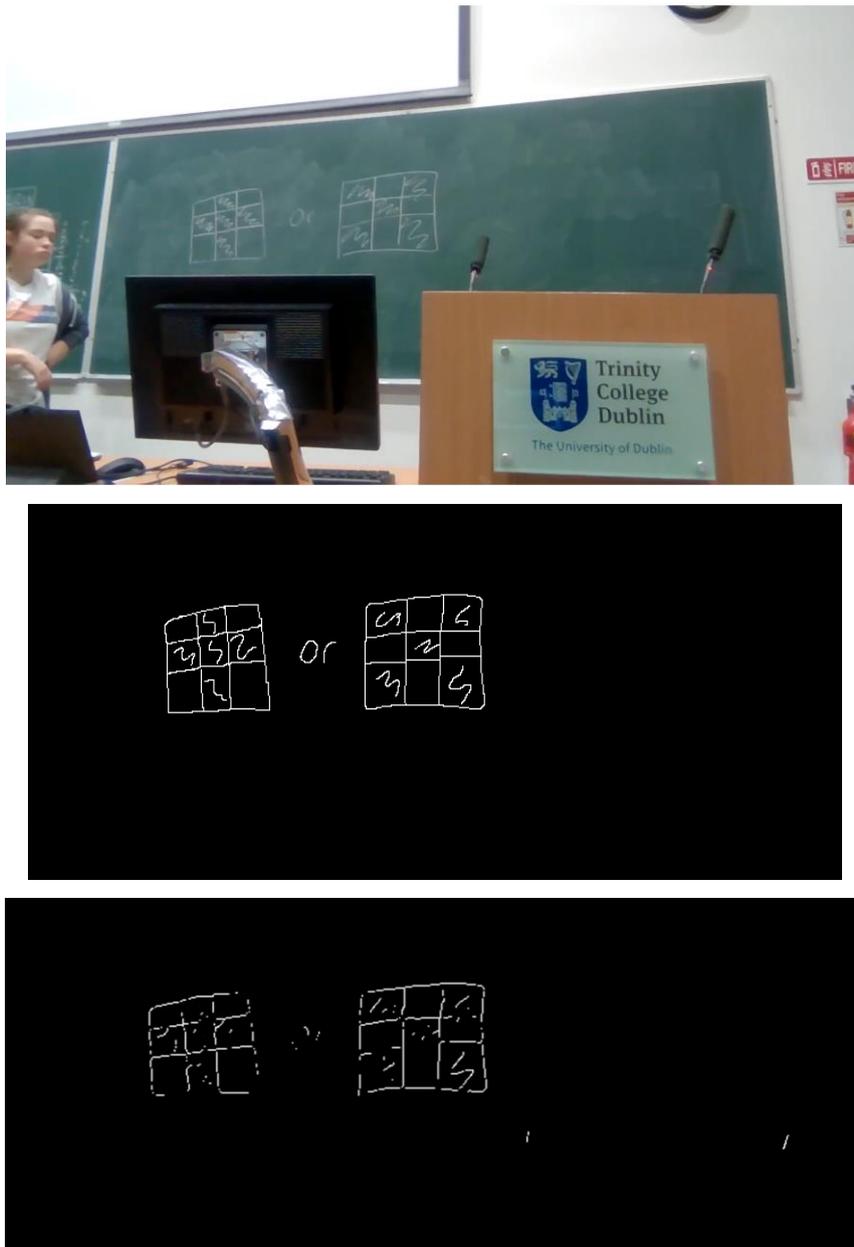


FIGURE 6.6: ORIGINAL VIDEO VERSUS GROUND TRUTH VERSUS WRITING OUTPUT

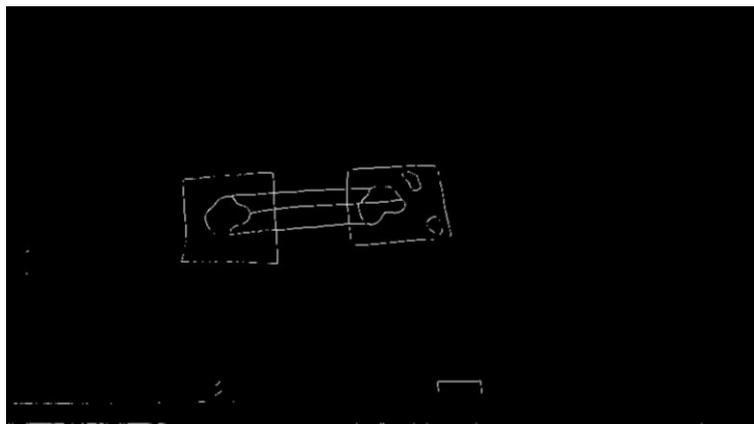
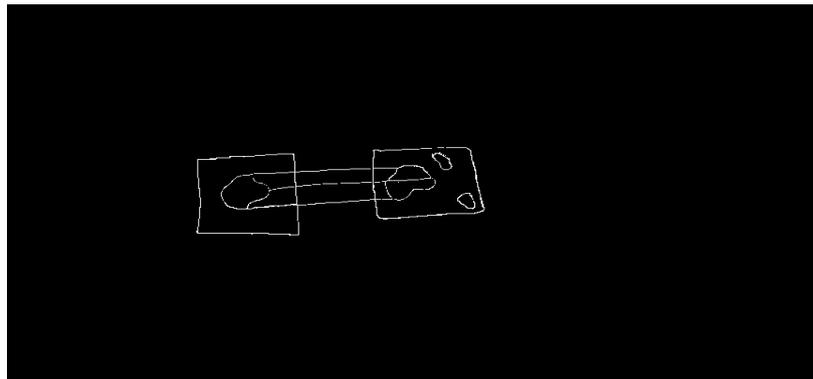
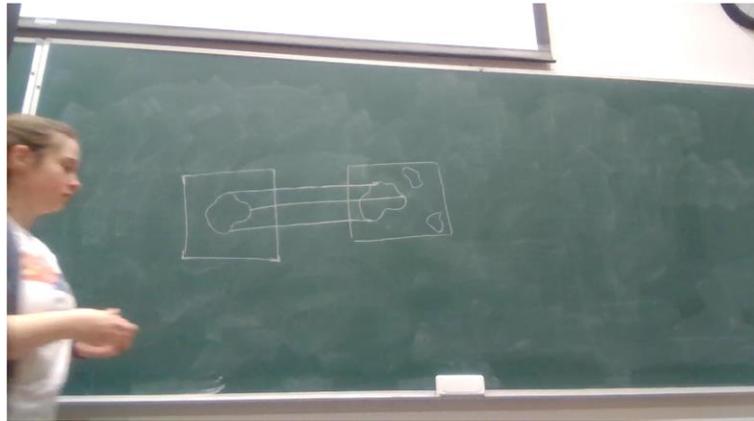


