INDUSTRIAL TRAINING REPORT

TRAINING ORGANIZATION : LIVING ANALYTICS RESEARCH CENTER

PERIOD OF TRAINING : FROM 25/02/2019 TO 26/07/2019

FIELD OF SPECIALIZATION : COMPUTER ENGINEERING

G. C. JAYATILAKA

E/14/158

ACKNOWLEDGMENTS

The success I achieved during my training was not a solo effort by any means. Hence, it is with great pleasure, that I acknowledge and extend my gratitude to those who contributed towards the success of my internship.

I would like to extend my heartfelt gratitude to Professor Archan Misra, the Dean of Research at the School of Information Systems, Singapore Management University, my primary supervisor and mentor throughout the period of the internship. He was the driving force behind my research project, constantly guiding, advising and motivating me to do my best.

Next, I would like to thank Mr. Vu Tran, a 5th year PhD student attached to LiveLabs Urban Lifestyle Innovation platform, for his support in making the project a success.

Further, I would like to thank my other advisor, Assistant Professor Ashwin Ashok, from Georgia State University, for his immense support and invaluable knowledge he imparted on me.

Ms. Lydia Tan, the administrative executive at LARC and Mr. Desmod Yap, the computer engineer at LARC deserves a thank you for handling the professional, financial, legal, technical matters so that I could focus my time and energy on research without having to worry about anything else.

I am grateful to my research group (mobile systems group), rest of the LARC staff including few researchers from Sri Lanka for making my stay a pleasant one.

Last but not least, I would like to thank my family and friends in Sri Lanka for cheering me up during my long stay in foreign land.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

BER	Bit error rate
CNN	Convolution neural network
DNN	Deep neural network
F_C	Camera frame
F_D	Display frame
IoU	Intersection over union
LED	Light emitting diode
LSTM	Long-term short-term memory
LARC	Living Analytics Research Center
ML	Machine learning
MOS	Mean opinion score
RC	Camera frame rate
RD	Display frame tare
RGB	Red green blue
SCC	Screen to camera communication
SIS	School of Information Systems
SMU	Singapore Management University
VLC	Visible light communication

Chapter 1

INTRODUCTION

1. 1. TRAINING SESSION

The training establishment that I underwent training with respect to the TR400 Industrial Training module was Living Analytics Research Center (LARC), which is situated inside and affiliated with the School of Information Systems (SIS), Singapore Management University (SMU), 80 Stamford Road Singapore 178902. My training period commenced on 25th February 2019 and terminated on 26th July 2019, extending throughout a duration of 22 weeks.

1.2. INTRODUCTION TO THE TRAINING ORGANIZATION

1.2.1 Singapore Management University

The iconic Singapore Management University was initiated in 1997 by the Singapore government, pioneered by the then Deputy Prime Minister Dr. Tony Tan. Being the third university in Singapore in the post-Lee Kuan Yew era, Singapore Management University was set to adopt the American learning environment, rather than the stereo-typical British learning environment, becoming the first of its kind to do so.

In 2000, the initial discussions of this new university became a reality, under the patronage of Professor Janice Bellace, located on the Evans Road, Singapore and it was famously known as the “Bukit Timah Campus”. Subsequently, in 2005, Singapore Management University was shifted to its current home in the Bras Basah District, around the Stamford Road and the Bugis area of Singapore. Presently, the university is thriving at the prime of its success, being a top-ranked university in Asia, under the leadership of its current president Professor Arnoud De Meyer. Figure 1.1 below, illustrates the logo of the Singapore Management University, symbolizing the iconic lion tangram whereas Figure 1.2 illustrates a magnificent view of the Singapore Management University.

Figure 1.1 SMU Logo

At present, the university comprises of six faculties, namely, School of Information Systems, School of Accountancy, Lee Kong Chian School of Business, School of Economics, School of Social Sciences and School of Law. It houses 7,827 full-time undergraduate students, 1,750 full-time and part-time graduate students, complemented by over 19,000 undergraduate and postgraduate alumni. Singapore Management University is widely known for its diversity, with 61% of its graduate students coming from 44 different countries and 10% of its undergraduate students coming from 28 different countries. Currently, 353 full-time staff mentor these students, with 44% of the staff coming from 30 different countries.

Figure 1.2. Singapore Management University

1.1.2. School of Information Systems

One of the strongest faculties of the Singapore Management University is the School of Information Systems. Being nearly as old as the university itself, the school houses 1,653 undergraduate students and 82 graduate students, as well as 50 members of the faculty and over 100 research staff members. The school offers two majors for undergraduates pursuing the Bachelor of Science in Information Systems, Smart city management and technology major and Information systems major. The school will start in Computer Science major in 2019.

