TRINITY COLLEGE

Cardiopulmonary Resuscitation Assistant

Gavin Corkery
B.A.(Mod.) Computer Science
Final Year Project May 2018
Supervisor: Dr Kenneth Dawson-Howe

School of Computer Science and Statistics
O'Reilly Institute, Trinity College, Dublin 2, Ireland
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

______________________________________________  _______________________
Name                                              Date
Acknowledgements

I would like to thank my supervisor, Dr. Kenneth Dawson-Howe, for his continued advice throughout the project. I would like to give a special thanks to the Dalkey First Responders for their help with providing test footage and their advice. I would also like to thank my family and friends for their support throughout the project, especially those who helped to test my application.
Abstract

The goal of this project is to develop a smartphone application to analyse cardiopulmonary resuscitation in real-time using computer vision.

One of the major health concerns worldwide is out-of-hospital cardiac arrest, which has hundreds of thousands of incidences in the US alone and a survival rate of only 12%. Good quality cardiopulmonary resuscitation (CPR) is extremely important for the chance of survival after cardiac arrest.

To gain insight into the domain, this project explores the current state of CPR, and the guidelines used when training rescuers. Research into past attempts to analyse CPR using technology is also done, investigating the performance of these solutions and their associated drawbacks. This analysis of the caveats of the past solutions validates the approach taken by this project.

This project seeks to develop a computer vision algorithm to assess the quality of key areas of CPR from live video. This algorithm is realised mainly by the analysis of moving regions in the video. After developing an algorithm that performs well, this algorithm is ported to a smartphone application which is capable of providing real-time feedback on CPR quality.

The computer vision algorithm performs excellently at recognising chest compressions and artificial ventilations, recognising over 99% of compressions, and 100% of breaths from 11 test videos. The developed smartphone application performs at an acceptable frame rate and provides audiovisual feedback in a timely manner. The developed product is a viable proof-of-concept, but further work should be done to realise its full potential in the application’s domain. Several methods of achieving this are proposed.
1 Introduction

1.1 Goal

The objective of this project is to analyse the administration of CPR (cardiopulmonary resuscitation) using computer vision techniques, with the final goal of being able to use a smartphone to obtain feedback for various CPR features of interest from live video. The system should be able to process video sufficiently in real time, and provide relevant, timely and accessible feedback to the user.

1.2 Motivation

The motivation for this project stems from the high mortality rate of out-of-hospital cardiac arrest (OHCA), and relatively low bystander rate of CPR. CPR is used on victims of cardiac arrest to restore partial blood flow to the vital organs. According to the most recently available statistics for cardiac arrest [1], there were over 350,000 incidences of OHCA in the USA in 2016 alone. The rate of bystander CPR (the number of bystanders who perform CPR on the victim) was 46.1%, which is quite low. Overall, the survival rate (percentage of patients who ended up being discharged from hospital) of a victim of OHCA was 12%.

There are several reasons for the low bystander CPR rate. A study [2] surveyed 684 bystanders after cardiac arrests. Of the bystanders who did not perform CPR, the main reasons cited were panicking (37.5% of people) and a lack of confidence in their abilities (9.1%). A study into the attitudes
regarding users’ confidence and competence performing CPR before and after training was done [3], and found that the attitude changes following instructor-led and self-training courses were similar. This study surprisingly suggested that live-training does not provide any measurable advantage over self-training in terms of attitudes towards performing CPR, and that even witnessing CPR training can improve attitudes. This information suggests that an application to help with CPR training and performance could improve confidence, and thus bystander CPR rates.

The impact of bystander CPR has been well-documented. One study [4] found an associated twofold increased chance of survival among sufferers of OHCA given bystander CPR of untrained laypersons. This chance of survival further increased when CPR was performed by trained professionals rather than laypersons. Although part of this increase is due to factors outside of CPR quality, such as superior response time, this study suggests training as many people in CPR as possible to increase survival rates. Another study [5] found that there was a strong link between survival and the administration of bystander CPR, and this increased survival chance persisted even with longer ambulance response times.

Taking into account the low rates of bystander CPR, along with the increased chance of survival when a person is trained in CPR and has confidence in their abilities, it is clear that a solution to aid training may have a great impact. Given the large rates of smartphone ownership, it should be possible to leverage this fact to develop a mobile system to aid in training and administering CPR. The near-ubiquity of smartphones is essential due to one of the main factors in CPR survival rates: timing. A study into in-hospital cardiac
arrests found that the survival rate more than doubled when CPR was administered within one minute of collapse [6]. 33% of patients who underwent CPR within one minute survived to discharge, while only 14% survived after this period. This emphasises the importance of early response, which can certainly be achieved given the abundance of smartphones.

The concept of providing real-time audiovisual feedback for CPR has been studied before. A study with 63 volunteers [7] used an autonomous CPR feedback device for feedback on compression depth and rate, and compared the volunteers’ compression depth and rate with and without the feedback device. The results showed a much-reduced variance in depth and rate when using the feedback device, as well as a more than double increase in the proportion of ”correct” rates and depths. This vast improvement appears to validate the goal of this project.

It is evident that the dual goal of empowering people to perform CPR, combined with providing a tool for immediate CPR analysis, could have a great impact on bystander rates and survival rates.