FIGURE 6.7: ORIGINAL VIDEO VERSUS GROUND TRUTH VERSUS WRITING OUTPUT

Figures 6.6 and 6.7 show examples of the writing output from the system in comparison to the ground truth drawn for the same section of video. Visually, it is easy to see what looks the same and what has been drawn incorrectly. To attempt to assess what actually is the same and what is different between the ground truth and the writing output, three different calculations were carried out. First, the percentage of output writing pixels which were correctly identified was assessed. This returns that, on average over the five videos, only 20.2% of the pixels are correct. This, however, is most likely due to the issues with the single pixel lines as discussed. When the ground truth images are dilated to incorporate more of the surrounding pixels, which may be correctly identified as writing, the percentage rises to 33.4%. This is still a very low

percentage when the output is still recognisable to the average person as very similar. If the entire image is taken into account, and not just the writing pixels, there is a much different outcome. On average over the 5 videos, just 2.1% of the pixels are different from the correct pixels in the ground truth. This figure seems to represent the similarity between the images more accurately as there is not a large difference between the ground truth and found writing. This figure however may be too low as there are some issues that occur.

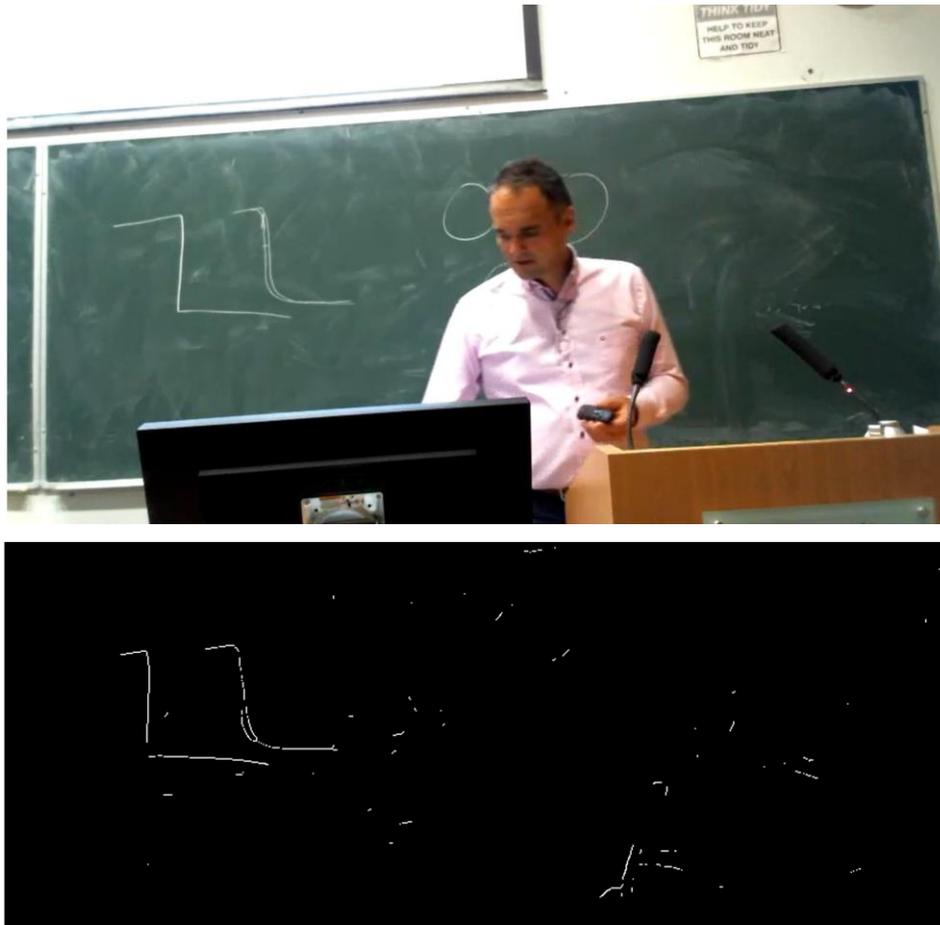


FIGURE 6.8: ORIGINAL VIDEO VERSUS FOUND WRITING SHOWING INCORRECTLY LABELLED WRITING

Figure 6.8 shows a section of video where incorrect areas are determined to be writing. In this video there are large areas of the board where previous chalk marks have been erased. These chalk marks being erased have left large white marks on the board. These marks pass every test for writing as they look like chalk marks that are faded. This causes small lines to be added across the surface of the board. This is most likely one of the reasons the percentage of pixels correctly labelled is so low.