SIS is a five-storied building allocated for the school of Information Systems. There are over 11 research labs and centers affiliated with this school, including the LiveLabs Urban Lifestyle Innovation platform,

Living Analytics Research Centre (LARC) and Urban Computing and Engineering Lab.

1.1.3 Living Analytics Research Center

This is a laboratory occupying the fifth floor of SIS. The laboratory is funded by National Research Fund. The lab is steered jointly by SMU and Carnegie Mellon University. The lab hosts around 40 researchers in total. The focus areas are

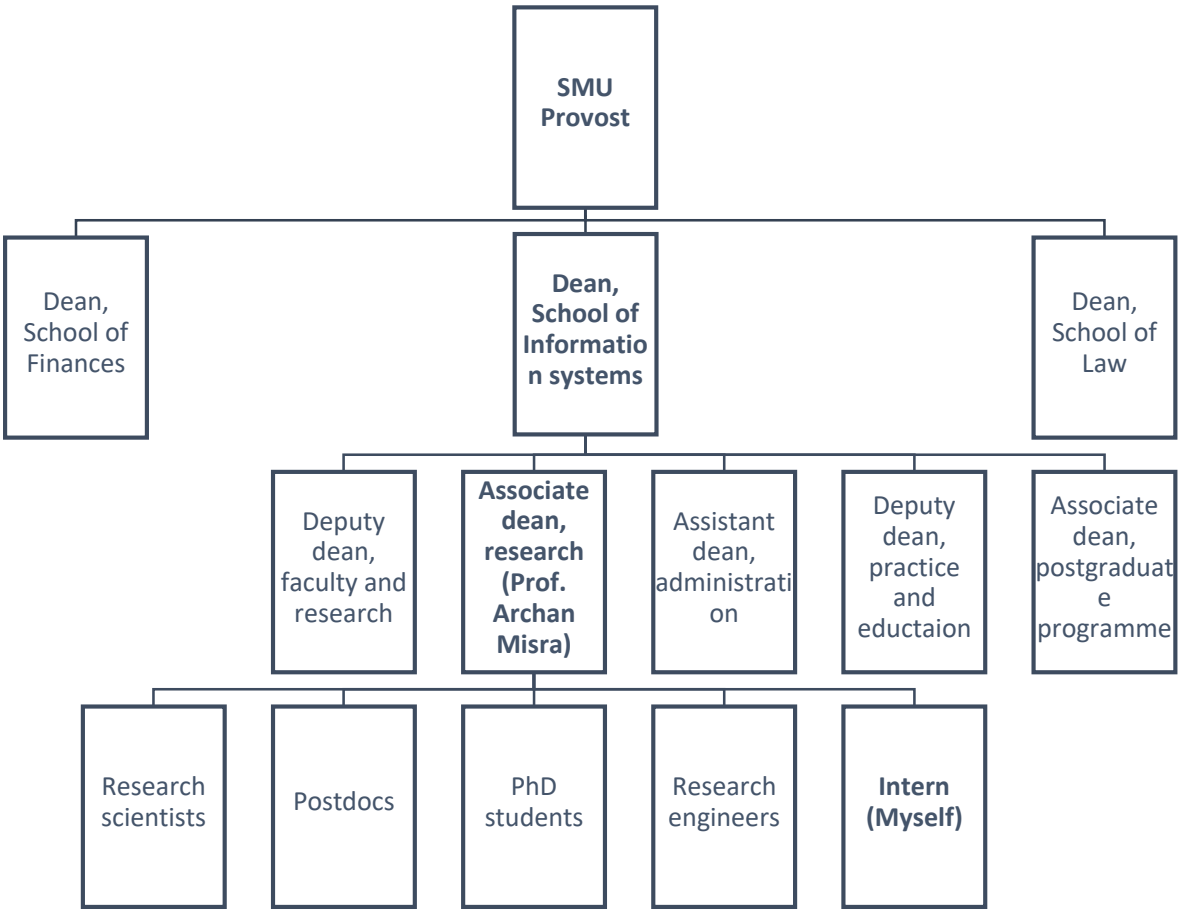

Figure 1.3. SMU organization structure (Hierarchy of reporting officers)

1.3. SUMMARY OF TRAINING EXPOSURE

SMU requires interns to work on a research project. I started working on a project called **"DeepLight"** which aims to develop a VLC (visible light communication) technology with SCC (screen to camera communication).