1.3 Project Overview

The project consists of two phases. The first phase involves developing a computer vision algorithm to accurately measure aspects of CPR. This algorithm will be validated against test footage, which will be the basis for the ground truth. The second phase involves placing this algorithm onto a mobile application such that it can provide real-time feedback to CPR rescuers.
Figure 1: The 30-day survival rate when bystander CPR is administered compared to when it isn’t, factoring in ambulance response times [5]

2 Background

2.1 CPR

CPR (cardiopulmonary resuscitation) is a life saving technique that is used to restore partial blood flow in a person who has suffered a cardiac arrest. During cardiac arrest, the heart fails to effectively pump blood. The ultimate goal of CPR is ROSC (restoration of spontaneous circulation). Some of the most important aspects of CPR to consider are the compression rate, depth, breaths and fraction of time performing compressions. The guidelines for performing CPR are constantly under review given new evidence. This section explores the current way CPR is performed, so that an application analysing CPR will be based off the correct constraints.
2.1.1 Terminology

- **CCR** Chest Compression Rate, the rate per minute at which compressions are being performed

- **CCF** Chest Compression Fraction, the fraction of time spent performing compressions

- **AHA** American Heart Association

- **OHCA** Out of Hospital Cardiac Arrest

- **AED** Automated External Defibrillator, used to deliver electric current to the heart

2.1.2 Performing CPR

The American Heart Association has published an international consensus on the current state of CPR in 2015, and a further update in 2017[8] [9]. These reports detail the correct steps that should be done to ensure the greatest chance of survival when CPR needs to be administered.

**Compression Quality** Quality compressions are extremely important to increase chances of ROSC. Chest compressions create blood flow to the vital organs, most importantly the brain and heart. Better quality compressions lead to better blood flow. Good compressions are characterised by correct hand placement and arm pose. The heel of one hand should be placed on the lower half of the victim’s sternum and the heel of the other hand should be placed on the top of the first hand, overlapping it[10]. The chest should not
Figure 2: An example of a chest compression, showing the hand positioning and depth of compression of the rescuer. The chest should be leaned on and should be allowed to fully recoil before commencing the next compression. The elbows should remain locked throughout the movement.

**Compression Depth** The 2015 AHA guidelines instruct to perform chest compressions at a depth of at least 5cm for the average adult, avoiding compressions with depth of greater than 6cm. For infants and children, the recommended minimum depth is one-third of the anterior-posterior diameter of the chest, ranging from 4cm to 5cm depending on age. Compression depth is not easily measurable without using a device for feedback. Many CPR dummies, such as the Vimetesca Practi-Man [11] used in testing, give an audible click when the compression reaches a suitable depth for an adult or child, depending on the adjustable setting. Studies [12] have found a strong link between increased depth and survival outcomes, however the most effective depth is still an unknown.

A recent study [13] found an increased risk of complications in males...
when the compression depth exceeded 6cm. While the injuries were mostly non-fatal, the American Heart Association has taken this into account and their guidelines now reflect this by suggesting to not exceed a depth of 6cm.

**Compression Rate**  The compression rate measures the number of compressions that occur each minute. The AHA guidelines recommend a rate of between 100-120 compressions per minute [10]. Increased compression rates have been found to be associated with lower compression depth [14], hence the reason why there is an upper limit on the rate. The CCR is an extremely important aspect of CPR, as if it is too low then the blood flow will be insufficient.

**Artificial Ventilation**  When the rescuer is aptly trained, it is recommended to perform artificial ventilations on the patient. Artificial ventilations (breaths) are used to provide oxygen to the patient. When performing breaths, a common technique used is the ”head tilt chin lift” manoeuvre[15], which is the most reliable way of opening the airway. It is done by tilting the head backwards and lifting the patient’s chin with two fingers. The next step is to pinch the patient’s nose, followed by forming a seal between the rescuer and the patient’s mouth and steadily blowing into the patient’s mouth for one second. It is important that the patient’s chest rises. This rescue breath should be performed twice.

It is important to keep the time taken for artificial ventilation to a minimum. It is recommended that the time to administer two rescue breaths does not exceed 10 seconds [10].
Compressions and artificial ventilations cycles The AHA recommends compressions to be performed with artificial ventilations for people who are trained to do so. It is recommended to perform cycles of 30 compressions to 2 breaths for adults, and 15 compressions to 1 breath for children.

Compression-only CPR For laypeople or people who are trained solely in compression-only CPR, it is advised to perform compression-only CPR. It is also advised for emergency services dispatchers to give instructions for compression-only CPR for adult victims [9].

Rescuer Fatigue Due to the energy-intensive nature of CPR, there is a danger involved with the onset of rescuer fatigue during ongoing compressions. It is recommended to switch rescuers every 2 minutes [16]. A previous study [17] looked at the fall in accurate compressions over time. It found a sharp drop-off from 79.7% in the first minute to 24.9% in the second minute. However, the rescuer is often not aware of the onset of fatigue until it has
already taken place. In the study, the mean time to report fatigue was 186 seconds, which was usually well after fatigue had begun to affect compression quality. These results suggest that even the recommendation to switch every 2 minutes is too conservative, and that it is important to ensure the quality of compressions persists over time.

**Chest Compression Fraction**  The Chest Compression Fraction (CCF) of a CPR session is the fraction of time during which compressions are being actively done. The factors which will cause a lower CCF include time for artificial ventilation, interruptions, and the "pre-shock pause" that occurs before defibrillation with an AED. The AHA suggests that it is reasonable to perform CPR with as high a fraction as possible [8], with a goal of 60% or higher, however the optimal fraction is unknown. One study used game theory to verify this number[18], however the results were inconclusive.

A statistical analysis of the relation between CCF and survival [19] found that an increased CCF is independently correlated with greater survival rates. However, this study notes that there are still open questions regarding the relative importance of greater CCF during certain important moments such as directly prior and after the AED shock.