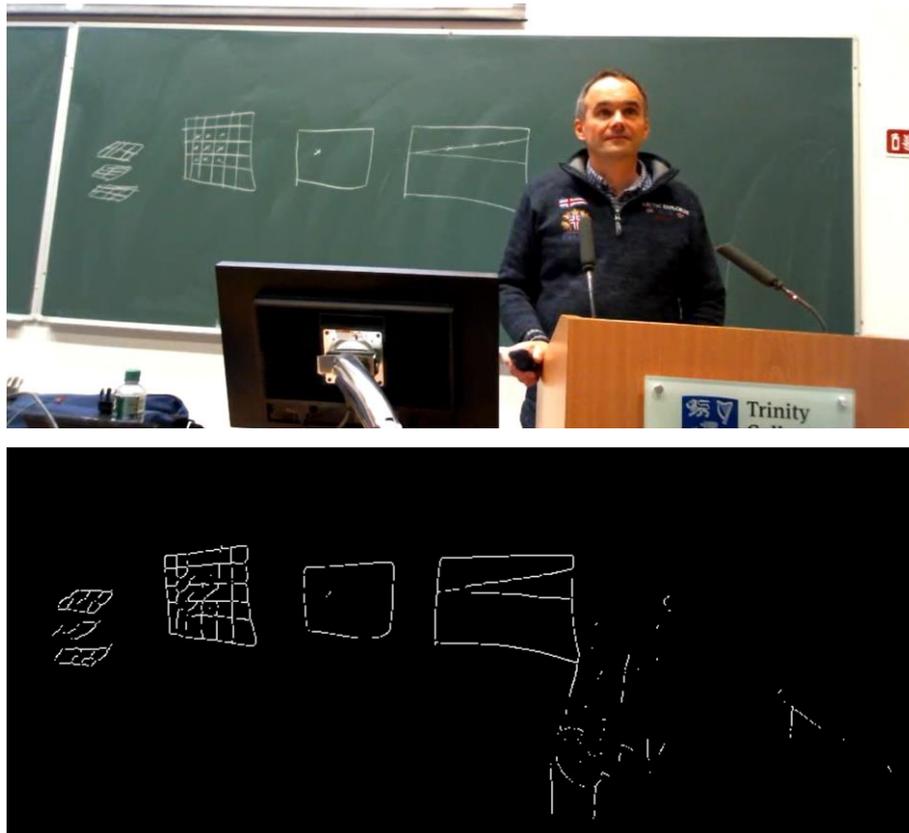


FIGURE 6.9: ORIGINAL VIDEO VERSUS FOUND WRITING SHOWING LECTURER BREAKTHROUGH

Figure 6.9 shows a frame of the video where the lecturer has remained still for long enough that their arm has been labelled as writing. For the most part, this issue has been removed by the extra validation steps. However, it still appears in some frames. The mask of the most common colour in the frame most likely has incorporated the edge of the lecturer's arm, hence allowing the lines to be labelled as writing.

The final issue the system sees is that it does not deal well with coloured in sections of diagrams. As writing has been strictly defined as lines, areas which have been coloured in are not included in the definition. Many of the coloured in areas are still found due to being surrounded by lines or the section not being fully coloured. If the section is not fully coloured in, it is possible to find lines within the coloured region. These areas cause issues with finding their surrounding lines as an accurate straight line no longer exists. The same can be said for any diagrams which have small intricate details. If details are close together with little definition, they can also cause issues with the area being detected correctly. This issue can be seen in figure 6.9 in the drawn grid. Due to crosses being added within some of the squares it is difficult to determine where the lines should actually lie.