The training period was spent on learning about existing VLC, SCC techniques, understanding the working principles and theory behind those implementations, assessing the shortcomings of those systems and developing the new SCC technique. To the latter part of the training, I worked on evaluating

and benchmarking the system. Finally, I put my time to write a research paper (along with the rest of the group) on our project.

Chapter 2

OVERVIEW, BACKGROUND AND RELATED WORK

2.1. INTRODUCTION

The first part of any scientific research is a study of the background and the previous work on the topic. Several weeks of my internship was spent on reading through literature to improve my knowledge on these matters.

The rapid growth of camera-embedded personal devices, together with the widespread deployment of digital displays, opens up opportunities for new forms of vision-driven, human-computer interactions. This paper tackles the problem of screen camera communication (SCC), whereby displays embed information (in a humanly imperceptible fashion) in their screen content, which is then de-coded by the camera sensor. We show that state-of-the-art SCC techniques are not robust enough for practical, real-world use as their performance degrades significantly when confronted with several real-world shortcomings in the decoding pipeline–such as the inability to precisely delineate the screen boundaries or filter out captured images that have spurious coded information. Inspired by the recent success of DNNs in tackling vision problems, we then develop DeepLight, an approach that overcomes these shortcomings via application of multiple, specialized DNN pipelines. Experimental evaluations show that this DNN-based approach indeed presents a simpler and robust solution, providing improved throughput while ensuring that the information remains visually imperceptible.

2.2. OVERVIEW OF DEEPLIGHT

Screen-camera communication (SCC), where information encoded into the images displayed on commodity screens are decoded via analysis of the images captured by a camera-sensor, have recently received strong interest, and can enable several interesting pervasive applications based on out-of-band communication. In contrast to pure visible light communication (VLC), SCC's key challenge is to enhance VLC communication throughput while simultaneously ensuring that any such visual encoding remains *imperceptible* while being applied to *unregulated* and dynamic video content displayed on regular displays. We observe that, while recently-proposed SCC techniques can achieve very high communication rates under controlled conditions, they are **not robust enough** to operate under real-world scenarios with *typical human mobility and usage-based artifacts*.

This project, **DeepLight** is a novel screen-camera communication system that employs machine learning (ML) based inferencing techniques in the decoding pipeline to achieve a degree of robustness that has

hitherto been unattainable}. We claim that DeepLight is the first system to realize the promise of SCC techniques for real-world scenarios, where:

(a) an individual user may direct the camera sensor towards the screen intermittently, at random instants (e.g., for a smart glass-based camera, the camera captures the screen only when the user gazes at the screen) and with different viewing angles,

(b) the screen content may exhibit a variety of dynamics (ranging from predominantly still images to movie-like videos and advertisements),

(c) the screens are located in environments with different ambient lighting levels and backgrounds (e.g., large fast-moving crowds vs. walls) and have different attributes (e.g., aspect ratios, frame refresh rates) that are *a priori* unknown to the mobile camera. Each of these real-world characteristics effectively introduces noise in the signal captured by the camera sensor—the challenge then is to devise a decoding scheme that is robust to such noise and can yet recover encoded information that remains imperceptible to human eyes.

To motivate the need and design goals for *DeepLight*, we first evaluate several state-of-the-art proposed techniques and show that they suffer from several key drawbacks:

- **Vulnerability to Imprecise Screen Detection:** Current SCC techniques encode individual information bits into distinct spatial grids of the screen and assume that the receiver camera is capable of perfectly extracting the screen's border, a necessary step in determining the pixels associated with each bit. Not surprisingly, most experimental results presented in prior work are either performed using static cameras (precalibrated with the screen boundaries) or by using special markers to delineate the screen edges. We shall show that the real-world artifacts (such as dynamic backgrounds or changes in viewing angles caused by observer motion) cause currently employed screen detection techniques to often perform very poorly-----Intersection-over-Union (IoU) values are as low as 62%--- thereby resulting in a cascaded failure of the proposed decoding techniques.

- **High Flicker Perception at Low Frame Rates:** Some of the proposed techniques involve modulation of the pixel intensity, with each grid/bit of the image being subject to either a fractional increase or decrease in the intensity, along all 3 (RGB) channels. While such intensity changes are less perceptible at high frame rates—e.g., we experimentally found an increase of 2% intensity to be not noticeable at screen rates in excess of 120 Hz--we found that even relatively modest changes become highly perceptible in displays with lower frame rates (public displays currently typically operate with screen refresh rates of 30 or 60 Hz).