**Number of rescuers**  CPR can be done by one or two people. In one-person CPR, the compressions and artificial ventilations are performed by the same person. In two-rescuer CPR, the compressions and artificial ventilation are performed by different people. The second person is also responsible for calling the emergency services. The benefit of two-person CPR is that
Figure 4: The survival rates given different chest compression fractions [19] there is less of a delay between compressions and breaths, and it splits the workload, reducing fatigue.

### 2.2 Prior Work

The high mortality and low bystander rate of CPR have led to many previous efforts to improve the quality of CPR using technology, such as smartphones and accelerometers.

#### 2.2.1 Using Accelerometers

One popular technique to analyse live CPR is through the use of accelerometers. Since CPR involves a great amount of upward and downward movement in compressions, it should be possible to calculate a live rate and depth of
compression using the acceleration of the device.

One previous method of evaluating CPR using technology was the use of a smartphone accelerometers placed on a dummy’s chest [20]. The depth and rate of compression were both considered for the solution. The best estimates for the depth was found using an acceleration-based algorithm, which estimated the phone’s displacement "from the approximation of the linear relationship between filtered acceleration peaks". The rate was found by measuring the time between compressions from the resulting displacement waveform. The RMSE (root mean squared error) for the depth was 2.94 mm when the phone was placed under the hands, and 7.46 mm in the armband. For the rate, the RMSE was 4.53 bpm with the phone under the hands, and 4.19 bpm using the armband. This solution had issues with accidental hardware button presses during compressions when placing the smartphone under the hands, and the depth measurement error when using the armband was quite large.

Another solution used two accelerometers [21] to calculate the compression depth. One accelerometer is placed on the mannequin’s chest, while the other is placed on the ground directly next to the mannequin to act as a control for the movement of the ground. Similarly to the previous method, this solution used the concept of inertia navigation to estimate the depth. The obvious advantage of this solution is that it is more flexible in unsteady environments, such as CPR rescue at sea, aboard aircrafts or on a mattress. The depth error was 4.3mm for unstable environments, and 1.6mm on flat surfaces. The downside is that having two readily-available accelerometers in a public or home environment is not a reasonable expectation.
Figure 5: Two uses of the smartphone accelerometer solution [20]: placing on mannequins’ chest (left) or using an armband (right)

2.2.2 Using Computer Vision

Another implementation used computer vision techniques to calculate the chest compression rate using live video from a smartphone camera [22]. This solution uses an upward-facing view of the rescuer in order to determine the CCR in real time. This solution had excellent results with an acceptable rate 99.8% of the time in both low and high noise environments. Post-processing steps were added to address noise from interrupting bystanders, medium/long hair of the rescuer, and random movements. The downside of this solution is that it only analyses the compression rate, and would have difficulty with analysing depth due to the upward-facing camera view.

2.2.3 Issues with Prior Work

The above methods show that there have been successful ways to extract valuable information about CPR using technology. However, the issues with the above methods include logistical impracticality and extra costs when using specialist equipment, or only analysing one aspect of CPR. These issues
suggest the need for a solution which is both readily-available and provides more utility, which is what this solution attempts to provide.
3 Recognising CPR Using Computer Vision

The goal of the computer vision techniques used was to extract the important moving object points at each frame. This was done using dense optical flow, provided by the open-source OpenCV library [23]. The movement throughout the scene form the basis for a recognition algorithm that identifies compressions and breaths.

3.1 Dense Optical Flow

To derive the moving points from the scene, dense optical flow considers the movement done by each pixel from one frame to the next. This is done by creating a motion field (optical flow) for the entire image.

Brightness The basis for optical flow is called the brightness constancy constraint. This states that moving object points will maintain the same brightness (grey value, or intensity) over a very short period of time. The other assumption for optical flow is that pixels that neighbour each other will have a similar motion. For an RGB image, the brightness (I) of a single pixel is a weighted sum of the red, green and blue components:

\[ I = 0.299R + 0.587G + 0.114B \]

To use optical flow, the input image must first be converted from a 3-channel RGB image to a 1-channel grayscale image. Converting to grayscale uses the above formula to form an image based on the brightness of each pixel.
Calculating Pixel Displacement  By using the brightness constancy constraint, given the movement of the x and y co-ordinates of pixel (i,j) between two times (\(\Delta t\)) to be \(\Delta i\) and \(\Delta j\), the movement can be described as

\[
I(i, j, t) = I(i + \Delta i, j + \Delta j, t + \Delta t)
\]

We can describe the image after \(\Delta t\) using partial derivatives [24]

\[
I(i + \Delta i, j + \Delta j, t + \Delta t) = I(i, j, t) + \frac{\partial I}{\partial i} \Delta i + \frac{\partial I}{\partial j} \Delta j + \frac{\partial I}{\partial t} \Delta t
\]

Given that we assume the brightness before and after to be equal, this can be written as

\[
\frac{\partial I}{\partial i} \Delta i + \frac{\partial I}{\partial j} \Delta j + \frac{\partial I}{\partial t} \Delta t = 0
\]

Dividing this formula by \(\Delta t\) gives us

\[
\frac{\partial I}{\partial i} \frac{\Delta i}{\Delta t} + \frac{\partial I}{\partial j} \frac{\Delta j}{\Delta t} + \frac{\partial I}{\partial t} = 0
\]
In the above equation, the two unknown variables are \((\frac{\Delta i}{\Delta t}, \frac{\Delta j}{\Delta t})\). This is referred to as the *optical flow*, and refers to the displacement in both directions with respect to time. The other parts of the equation are image gradients, which are known. It is not trivial to solve for the optical flow due to there being two unknown variables in the equation.