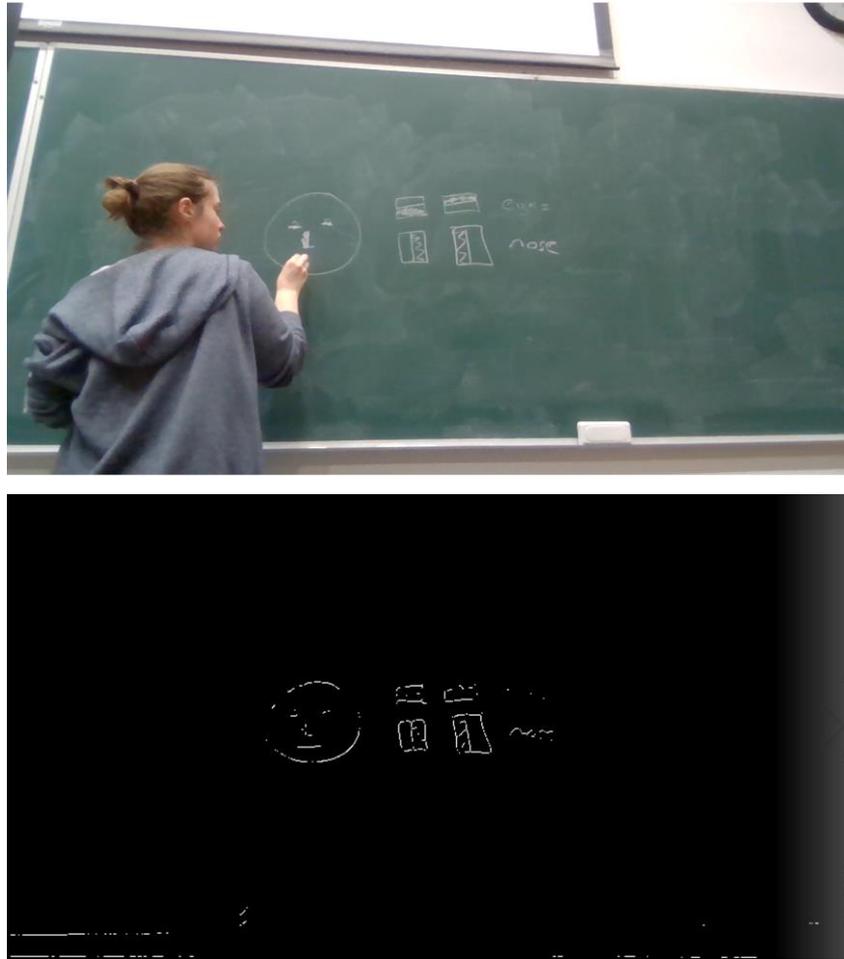


FIGURE 6.10: ORIGINAL VIDEO VERSUS FOUND WRITING SHOWING ISSUES WITH COLOURED IN SECTIONS

From looking at the outputted writing it is quite clear why this section is so difficult to evaluate. The success can be very subjective. The success of this step in the process is vital for the success of the final enhancement step as this output is used for it.

6.3 Enhancement results

Assessing the results of the enhancement output is equally as difficult as assessing the output of the writing. Creating the ground truth poses the same issues as before in regard to who should be creating it. There are also issues around the width of the lines and accuracy of the ground truth as before. These issues arise with the enhancement due to the angles of the lines drawn. Due to the points picked to draw lines, and the lines being adjusted to connect them, the lines can be drawn at slightly different angles to the ground truth. While these slight variances in angle may not make a large difference to the appearance of the output to the average person viewing the video, it may cause the line to be labelled as incorrect. The ground truth is once again drawn using the board location ground truth coordinates and translated back to the original coordinate space.

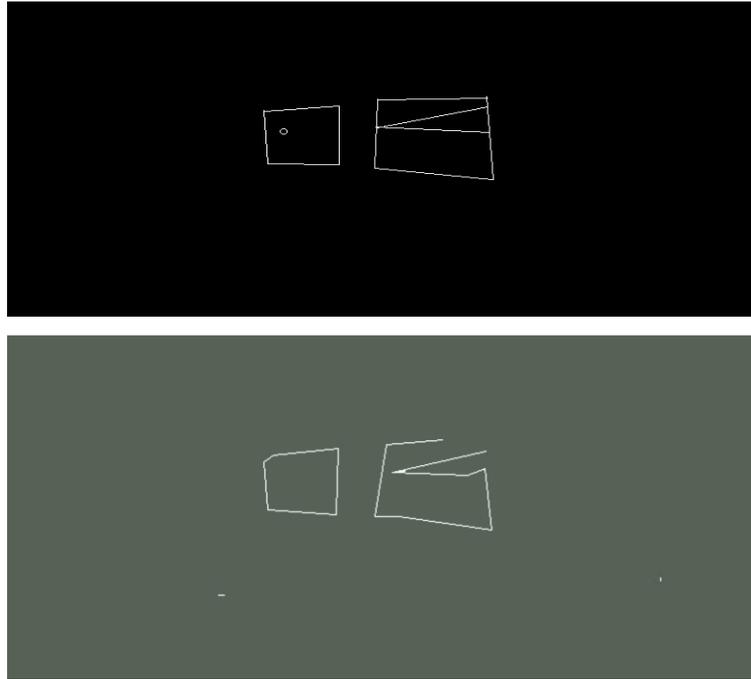


FIGURE 6.11: GROUND TRUTH VERSUS ENHANCED OUTPUT

The final output is written out onto the average colour of the frame as shown in figure 6.11. To do an accurate comparison, the ground truth was drawn on a black background and the output is processed with all pixels which are not white set to 0. This means the output is changed to have a black background rather than the average colour background for this comparison step. They are processed with a black background rather than the average background as the average background colour changes from frame to frame, thus it is easier to set everything to a static colour. An average of 40.6% of the pixels are accurately labelled in the enhancement step. This calculation is done by getting the number of correctly labelled pixels over the number of pixels drawn in the image. The ground truth image has been dilated for this calculation due to the variance in the enhanced lines found. In the overall image only 0.9% of pixels are incorrectly labelled. This calculation includes all pixels in the frame, background and writing, whereas the previous calculation only takes the writing into account.

There are shortcomings of this section of the system also. Some of these stem from the issues locating the writing in the previous step, and some stem from the process applied in the enhancement stage. When the small areas of the diagrams are incorrectly found in the writing, it is impossible for the enhancement to be accurate. However, there are some issues with these sections of the enhanced output where the writing has been accurately located. In figure 6.12 it can clearly be seen that where the writing has been incorrectly located causes an issue with the writing being enhanced. On the other hand, at the bottom of the grid where the writing

appears to have been correctly located, there are issues with the lines. This is caused by the section of the enhancement which connects the endpoints of the closest lines. An endpoint lies in the middle of a line section and two of the smaller lines have been attached to it as it is the closest endpoint to both of them. The easiest way to see where the issues have come from in the system is to view the output from each section. This provides an overall picture of why the output appears as it does.