- **Need for Explicit Screen Markers & Knowledge of Display Rates:** Recent techniques for high bit-rate SCC often make implicit assumptions on tacit knowledge being available to the decoding camera sensor. For example, accurate decoding requires the camera to identify and filter out *transition* frames (those that capture, due to the rolling shutter effect, a mix of two consecutively coded images): approaches such as assume the presence of specific markers in each display frame that help in identifying such mixed `boundary' frames. Similarly, to recover individual bits when the bit is encoded across multiple consecutive display frames (e.g., due to the use of Manchester coding techniques), the camera is assumed to be already aware of the precise display rate. In real-world environments, where different venues/displays may operate independently, it is infeasible to assume that the decoding camera sensor is aware of such 'configuration parameters.

We shall show, through a careful set of design choices and systematic experimentation, that DeepLight is able to overcome each of these limitations and challenges. The key to the elimination of the need for such prior information or noise-free sensing lies in DeepLight's use of multiple ML-based inferencing pipelines, each of which learns the salient latent features from a diverse set of labeled training images. Our choice of ML-based mechanisms is motivated by the recent dramatic successes reported in the application of Deep Neural Network (DNN) models to various vision-based problems, including object detection and object localization, *especially in the presence of noise*. More specifically, \name utilizes distinct DNN pipelines to address different challenges including:

(a) Accurate Display Screen Detection—this is performed by using a modified object-detection DNN, which learns and exploits the pixel intensity variations that are characteristic of screen content;

(b) Accurate Inference of Non-Transition Frames—this is performed by a modified LSTM-like DNN that takes multiple consecutive camera images as input and determines the best representative non-transition images, without any knowledge of the underlying frame rate or without needing explicit embedded markers; and

(c) Accurate Bit Decoding without Perfect Spatial Alignment—-this is performed by a low-complexity Convolutional DNN that to recover individual code bits without being explicitly aware of the grid of pixels corresponding to each bit. Besides describing each of these individual components, we shall provide a real-world empirical evaluation of the robust operational of the overall DeepLight system.

2.3. KEY CONTRIBUTIONS

Via a detailed description of the individual DeepLight components and an overall evaluation of the system, we make the following key contributions:

- **DNN-based Inferencing for Accurate Screen Detection:** We demonstrate a DNN-based pipeline, which utilizes multiple images captured by a camera, to accurately identify the coordinates of the display screen without assuming any external markers or a-priori knowledge of display dimensions or shape. The *ScreeNNet* pipeline effectively operates over multiple consecutive images to extract out the low-level intensity changes that are typical of encoded content displayed on a screen, thereby isolating the screen *object* from another ambient/background content. Experimental studies, across a variety of representative indoor public environments, shows that ScreeNNet has an average IoU value of 91%, in contrast to prior "edge detection" based detectors that perform extremely poorly (average IoU=62%).
- **DNN-based Identification of Non-Transition Representative Frames:** We develop a novel processing pipeline, which takes multiple consecutive camera images as input, and then filters out non-transition frames from the subsequent decoding process.

  Typically, a pair of frames, corresponding to the use of Manchester Coding, where an information bit is coded over two distinct display frames) for subsequent decoding. This pipeline, called *TransitionNet*, effectively uses a DNN to identify only those frame sequences that contain a valid pair of Manchester-encoded frames---note that this is achieved without requiring any explicit in-frame markers or any offline knowledge of the screen's display rate.
- **Collective Bit Decoding without Perfect Screen Alignment:** While Screen-ML is better at extracting the screen portion of a captured image, it is not perfect: the inferred screen portion may either be smaller or larger than the actual screen, and may also have offsets in both (x,y) dimensions. To perform reliable decoding under such noisy screen detection, we first develop *DLNet*, a convolutional DNN pipeline that infers a collective set of information bits without explicit knowledge of the spatial delimitation of each bit's spatial grid, and develop *LightNet*, a reduced complexity model that takes advantage of the temporal information across frames. Through extensive experiments, we show that LightNet is able to achieve bit error rates (BER) of 7% (an improvement of almost 10% over prior techniques that perform explicit spatial partitioning) , across a variety of static and dynamic screen content, even when each bit is encoded via only a largely-imperceptible 2% change in pixel intensity.
- **Robust, High Bit Rate SCC under Real-World Artifacts:** After describing each individual component, we present and evaluate the entire Deeplight pipeline for screen-camera communication under realistic human motion, background/ambient artifacts and without explicit

use of screen parameters (such as display rate or markers). To make such intensity changes more imperceptible, \name adopts an approach of selectively modified only the Blue channel of a screen's RGB content, which is known to have the least visual sensitivity. Experimental results, using Reed-Solomon block codes, show that Deeplight can achieve a goodput of approximately 8 Kbps, with a distortion/flickering perception score of 3.84 (mean opinion score), an improvement of over 200% of state-of-the-art techniques such as HiLight (4Kbps) and Chromacode (4Kbps).