There are several methods which attempt to solve for these unknown values. The implementation of dense optical flow in OpenCV uses Farneback’s algorithm [25], which estimates motion based on polynomial expansion. The polynomial expansion estimates the neighbourhood of a pixel using a polynomial. The displacement of each pixel is estimated based on knowledge of how the local neighbourhood polynomial changes upon translation.

**Dense and sparse optical flow** Optical flow can be “dense” or “sparse”. Sparse optical flow (also known as the Lucas-Kanade method) is interested in the movement of certain features of interest in an image, such as corners. These features to track are usually pre-calculated using a feature detector, such as a corner detector, or using features that are invariant to scale or orientation (SIFT features [24]). The feature points are iteratively tracked from frame to frame. Dense optical flow does no pre-computation of feature locations and instead calculates the displacement of each pixel. Dense optical flow was used in the implementation due to the fact that tracking the overall movement within the scene was sufficient due to the strong movement of CPR.

For speed purposes, the vision algorithm does not analyse every pixel, but samples at a constant rate. This is due to the observation that the
moving area will be a large, mainly contiguous area. Analysing every pixel is therefore a waste of computation as the area of interest will be greater than one pixel wide. Every 16th pixel in each row or column is analysed, leading to \( \frac{1}{16} \) of movement to analyse. This spacing works well for all tested distances, but may be too wide if the rescuer is extremely far from the camera. The assumption, however, is that the rescuer will never be so far away from the camera for this to be an issue.

### 3.1.1 Movement Thresholds

In order to eliminate some noise, the amount of movement from frame to frame must meet a certain threshold before it is considered. This reduces the impact of very small movements on the algorithm. Examining video footage showed a great amount of movement in the compressions, so a small threshold should not limit the recognition of this movement. The threshold was chosen to be quite small, requiring at least 0.5 pixels of movement between frames. This recognises the moving region of interest while reducing a lot of noise. Testing with higher thresholds, however, meant that much of the moving area was not recognised and the moving area was confined to the areas moving the most (usually the arms and hands).

### 3.2 Weighted Moving Region Modelling

In order to reduce noise from the movement of background objects, a model of the moving and non-moving regions of the scene is maintained. As the CPR session progresses, the regions with a great amount of historical movement
Figure 7: Dense optical flow showing downward movement (left) and upward movement (right)
Figure 8: The impact of thresholds: no threshold (left), threshold of 0.5 pixels (middle), threshold of 2.5 pixels (right). The middle threshold recognises most of the movement with minimal noise.

should be given a greater weighting over those that have been historically static, meaning that background movement will cause fewer false positives.

The weighting is updated at each frame for each pixel by the amount of movement from the previous frame, multiplied by a small learning rate (experimentally chosen to be .005). After several compressions, there will be a clear model of the moving regions of the image, which include the head, chest and arms of the administrator. The weighting is not updated during the breathing phase, as the movement model is focused solely on the movement during compressions.
3.3 Recognising Compressions

Having a weighted motion field for each frame forms the basis for recognising chest compressions. Through analysis of test videos, it is clear that a chest compression can be described by two phases: a strong upward motion phase and a strong downward motion phase (see figure 7). A compression is deemed to be found when strong upward movement occurs after a strong downward movement. The upward movement must be within a small window of time after the downward movement, otherwise it is attributed to noise.

3.3.1 Movement Thresholds and Ratios

Compressions are characterised by both strong upward and downward movement. In either direction, the movement must first meet a total threshold,
which is approximately equal to 25% of the scene meeting the minimum movement per pixel threshold. In reality, many of the movements are much stronger than the threshold so less than 25% of the scene will need to be moving if the movement is strong enough. To account for noisy scenes, the dominant movement must also be several times (the ratio chosen was 3) greater than the movement in the other direction. This ratio is useful in a scene with a lot of background movement. There is also a threshold for the amount of the scene that must be moving. This was chosen to be 10% to ensure that the movement takes up at least a small portion of the scene.

3.3.2 Movement Timing

In order to reduce false positives, the timing of the upward movement in relation to the downward movement is also measured. For a compression to be recognised, the upward movement must take place no longer than 1 second after the downward movement. This value was chosen to reduce false positives, however it may cause difficulties with recognising extremely slow compressions.

3.4 Recognising Breaths

The recognition of breaths uses a similar technique to the recognition of compressions. The recognition of a breath is split into two phases: the downward lateral movement and the upward movement at the end of the breath.

The lateral movement is recognised by using a threshold based on the
mean lateral movement during compressions. The total movement to the left or right must be 5 times greater than the mean movement in that direction. The lateral movement must also be at least a third of the downward movement. This movement must take place in at least 3 consecutive frames to be counted as the start of a breath. This is to ensure that the movement is actually taking place. The upward movement at the end of the breath is recognised in the same way that the end of a compression is recognised. This also acts as a mechanism for recovering from error if a breath was falsely identified.

Figure 10: Rescuer movement while starting breaths (left), during breaths (middle) and moving back into compressions (right)
Figure 11: Simplified flow of vision algorithm upon receiving a new frame

Next Frame

Calculate movement in all directions
Update weights

Upward movement?

Yes

Currently breathing?

No

Note end of breath at current time

No

Downward movement in last second?

Yes

Note compression at current time

No

Lateral movement?

No

No

Yes

Note breath start at current time
4 Mobile Application

The implementation of the aforementioned techniques is realised through a smartphone application. The landing page of the application provides the user with a view of the current scene through either the front or rear camera, depending on user selection. In training/practice situations, the smartphone should be placed at a close distance from the CPR dummy, with the CPR dummy lying between the smartphone and the rescuer. Since different smartphone cameras will have different fields of view, a precise optimal distance from the phone cannot be recommended. Ideally, the rescuer and dummy should both be clearly visible and fill a majority of the camera view.