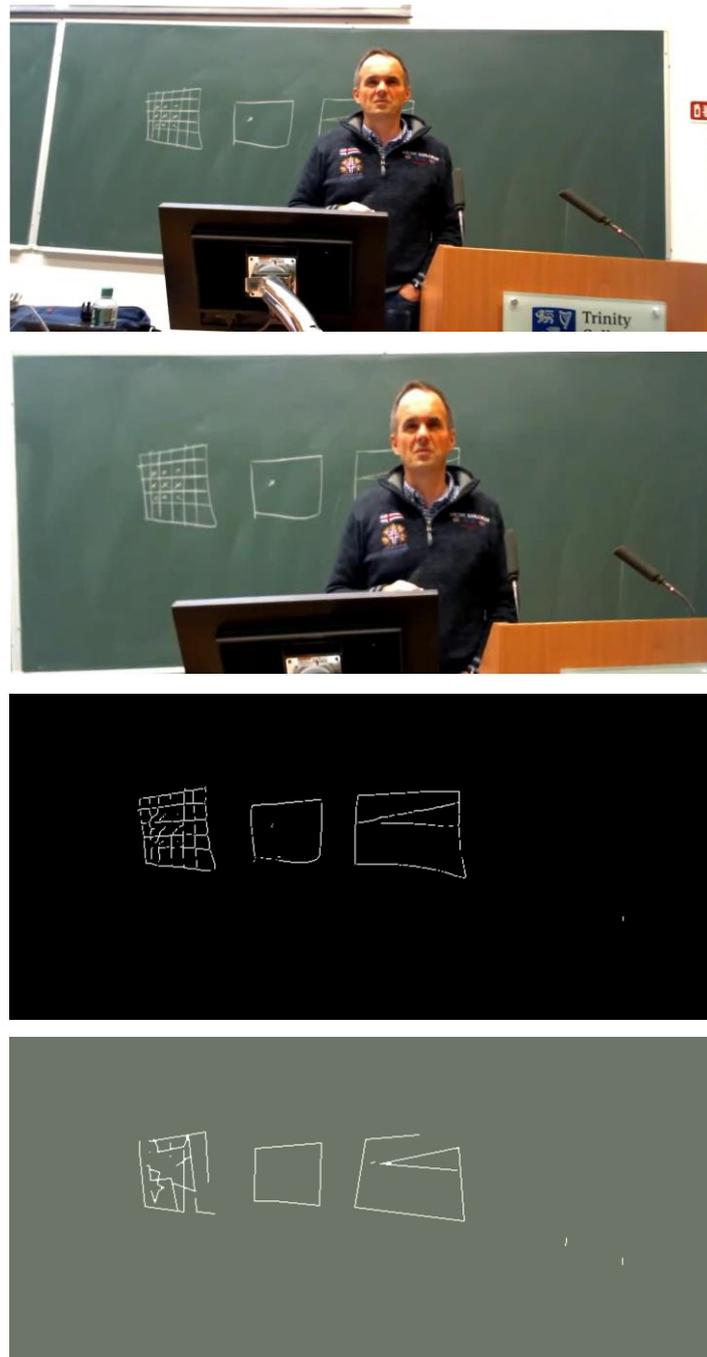


FIGURE 6.12: FOUR STAGES OF PROCESSING VIDEO; ISSUES WITH LINE CONNECTIONS

The addition of the Hough transform for circles has many positives and negatives. Circles which are otherwise shown as jagged lines are correctly drawn with the addition of Hough circles. On the other hand, there are many incorrect circles located in the video. Figure 6.13 shows Hough circles working as intended. The circle drawn on the board is correctly located and drawn alongside the lines in the image. The extra lines at the circle are left to ensure any tangents to the circle will still be correctly drawn.