We emphasize again that Deeplight is a system that builds on, instead of competing with, prior core SCC technologies such as HiLight and ChromaCode. Deeplight is notable for selectively introducing DNN pipelines into the decoding technique, and for demonstrating that such DNN-based approaches are more robust than pure signal processing-based approaches under real world noise and impairments.

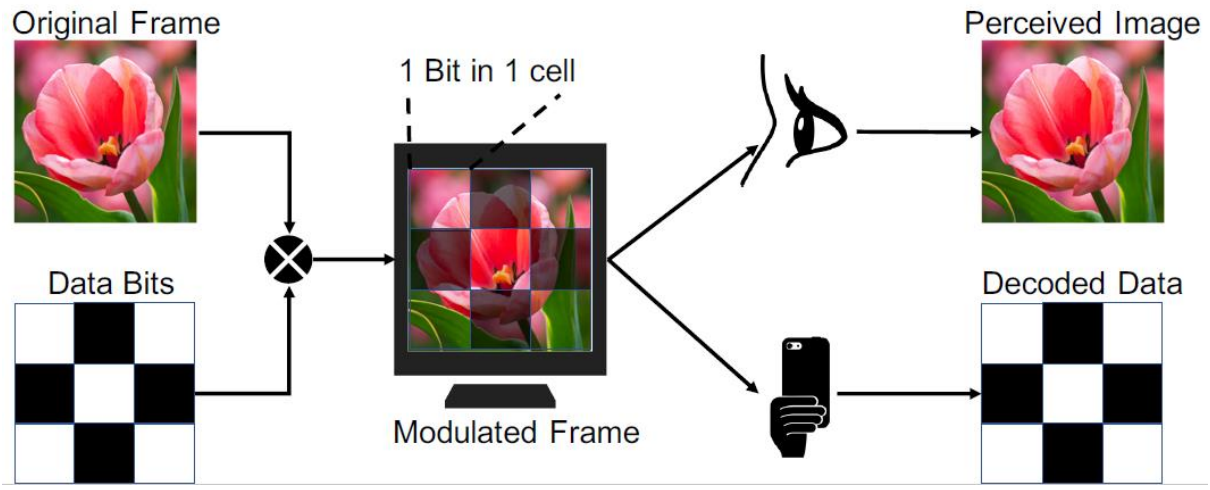

Figure 2.1 Conceptual illustration of Screen-Camera Communication (SCC)

2.4. BACKGROUND OF SCREEN CAMERA COMMUNICATION

Screen-Camera communication (SCC) is an emerging variant of visible light communication (VLC). In pure VLC, there is often a single light source (e.g., an LED bulb) and the aim is to ensure no humanly-perceptible change in the *ambient* illumination. In contrast, SCC involves a large multi-pixel screen (hence, potentially tens of thousands of independently controllable light sources (LEDs), with potentially multiple encodable objects). Moreover, unlike a photodiode receiver that can measure only an aggregated light intensity, a camera sensor can record the luminance of different parts of the screen independently. Finally, because the light source is a display screen that is directly viewed by a human

observer (e.g., an advertising display in a mall), it is important that the display content itself (and not just the ambient illumination changes) remain below the threshold of human perception.

The basic idea of SCC is derived from the concept of digital watermarking and is illustrated in Figure 2.1. At the display screen or *transmitter* end, the set of image pixels is logically divided into rectangular grids, where each grid represents one encoded bit. Consider an image of resolution rows x cols pixels, divided into equal-sized rectangular grids of NxM.

By appropriately modifying the attributes (e.g. pixel intensity) of the constituent pixels, information can be *encoded* or *embedded* into each grid. The pixel intensity variation can be as simple as adding (representing a `1') or subtracting (representing a `0') a Δ value from the intensity value of each pixel in a specific grid. The displayed content becomes a carrier of NxM bits of information. Of course, the embedding process must ensure that while each grid carries information, the modifications to the original pixels remain “hidden”--i.e., are imperceptible to the human eye. However, the camera (i.e., the *receiver* device) is able to decode the original embedded information from its captured images. While watermarking systems involve the lossless (e.g., file transfer) of the encoded content, SCC differs in use of a camera sensor to capture a potentially-modified image, with the decoding applied on this *camera-captured* image. The use of a camera sensor in the real world introduces distortive effects such as non-linear camera optics, effects of ambient lighting and motion artifacts (caused by the mobility of the camera-embedded device).