4.1 Development

Due to the availability of access to the technology, the application is developed on the Android platform. The same OpenCV libraries used are available through the OpenCV4Android SDK[26]. The original vision code was rewritten in Java for easier integration into the Android environment and interaction with other data structures within the app, however C++ code could have been reused.

4.2 Calculating CCR and CCF

In order to provide useful feedback, the application calculates the CCR while performing compressions, and the CCF when the CPR session has concluded. The CCR is calculated using a weighted learning approach. For each compression, the calculated rate is computed as 36000 (the number of mil-
liseconds in a minute) divided by the number of milliseconds between compressions. This means that if there is half a second (500 milliseconds) between two compressions, the rate is 120 compressions per minute. The "time" of a compression is defined as the time of the first strong upward movement after downward movement.

For the first 5 compressions, the impact of each new compression is 50% of the overall rate. This allows for a quick establishment of the rate at the start of compressions. After the first 5 compressions, each new compression makes up 15% of the overall rate. The reason for this lower rate is to minimise the user’s oscillation between slow and fast compressions due to negative feedback causing vast overcorrections.

The CCF is calculated using the following formula

\[ CCF = \frac{(Totaltime - InterruptedTime)}{Totaltime} \]

In this case, the interrupted time refers to the sum of the periods of time which were at least 2 seconds long and did not contain any compressions. This will include time taken for breaths.

4.3 Realising Acceptable Frame Rate

The most important aspect to consider in an application processing live video is the resulting frame rate after processing each frame, which can be quite expensive. Using the native, 1920x1080, video results in an unusable application with a speed of approximately 1 frame per second (fps). Since the application may be processing several compressions per second, it would be
unfeasible for such a rate to give any meaningful results. It would also not be feasible to have a large delay in processing during a live situation, given the time-critical nature of CPR. To handle this, each image frame is resized to a 216x216 pixels frame, which is over 25 times smaller. This gives a massive performance boost without degradation to the computer vision algorithm. The resulting performance increase affords a frame rate of approximately 15fps, which is capable of performing real-time analysis. Previous solutions have used 15fps video [22] with good results, so 15fps appears to be acceptable for the task at hand. The smartphone used in development records footage at 30 frames per second, which would suggest that a higher frame rate is achievable with more optimisation and increased speed of future phone processors.

It was also found that the application would slow down significantly and eventually run out of usable memory due to how Java’s compiler handles OpenCV image matrices. It is therefore necessary to manually invoke garbage collection periodically and call a release function on every new matrix constructed. This will deallocate the memory used by the live image frames after they have been processed, ensuring the application does not slow down or run out of memory during use.

4.4 Methods of Feedback

As CPR administrators would be focused on the task, it is important that feedback is clear and does not require careful analysis of the phone screen. The visual feedback is as obvious and legible as possible, while audio feedback
is used to provide evaluation without having to focus on the smartphone screen. The information needed to provide metrics are processed after the previous camera frame has been processed, ensuring that feedback is always as up-to-date as possible.

4.4.1 Visual Feedback

The primary measurable metric of CPR by the application is the chest compression rate. When performing CPR compressions, this is the aspect that should be the main focus of the administrator’s attention. The region at the top of the screen is used to provide an easily-readable indicator of their current rate. The rate is shown as a large number, which is colour-coded to align with the recommended compression rate (figure 12). Rates between 100 and 120 compressions per minute are optimal, and are thus displayed with a green colour. Rates within 10 compressions per minute of being optimal are shown with orange, and rates further outside the optimal range are shown in red. The displayed rate is updated after every compression. This is done to ensure that any large rate changes can be rectified as soon as possible.

Visual feedback is also provided to tell the user to start breaths or to resume compressions (figure 13).

4.4.2 Audio Feedback

In cases where the administrator may not be able to focus their attention on the phone screen, it is preferable to have another way of delivering feedback in a timely manner. This is achieved through Android’s TextToSpeech API[27]. This API syntheses the given feedback into spoken language with
Figure 12: Colour coded rates: optimal compression rate (left), sub-optimal rate (middle), poor compression rate (right)

Figure 13: Visual cues to perform breathing (left), that breathing is happening (middle), and that breaths have gone for too long and compressions need to resume (right)
a clear voice.

The audio feedback is related to the same pertinent information dealt with by the visual feedback. During the compression phase, the application will tell the user to keep going at their current rate if they are achieving an optimal rate. If they are slightly out of the optimal range they will be told to speed up slightly or slow down slightly, and if they are very far out of the range they will be told to speed up or slow down. The frequency at which the user receives feedback on their compression rate can be altered through the settings page. This frequency can be slow, medium or fast (feedback every 6, 4 or 2 seconds respectively).

Feedback is also provided for other important aspects. If the user chooses to perform the 30:2 cycles detailed above, they can choose to be notified to start artificial ventilation every 30 compressions. Along with this, they will be notified if their breaths have taken too long, and told to continue compressions. The audio aspect of this is important, as the user would not be focused on the phone screen if they are performing artificial ventilation.

### 4.5 Viewing Summaries

At the conclusion of the CPR session, the administrator is presented with a more in-depth analysis on the summary screen.