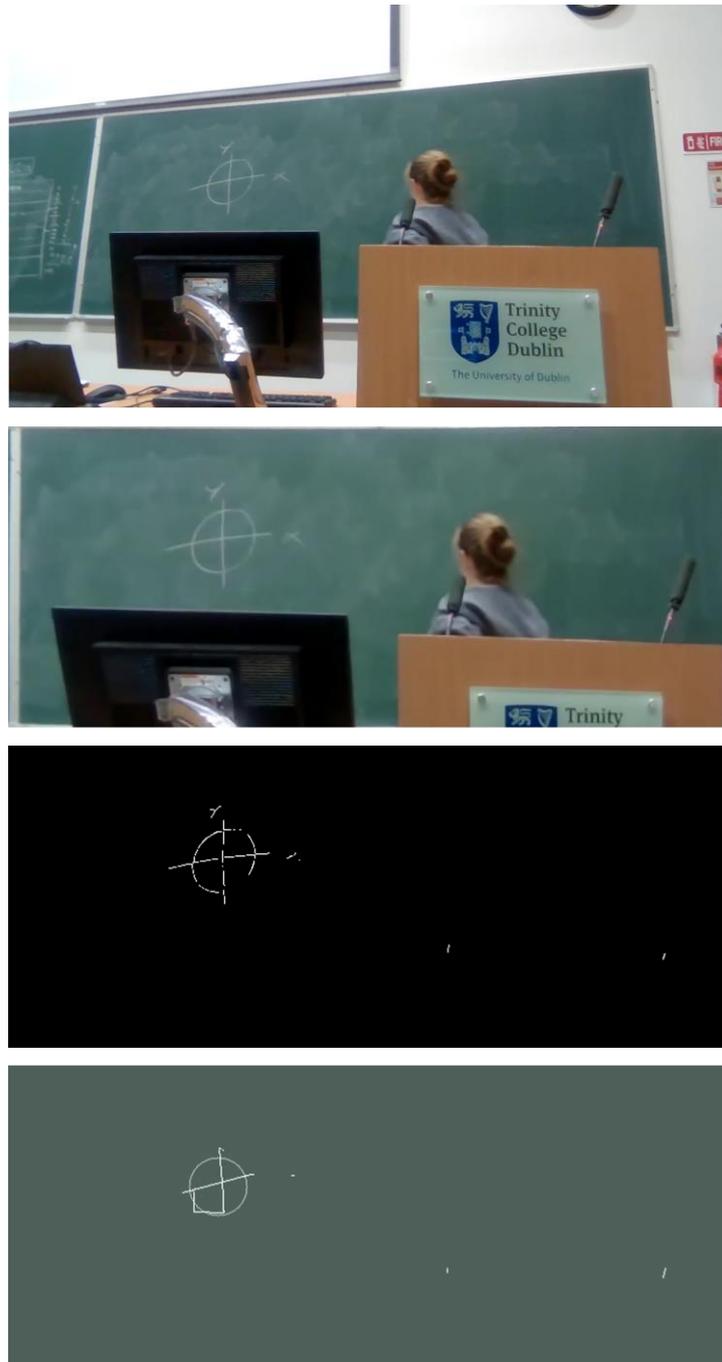


FIGURE 6.13: ENHANCEMENT USING HOUGH CIRCLES CORRECTLY

Figure 6.14 shows the addition of incorrect Hough circles to the enhancement output. These circles are drawn in the areas which have been coloured in. The inability of the writing stage of the process to correctly represent these areas leads to them being misinterpreted. The circles have been found with a high confidence rate due to the excess pixels in the writing image. Multiple crossings of circles are easily found when there is a high concentration of pixels in the one area. This is why the circles have a flag which must be set to be used and are not automatically added to every video.

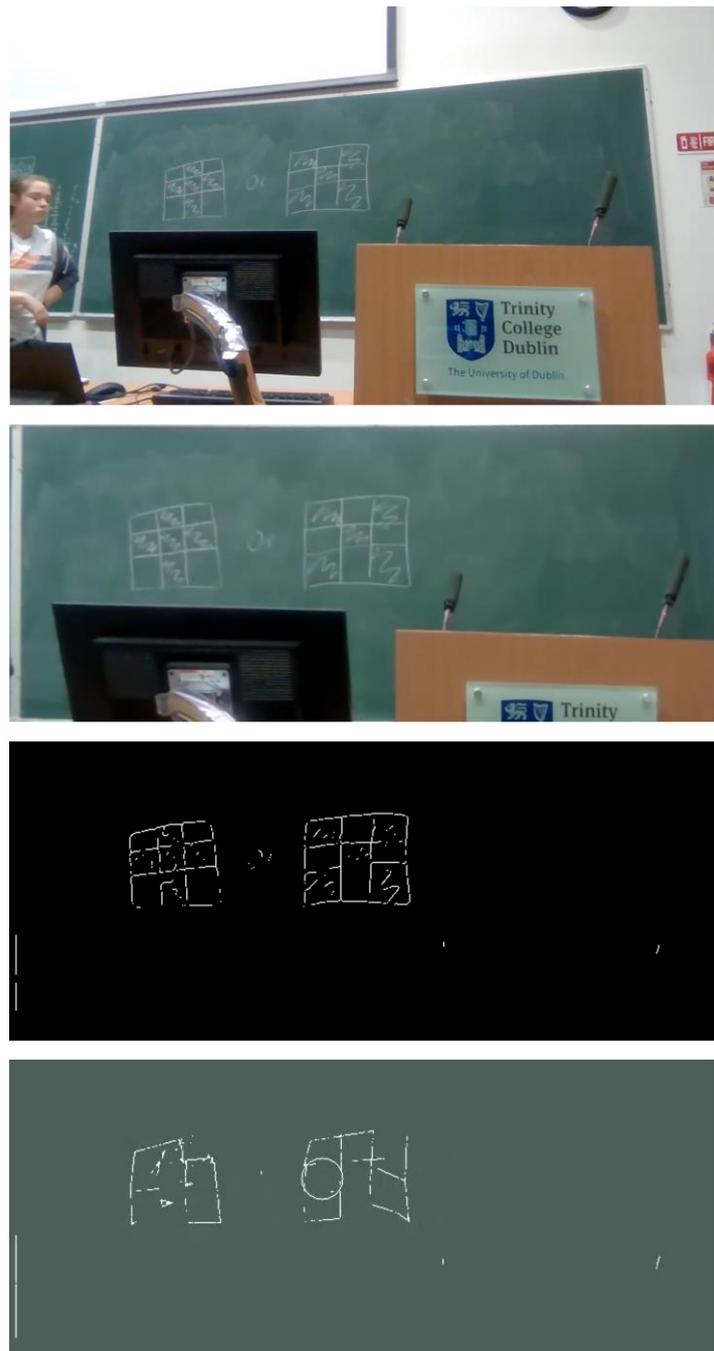


FIGURE 6.14: ENHANCEMENT WITH HOUGH CIRCLES INCORRECT OUTPUT

Overall the system has many advantages and shortcomings. The easiest way to represent these is by directly looking at the output of the system rather than specific calculations. This is due to there being many questions surrounding what constitutes correct ground truth and whether a diagram is labelled incorrect if it does not exactly match the ground truth. The numerical values calculated for the output can be argued to be inaccurate or even unimportant due to these major issues surrounding the ground truth. Any further development of the system should involve human marking of its success. A qualitative assessment of the output and how accurate, and useful it is to the students who will use it, is the next step in evaluation.