If each pixel is modulated with a +-Δ, it is possible to communicate rows x cols number of bits per frame. Considering the high-resolution capability of current day cameras, SCC can theoretically communicate 10s of Megabits/frame, thereby achieving throughput values of tens-hundreds Mbps with even 30fps video frame rates. Accurately determining the change in an *individual* pixel's value (without any a-priori knowledge of the image content) is, of course, effectively impossible--accordingly, each bit is encoded across multiple bits, with a proportionate reduction in throughput. While decoding accuracy can then be increased by use of a larger Δ, this leads to greater per-pixel/grid distortion, resulting in the human eye perceiving a flickering chessboard-like pattern. Thus, the central challenge in SCC is to achieve higher throughput (without the grid size becoming too small to permit accurate decoding) while avoiding perceptions of flicker.

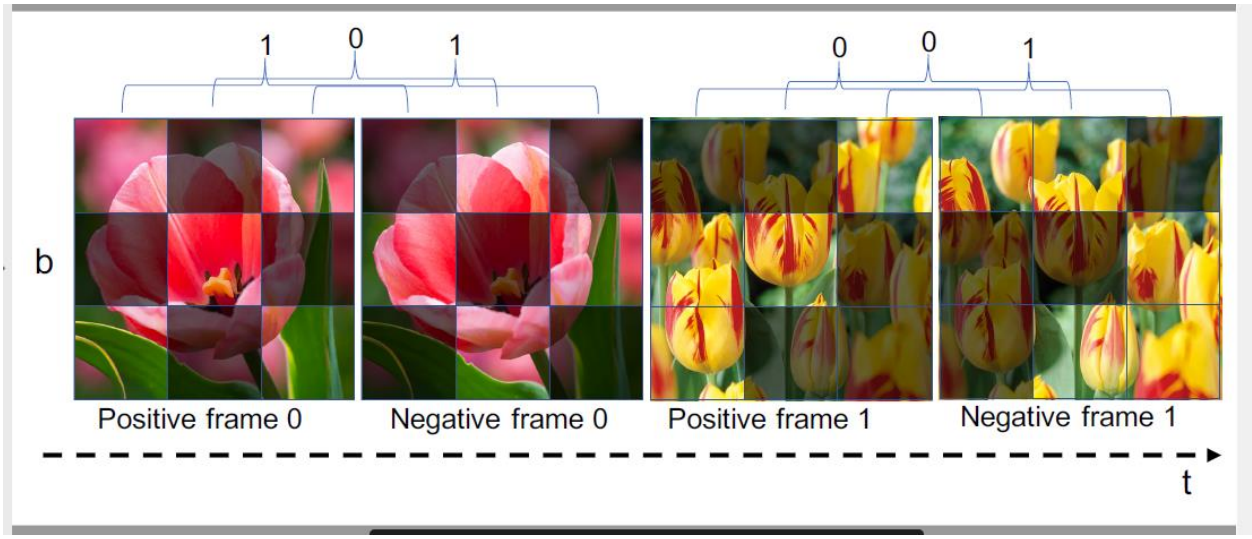

Figure 2.2 Illustrating Manchester coding for flicker suppression

2.5. RELATED WORK

Several novel techniques for improved SCC have been proposed in recent years. Prior work has focused primarily on designing reliable and efficient

(a) encoding mechanisms for improving throughput, and/or

(b) embedding mechanisms to reduce flicker perception. Broadly speaking, the encoder schemes follow two approaches:

- **Manchester Coding:** In this approach, introduced by VRCod, a single bit is encoded over two consecutive frames, with the aim of suppressing flicker effects.
  Figure 2.2 illustrates an example of Manchester encoded embedding in SCC, where bit "1" is encoded as a brightness decrease (first frame +Δ, second frame -Δ), and a bit "0" is encoded as a brightness increase (first frame -Δ, second frame +Δ). This approach leverages the fact that human eyes essentially average out content changes at rates beyond approximately 50Hz; accordingly, if the screen refresh rate R_S is greater than 100 Hz, the +Δ & -Δ effects are perceptually cancelled out. Therefore, Manchester encoded frames are visually imperceptible only at high display rates (unless of course Δ is very small) and thus are ineffective for commonly-used displays with R_s=50/60Hz. Another key challenge when using Manchester coding is the requirement of subtracting two consecutive encoded frames to detect the high-to-low or low-to-high bit transition. VRCode circumvents this challenge by allowing for only time-invariant encoding, where the bits transmitted do not change over time (e.g. embedding a single

QR code-like pattern in every display image frame pair.) While Inframe and Chromacode improve over VRCode by embedding time-varying content, the techniques work well for for only carefully-controlled scenarios, where the camera receiver is static (fixed on a tripod) and the background portions captured the camera remain relatively static and are known in advance.