The goal of the summary screen is to provide further insight into the user’s habits that may not be apparent from their CPR session. Examples of this could be that the user starts their compressions too slowly during each compression cycle, or that they are performing too few or too many
Figure 14: Summary after CPR compressions

compressions per cycle. Similarly to the visual feedback, the compression rate per cycle is colour-coded in the same rate ranges. The compression rate over time is graphed using the open source Android GraphView library [28], which allows for zooming to view the rate changes at a particular time. All past CPR sessions are stored locally on the smartphone through the Android SQLite database and can be viewed at any time. If the user has improved their CPR quality significantly over time, this should be identifiable through analysis of current and past CPR sessions.

4.6 Application Settings

In order to account for different situations and user preferences, there are a number of settings available to the user. The user can choose which camera
to use, whether they are performing compression-only CPR, whether they want to be informed to perform breaths every 30 compressions, and whether to use audio feedback or not. The speed of audio feedback can also be tuned based on the user’s preference. People with more CPR experience may prefer less frequent feedback, and beginners may require more frequent feedback.
5 Evaluation

The performance of the computer vision algorithm was measured on 11 pre-recorded videos of CPR being performed.

5.1 Ground Truth

The ground truth was generated from test footage obtained from the Dalkey First Responders Group. This footage consists of over 25 minutes of CPR. The pertinent information to record was the frame number of the lowest point of each compression, and the presence of breaths between sets of compressions. This information allows the derivation of metrics such as the number of compressions, rate per cycle of compressions and timing of breaths. The ground truth video footage consisted of 8 low-resolution (768x576 pixels) videos at 25 frames per second, and 3 1920x1080 videos at 30 frames per second. Different starting resolutions and frame rates allow some insight into the robustness of the vision algorithm with different cameras, however more systematic testing would be required.

5.2 Evaluation Metrics

In order to measure the results, several metrics were used. It was important to use these metrics to ensure that the system detects each compression and provides reasonable, timely feedback of the compression rate.

For compressions and breaths, the metrics used were accuracy, recall and precision. For the CCF and CCR, the values for both the ground truth and the vision algorithm were measured and compared.
Accuracy, recall and precision for compressions and breaths are defined in terms of four different classifications. These are as follows:

- **TP** True Positive: A compression or breath which occurred and was recognised by the system.

- **FP** False Positive: A compression or breath which was recognised by the system, which did not actually occur.

- **TN** True Negative: A compression or breath which did not occur and was recognised by the system as not occurring. This is not applicable to this system.

- **FN** False Negative: A compression or breath which did occur but was not recognised by the system. The compression or breath was "missed".

When performing the calculations for each test video, the following formulae are used:

\[
\text{Accuracy} = \frac{TP}{TP + FP + FN}
\]

Accuracy defines the proportion of times that the classification is correct.

\[
\text{Recall} = \frac{TP}{TP + FN}
\]

Recall defines the proportion of breaths and compressions which are correctly identified.

\[
\text{Precision} = \frac{TP}{TP + FP}
\]
Precision defines the proportion of times the classification is correct when it says a breath or compression has been identified.

5.3 Results

5.3.1 Compressions

<table>
<thead>
<tr>
<th>Video #</th>
<th>True Positives</th>
<th>False Positives</th>
<th>False Negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>144</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.1</td>
<td>127</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2.2</td>
<td>174</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>169</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>186</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5.1</td>
<td>211</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>5.2</td>
<td>211</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6.1</td>
<td>175</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6.2</td>
<td>200</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>6.3</td>
<td>170</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6.4</td>
<td>201</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Overall</td>
<td>1974</td>
<td>14</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 1: True positives, false positives and false negatives of compressions for each test video, along with the total accuracy, precision and recall.

The results for recognising compressions are very good. The vision algorithm has an accuracy, precision and recall of 0.99. The false positives mainly occur at the beginning and ending of the CPR session, where the rescuer is kneeling down and positioning their hands, or standing up and exiting the frame. This occurs due to the upward or downward movement within the scene, which is falsely attributed to compressions. This will not affect the live rate hugely, but it will affect the compression statistics on the summary page.
False negatives (missed compressions) occur in two test videos. One reason for this is a large amount of lateral movement is mistakenly assumed to be the start of artificial ventilation, and subsequent compressions are missed before the algorithm self-corrects. The other reason is that compressions in one video did not meet the movement threshold.

### 5.3.2 Breaths

<table>
<thead>
<tr>
<th>Video #</th>
<th>True Positives</th>
<th>False Positives</th>
<th>False Negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2.1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2.2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5.1</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5.2</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6.1</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6.2</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6.3</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6.4</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Overall</td>
<td>53</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.96</td>
<td>0.96</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: True positives, false positives and false negatives of breaths for each test video, along with the total accuracy, precision and recall.

The results for recognising breaths are also very good. Most notable is the fact that there are no missed breaths. If missed breaths were to occur it is possible that the overall rate would become inaccurate due to a large gap between compressions without recognising breaths in between them.
5.3.3 Chest Compression Rate