7 Future Work

The system has many successes and failures but shows a detailed proof of concept. It is possible to enhance diagrams without pre-existing knowledge of what they should look like. Locating writing in the scene is quite successful and produces recognisable results. Enhancing the lines produces results which are recognisable as the same diagrams. The use of Hough circles has multiple advantages and disadvantages. The overall assessment of the system is a difficult task, but the output images show many successful results alongside results which the enhancement has not quite worked in correctly. There are also multiple different routes that have not yet been explored that could be great additions to the system as a whole. By exploring these further additions to the system, it could produce a fully working application that could be added as a pre-processing step into PanOpto.

7.1 Accurate circle enhancement

The most obvious issue with the system is the location and drawing of circles in the enhancement stage. The use of Hough circles is not the ideal solution to this problem, as shown by the allocation of extra circles where they do not exist. Even with the tweaking of parameters and the extra verification steps to attempt to mitigate the addition of incorrect circles, there are still multiple false positives reported. It is theoretically possible to extend the current line fitting algorithm to also fit curves. Currently the lines are fit by using the two furthest points of the line segment found. A straight line is drawn between the two points. While there were some issues when attempting to use a fit line function, it may be easier to fix these issues with curves. Using the set of points found in the line drawing algorithm an attempt could be made to fit multiple polynomials of different degrees. This allows the points to also have a curve fit as well as a straight line. The best fit is determined by the line which crosses the most points. If a first-degree polynomial is the best fit, they can be drawn in the same way as previously drawn. If a curve is the best fit, the original contour used to identify the correct points can be dilated further and then a bitwise operation carried out to only display the correct amount of the curve. This can be done with the curve, but not the line, as it is more likely the curve ends will move back towards each other, whereas the line ends move outwards from the diagram and often into other diagrams. This means it is less important if the ends of the curve are longer than intended than if the line ends are longer than intended.

Another simple addition to this process of enhancement would be not to output the enhanced version of every section of the image. It is clear from the writing location that some of the

enhanced images are less accurate than the writing location images. By outputting the located writing when the enhancement fails, rather than leaving the board blank, it will produce an overall better image of the diagrams. It will include more details in the diagrams which are currently being lost in the enhancement step. This would also help in the drawing of circles and curves which appear clearly in the writing location step.

7.2 Solving the ground truth conundrum

Choosing the correct ground truth for the diagrams and determining whether the diagrams outputted are accurate to the intended diagram are a major issue. While there is no simple fix for this when testing the system, there may be a way around it within a real-world application. Once the video has been processed and is ready for upload, it should have a screening stage. This screening stage would provide the lecturer an opportunity to look at the output and approve or disapprove it. Currently the diagrams are updated throughout the video as they are drawn. With the addition of this stage it would make sense to add in a final processing step with no regard for the time stamp of the video. This step would take in the whole processed video and pick out multiple versions of each diagram drawn on the board. This could be done using similar techniques to the way in which indexing lectures is done. Key frames would be picked alongside a few similar frames to provide alternative versions of the diagram. These key frames would be presented to the lecturer post-processing and they would choose a best match for each key frame. These key frames would then be used to reprocess the video. The best fit diagrams would appear for the duration of the video where they best match the original output from the system. This would produce a better output which matches more accurately with what the lecturer had in mind when originally drawing the diagrams.

7.3 Remove all non-writing lines

Locating only lines that are the writing was a major issue and focal point in this project. One of the major aspects which took a lot of fine tuning was the choice of background model. As discussed earlier the choice of the KNN background subtraction model was not perfect. It still carries many unsolved issues. Using a background model such as codebook also does not solve all the issues. It removes the lighting changes but there are then issues with other movement going undetected. In order for this system to work perfectly, a specific background subtraction model would need to be designed. It would need to not count lighting changes as motion and never incorporate the lecturer's movement into the background but always incorporate the writing into the background when it has remained static for long enough. The writing and the lecturer often remain static for the same period of time which is what makes this task so

difficult. Creating an image of the board in its blank state without the lecturer in the scene would make it possible to remove all static lines in the video. The lecturer does not always leave the frame during the video. This image would need to be created by locating the frame in which the largest amount of the board is visible. If the lecturer is in this frame, another frame where the lecturer is in a different location, such as the opposite side of the board, would need to be added to remove the lecturer. This would solve the static lines issue, but the lecturer lines would still remain an issue, as another background subtractor which slowly incorporates static objects still needs to be used to identify writing. There is no simple solution to this. Perhaps assessing the colours being added to the background image may make it possible to stop objects which are not writing being added to the background.

7.4 Machine learning

This project is primarily designed to identify diagrams in video footage. Some diagrams contain writing and the system struggles to identify it. Many lectures consist of a mixture of both diagrams and words written on the blackboard. There is a large amount of previous research into identifying handwritten words using machine learning [14]. There are multiple open source pre-trained neural nets available which can be used out of the box to identify handwritten letters and words. As this project has been written in Python, it is possible to simply import one of these neural nets into the system. By using this technique to identify writing, letters which have been identified can be substituted with typed text, making it easier to read/interpret the writing on the blackboard. This could be expanded for many different languages and character sequences. This technique could not be used for diagrams as there is no reasonable dataset which can be used that will identify all diagrams. As there is no set output for what the diagrams can look like, it is impossible to train a neural net to recognise all diagrams. However, letters are a set shape and therefore can easily be trained.