- **Frequency Encoding:** In this approach, each bit's information is encoded as a temporal pattern over a long set of consecutive frames. The HiLight approach adjusts the screen's brightness (commonly referred to as alpha channel in graphics), instead of the pixel intensity of each image, over a longer run of 16 consecutive frames, using these slower time-scale variations to suppress the flickering perception. More specifically, HiLight encodes a bit ``1'' through brightness variation at 30Hz, and bit ``0'' as variations at 20Hz, considering a screen refresh rate of 60Hz. At the decoder, HiLight uses an FFT to detect a peak at one of the two frequencies (30Hz and 20Hz) to identify the corresponding bit value. The use of a large number of consecutive frames for encoding a single bit, however, results in very low throughput.

As we shall show shortly, while these approaches help establish the foundational techniques for SCC, they make several implicit assumptions that present real-world drawbacks.

Smart without Barcodes/Computer Vision works

Barcode/QRcode (hereafter referred to as QRcode) has been adopted and widely used for years. While a majority of the QRcode are laser readers, recently smartphones with high quality cameras have been used as QRcode readers. Using camera to decode QRcode has been studied intensively and currently becomes a function in almost all smartphones.

Chapter 3

DRAWBACKS OF PREVIOUS WORK AND DESIGN GOALS

3.1. INTRODUCTION

This chapter studies the drawbacks of the previous work by going through the literature, implementing their approaches and running technical as well as user tests. Then we go on to set up design goals for the full project.

3.2. DRAWBACKS OF PREVIOUS WORK

We note that existing SCC designs suffer from three key limitations, detailed in each of the 3 sub-sections below. We discuss and illustrate the limitations via two state-of-the-art methods in SCC: HiLight and Chromacode. We pick these two for comparison as HiLight is a low-rate SCC system with high flicker reduction, while Chromacode, while having lower flicker reduction, has achieved the highest reported throughput among existing SCC prototypes. Thanks to HiLight's source code availability, we re-use the code for conducting feasibility experiments to demonstrate the underlying limitations. As Chromacode's source code is not publicly available, we rely on the information from the published paper to infer various parameter choices and the resulting system performance. Table 3.1 summarizes the specific strategies for HiLight & Chromacode that we shall investigate.

Table 3.1. Summary of the previous work

	Hilight	Chromacode
Screen Detection	Canny & Hough edge detector	Pre-calibrated marker
Flicker Elimination	Embed in brightness (alpha) channel	Embed using manchester coding \& operate at high frame rates
Transition frames	preamble bit pattern for synchronization	specific marker patterns on border pixels

Inaccurate Screen Detection: Need for Explicit Screen Markers

Prior works assume that the receiver camera is capable of *perfectly* extracting the screen's border. Such extraction is key to eventual bit decoding: as the pixels corresponding to each grid (bit) of the display image are obtained by effectively dividing the camera-captured image into N x M grids, any error in extraction effectively results in the decoder using an incorrect set of captured pixels. Prior works have tackled this problem by either using a pre-calibrated screen border (i.e., assuming that the border is known via an external mechanism) or by employing a screen border extraction mechanism. As an example of the former, Chromacode assumes that the screen border is demarcated with a special marker or pattern that is used during an explicit pre-calibration step--this is clearly impractical in a real-world scenario. As an example of the latter, HiLight employs computer vision techniques such as line detection (using the Canny and Hough algorithm) to detect screen borders. However, these detection methods are erroneous in scenarios where

(a) the camera may be moving (e.g., a smartglass-mounted camera), or

(b) the background is not smooth (has significant scene variation) and non-homogeneous.