In all cases, the chest compression rate is sampled every 2 seconds from the beginning of compressions in the ground truth. This means that the results will indicate the rate at the exact same points in time to give a fair comparison. The chosen interval is 2 seconds as this is the quickest option available for audio feedback on compression rates. The metrics used were both the mean compression rate calculated throughout each test video, but also the mean difference between each calculated rate and the percentage of rates within 3 compressions per minute of each other. It is not only important for the mean rates to be close, but for the majority of rates at any point in time to be close to one another. The results were very good (table 3 and 4), with the mean rates all within 1 compression per minute of each other, and over 95% of rates in each video being within 3 compressions per minute of the real value. This suggests the calculated rate is good enough to be considered an accurate assessment.
<table>
<thead>
<tr>
<th>Video #</th>
<th>Ground Truth Mean Rate</th>
<th>Vision Mean Rate</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>108</td>
<td>108.15</td>
<td>0.15</td>
</tr>
<tr>
<td>2.1</td>
<td>103.8</td>
<td>103.9</td>
<td>0.1</td>
</tr>
<tr>
<td>2.2</td>
<td>73.95</td>
<td>73.6</td>
<td>0.35</td>
</tr>
<tr>
<td>3</td>
<td>123.27</td>
<td>122.98</td>
<td>0.29</td>
</tr>
<tr>
<td>4</td>
<td>125.55</td>
<td>125.74</td>
<td>0.19</td>
</tr>
<tr>
<td>5.1</td>
<td>115.1</td>
<td>114.28</td>
<td>0.82</td>
</tr>
<tr>
<td>5.2</td>
<td>114.98</td>
<td>114.6</td>
<td>0.38</td>
</tr>
<tr>
<td>6.1</td>
<td>92.5</td>
<td>92.5</td>
<td>0</td>
</tr>
<tr>
<td>6.2</td>
<td>92.17</td>
<td>91.6</td>
<td>0.57</td>
</tr>
<tr>
<td>6.3</td>
<td>92.84</td>
<td>92.69</td>
<td>0.15</td>
</tr>
<tr>
<td>6.4</td>
<td>92.22</td>
<td>91.59</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Table 3: Mean CCR calculated from the ground truth and vision algorithm for each video.

<table>
<thead>
<tr>
<th>Video #</th>
<th>Mean Difference Between Each Rate</th>
<th>Proportion of Rates Within 3 Per Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.54</td>
<td>1</td>
</tr>
<tr>
<td>2.1</td>
<td>0.59</td>
<td>0.95</td>
</tr>
<tr>
<td>2.2</td>
<td>0.98</td>
<td>0.93</td>
</tr>
<tr>
<td>3</td>
<td>0.78</td>
<td>0.99</td>
</tr>
<tr>
<td>4</td>
<td>0.93</td>
<td>1</td>
</tr>
<tr>
<td>5.1</td>
<td>1.42</td>
<td>0.93</td>
</tr>
<tr>
<td>5.2</td>
<td>0.7</td>
<td>0.96</td>
</tr>
<tr>
<td>6.1</td>
<td>0.49</td>
<td>1</td>
</tr>
<tr>
<td>6.2</td>
<td>0.82</td>
<td>0.95</td>
</tr>
<tr>
<td>6.3</td>
<td>0.42</td>
<td>1</td>
</tr>
<tr>
<td>6.4</td>
<td>0.85</td>
<td>0.96</td>
</tr>
</tbody>
</table>

Table 4: Mean difference between each rate calculated from the ground truth and vision algorithm for each video.

5.3.4 Chest Compression Fraction

Discrepancies in the CCF between the vision algorithm and the ground truth are mainly due to false positives at the start and end of the CPR session. These false positives will falsely indicate that the CPR session was longer than it actually was, giving a greater overall fraction. The results for the CCF
are encouraging, with most results being within 1% of the true result. In all cases, the computed CCF gives a reasonable idea of whether the compressions are being done for a suitable portion of time.

<table>
<thead>
<tr>
<th>Video #</th>
<th>Vision CCF</th>
<th>Ground Truth CCF</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.773</td>
<td>0.773</td>
<td>0</td>
</tr>
<tr>
<td>2.1</td>
<td>0.958</td>
<td>0.958</td>
<td>0</td>
</tr>
<tr>
<td>2.2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0.558</td>
<td>0.555</td>
<td>0.003</td>
</tr>
<tr>
<td>4</td>
<td>0.7</td>
<td>0.7</td>
<td>0</td>
</tr>
<tr>
<td>5.1</td>
<td>0.696</td>
<td>0.701</td>
<td>0.005</td>
</tr>
<tr>
<td>5.2</td>
<td>0.731</td>
<td>0.732</td>
<td>0.001</td>
</tr>
<tr>
<td>6.1</td>
<td>0.8</td>
<td>0.796</td>
<td>0.004</td>
</tr>
<tr>
<td>6.2</td>
<td>0.795</td>
<td>0.807</td>
<td>0.012</td>
</tr>
<tr>
<td>6.3</td>
<td>0.8</td>
<td>0.791</td>
<td>0.009</td>
</tr>
<tr>
<td>6.4</td>
<td>0.771</td>
<td>0.807</td>
<td>0.036</td>
</tr>
</tbody>
</table>

Table 5: Chest compression fraction for the ground truth and vision algorithm

5.4 Comment on Results

The results show that the system gives extremely good results. Given that there are very few false positives or false negatives for compressions and breaths, it follows that the calculated CCR and CCF will also be very close to each other. The vision algorithm is not only able to recognise breaths and compressions, but also of giving good results of the current rate at most points in time during the CPR administration. Using weighted learning for the rate diminishes the impact of one bad recognition, which leads to good results.
6 Conclusions

6.1 Vision Algorithm Results

The vision algorithm performs excellently for the given test videos, with very few false positives or false negatives. There is, however, a need to further tune certain parameters used for compression and breath recognition. These thresholds and ratios were mainly derived from analysing the total movement in the test videos and creating heuristics to describe this movement. While this led to good results for test videos and real-life use, a more robust solution may be obtainable.

More testing should also be done with noisy videos. The majority of test videos are noise-free, with the only background noise being an opening door or person walking in the background. Testing with more noisy situations would ensure robustness while also validating the use of the weighted moving object model.