7.5 Improving efficiency

The overall processing of videos using this system is not very efficient due to the use of many complex algorithms. Some of the algorithms have been made more efficient by making the input image smaller as to speed up processing time. There are most likely more stages in the system pipeline where this can be done without greatly reducing the accuracy of the system as a whole. However, these sections must be carefully chosen, as identifying the correct positions for the writing and enhanced writing is important for the overall output. If the wrong stage of the pipeline is adjusted, it could cause a knock-on effect to the overall output rendering it unrecognisable. This entire system has been designed as post-processing using

webcams and equipment already available to most lecturers. If specifically designed microchips could be used to add in edge processing to pre-process some stages of the pipeline it may speed up the overall processing of the video. There are multiple edge processing chips available on the market that could be used for such a system. The only issue with this is that it would require lecturers to have specific hardware to use the system rather than being open for use by anyone with a webcam and computer.

Bibliography

- [1] M. R. Edwards and M. E. Clinton, “A study exploring the impact of lecture capture availability and lecture capture usage on student attendance and attainment,” *High. Educ.*, vol. 77, no. 3, pp. 403–421, Mar. 2019.
- [2] N. Bos, C. Groeneveld, J. van Bruggen, and S. Brand-Gruwel, “The use of recorded lectures in education and the impact on lecture attendance and exam performance,” *Br. J. Educ. Technol.*, vol. 47, no. 5, pp. 906–917, Sep. 2016.
- [3] opencv dev team, “Welcome to opencv documentation! — OpenCV 2.4.13.4 documentation,” *opencv.org*, 2017. <https://docs.opencv.org/2.4/index.html> (accessed Apr. 13, 2020).
- [4] “Blackboard Learn.” <https://tcd.blackboard.com/webapps/login/> (accessed Apr. 23, 2020).
- [5] “Trinity lectures move online, DCU postpones graduations.” <https://www.rte.ie/news/coronavirus/2020/0310/1121384-education-coronavirus/> (accessed Apr. 23, 2020).
- [6] H. Yang, M. Siebert, P. Lühne, H. Sack, and C. Meinel, “Automatic lecture video indexing using video OCR technology,” in *Proceedings - 2011 IEEE International Symposium on Multimedia, ISM 2011*, 2011, pp. 111–116.
- [7] F. Wang, C.-W. Ngo, and T.-C. Pong, “Synchronization of lecture videos and electronic slides by video text analysis,” in *Proceedings of the eleventh ACM international conference on Multimedia - MULTIMEDIA '03*, 2003, p. 315.
- [8] H. Yang and C. Meinel, “Content based lecture video retrieval using speech and video text information,” *IEEE Trans. Learn. Technol.*, vol. 7, no. 2, pp. 142–154, 2014.
- [9] “Panopto Features - Secure, Searchable Online Video Platform.” <https://www.panopto.com/features/> (accessed Apr. 13, 2020).
- [10] S. E. Garber *et al.*, “A two tier approach to chalkboard video lecture summary,” in *Proceedings - Frontiers in Education Conference, FIE*, 2017, vol. 2017-Octob, pp. 1–9.
- [11] M. Onishi, M. Izumi, and K. Fukunaga, “Blackboard segmentation using video image of lecture and its applications,” *Proc. - Int. Conf. Pattern Recognit.*, vol. 15, no. 4, pp.

- 615–618, 2000.
- [12] C. C. Tappert, C. Y. Suen, and T. Wakahara, “The State of the Art in On-Line Handwriting Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 8, pp. 787–808, 1990.
- [13] D. Keysers, T. Deselaers, H. A. Rowley, L. L. Wang, and V. Carbune, “Multi-Language Online Handwriting Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1180–1194, 2017.
- [14] M. Zafar, D. Mohamad, R. O.-W. A. of Science, and undefined 2005, “On-line handwritten character recognition: An implementation of counterpropagation neural net.”
- [15] D. Song, D. Wang, and X. Chen, “Retrieving geometric information from images: the case of hand-drawn diagrams,” *Data Min. Knowl. Discov.*, vol. 31, no. 4, pp. 934–971, Jul. 2017.
- [16] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
- [17] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
- [18] J. B. Burns, A. R. Hanson, and E. M. Riseman, “Extracting Straight Lines,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 4, pp. 425–455, 1986.
- [19] N. Kiryati, Y. Eldar, and A. M. Bruckstein, “A probabilistic Hough transform,” *Pattern Recognit.*, vol. 24, no. 4, pp. 303–316, 1991.
- [20] T. Y. Zhang and C. Y. Suen, “A fast parallel algorithm for thinning digital patterns,” *Commun. ACM*, vol. 27, no. 3, pp. 236–239, Mar. 1984.
- [21] Z. Zivkovic and F. Van Der Heijden, “Efficient adaptive density estimation per image pixel for the task of background subtraction,” *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 773–780, May 2006.
- [22] K. Dawson-Howe, *A Practical Introduction to Computer Vision with OpenCV*. 2014.
- [23] C. Kimme, D. Ballard, and J. Sklansky, “Finding Circles by an Array of Accumulators,” *Commun. ACM*, vol. 18, no. 2, pp. 120–122, Feb. 1975.

- [24] T. Porter and T. Duff, “Compositing digital images,” in *Proceedings of the 11th annual conference on Computer graphics and interactive techniques - SIGGRAPH '84*, 1984, pp. 253–259.