6.2 Viability of Use

6.2.1 Strengths

Frame Rate The frame rate of the developed system is one of the strong points of the system. The fact that reducing the size of the input camera frame significantly does not impact results means that the application is usable. Having a lower frame rate would result in an unusable application, regardless of the quality of the vision algorithm. An even greater frame rate should be possible as phone processors improve.
Vision Results and Speed of Feedback  The vision algorithm’s excellent results, combined with the swift feedback, mean that the feedback will accurately and quickly inform the user of anything about their CPR that needs to change. Having a quick feedback loop like this is essential to the viability of the use of the app.

6.2.2 Weaknesses

Static camera  The basis of the vision algorithm utilised is that the movement takes place in a static scene. This constraint limits the flexibility of the application, as it requires the phone to be held still. This may hinder the use of the application in real-world situations where it is not possible to have a static camera, especially in open spaces or where there is nobody to assist the rescuer.

Limitation of Use  Another limitation is that the application will not perform well during two-rescuer CPR. This is due to the fact that the second rescuer is likely to obscure the camera view, and that there will not be a movement from compressions position into breathing position. Dealing with two-person CPR would entail an entirely new approach.

The application is also limited in terms of its analysis, as there are other aspects of CPR (most notably depth) which should be explored further.

6.2.3 Overview

The application that has been developed is a viable proof-of-concept and fully capable of performing the task set out: to give real-time feedback on the
CCR and breaths. It is important, however, to note that the domain of CPR extends far beyond the application’s current functionality. The application also performs under assumptions (static scene, one-person CPR) that will work well in a training environment, but does not account for the chaotic reality of sudden cardiac arrest. In developing the application further these areas must be looked at in greater detail.
7 Future Work

The developed application performs well at analysing certain aspects of live CPR in non-emergency situations. There are many areas which should be explored in order to improve the app’s utility.

7.1 Emergency Services Tools

Currently the system acts primarily as a tool for use in training purposes. In order to realise real-world utility, further work should be done to facilitate its use in an emergency scenario. This could be done by implementing features to automatically call emergency services or broadcasting GPS co-ordinates to groups who may be able to assist.

7.2 Further Analysis of CPR Characteristics

Many of the areas upon which this system could be developed are associated with application’s domain and ways in which it is used. However it is important to note that there are still areas which this system does not yet perform analysis. To extract a greater understanding of the rescuer’s CPR, the following areas should be looked at.

7.2.1 Depth Analysis

One future area to explore that is not covered by this project is the analysis of the depth of compression. The chest compression depth is an extremely important aspect of CPR. Analysing depth would involve either knowing the precise distance from the camera to the rescuer, or knowing the length of
the rescuer’s torso and calculating the depth based on the average relative movement in the scene.

7.2.2 Breathing Analysis

In its current state, the application merely takes note of the start and finish of artificial ventilation, demonstrated by lateral and upward/downward movement. The quality of the breaths is an essential feature to examine when dealing with CPR. Areas that could be explored include the correct head tilt, chin lift manoeuvre, and ensuring that the chest expands during the breathing.

7.3 Non-Static Scene

This project assumes a static scene, with no movement of the smartphone. Achieving such a scene requires either the use of a tripod, or the availability of a wall or other stable structure to prop up the smartphone. Given the time-sensitive nature of CPR, it would be preferable to be able to account for small movements in the phone from someone holding it.

A possible method for achieving this would be to leverage the smartphone’s accelerometer to detect small sideways movements and counteracting that movement accordingly. If the phone moves slightly to the left then the relative motion of the scene will appear to move slightly to the right. Knowledge of the movement of the phone could be used to account for this movement to the right and minimise it.
7.4 Features to Encourage Learning

7.4.1 Improved Insight into User’s Weak and Strong Areas

The current implementation gives good feedback during and after a single CPR session. These sessions can also be viewed retroactively. One aspect of this that could be improved is the ability of the user to identify their weak and strong points when it comes to CPR, and view their improvements over time. Having insight into this could allow the user to focus their attention on improving their weak points, and subsequently be able to view the progress they’ve made at improving these areas.

7.4.2 Gamification

One way to encourage learning is through gamification. Gamification is described as "the use of game design elements in non-game contexts"[29]. Examples of game design elements include leaderboards, badges and points. Badges could reward the user for consistency (e.g. "Train CPR for 5 days in row" or "Average an ideal rate for 8 compression cycles in a row"). Leaderboards could form the basis for a competitive aspect of performing CPR, such as with CPR response groups who perform CPR on a regular basis.

Work has been done to study the impact of CPR gamification on training [30]. The study gave students percentage scores based on their CPR with a direct feedback mannequin, and assessed their perceived competence, confidence and willingness to perform CPR with pre and post-questionnaires. The results showed a strong positive response to gamification, but suggested that care should be taken to not trivialise the issue.
7.5 Expansion to Other Platforms

One of the main motivations for this project is the ubiquity of smartphones and their integration into modern daily life. Therefore, expanding this system onto other platforms, mainly iOS, would be a great step towards increasing the number of phones capable of running the software. As of March 2018, iOS has a market share of 20.83% worldwide [31] and 43.05% in Ireland [32], with Android making up the overall majority in both cases. It is clear that expanding to iOS would give a much greater opportunity for a bystander to be able to provide assistance with their smartphone in an emergency.
8 Appendix

8.1 Attached Electronic Materials

The attached USB key contains the following data:

- All test videos used when assessing the vision algorithm
- A folder containing videos of the application in operation
- C++ code used for assessing the test videos. This was run in Visual Studio 2017 during development.
- The main Android code of importance
- A README file discussing the present code
References