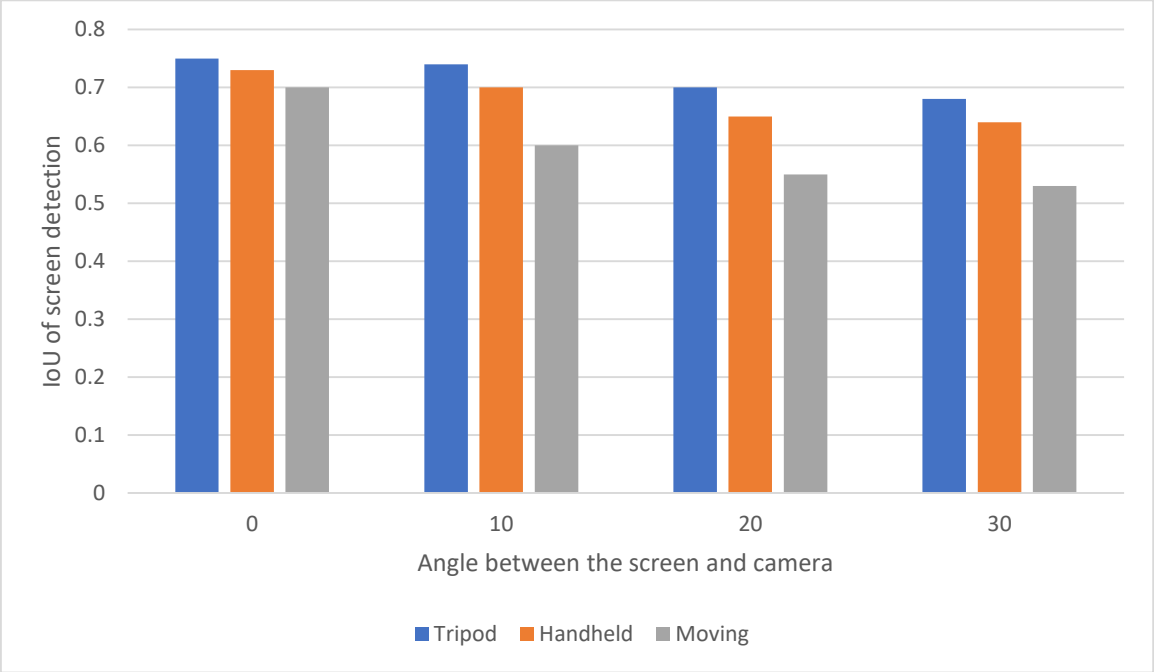

Figure 3.1 Screen detection performance of HiLight using computer vision feature methods.

Figure 3.1 denotes the results from Canny and Hough edge detection methods. This experiment was conducted in a research lab environment with conventional office ceiling white lighting.

To establish the limitation of these techniques, we evaluated the accuracy of the Canny & Hough detector (of HiLight) in detecting a screen under a research lab environment, with different angles of observation. Figure 3.1 plots the Intersection-over-Union (IoU) metric, capturing the accuracy between the detected screen boundaries and the ground truth, for different observation angles. To provide an understanding of how such inaccurate screen estimation impacts the final SCC decoding performance, Figure 3.2 then plots the resulting bit error rate (BER) as a function of the *screen offset*: --i.e.,. We clearly see the brittleness of current approaches that rely on explicit extraction of individual cells/grids: even a modest 25\% offset in grid alignment, pushes the BER to <15%.

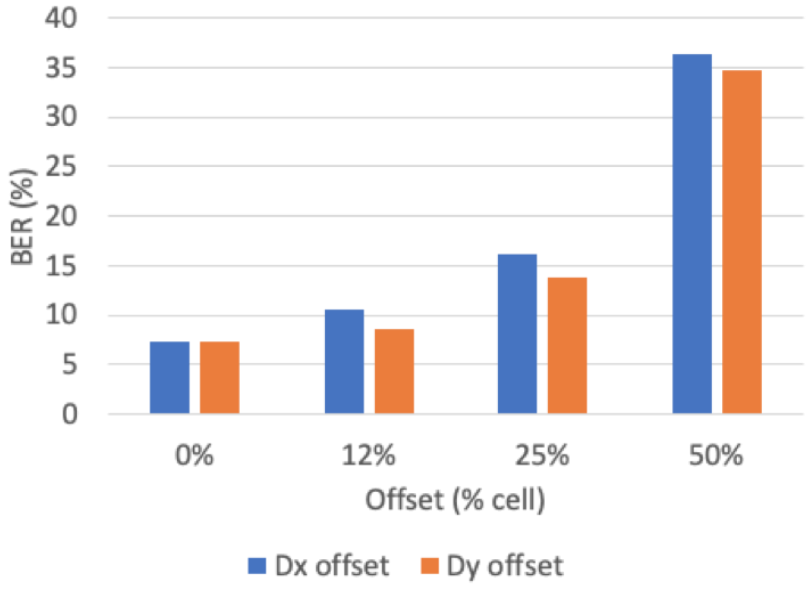

Figure 3.2. Hilight: BER vs. Screen offset

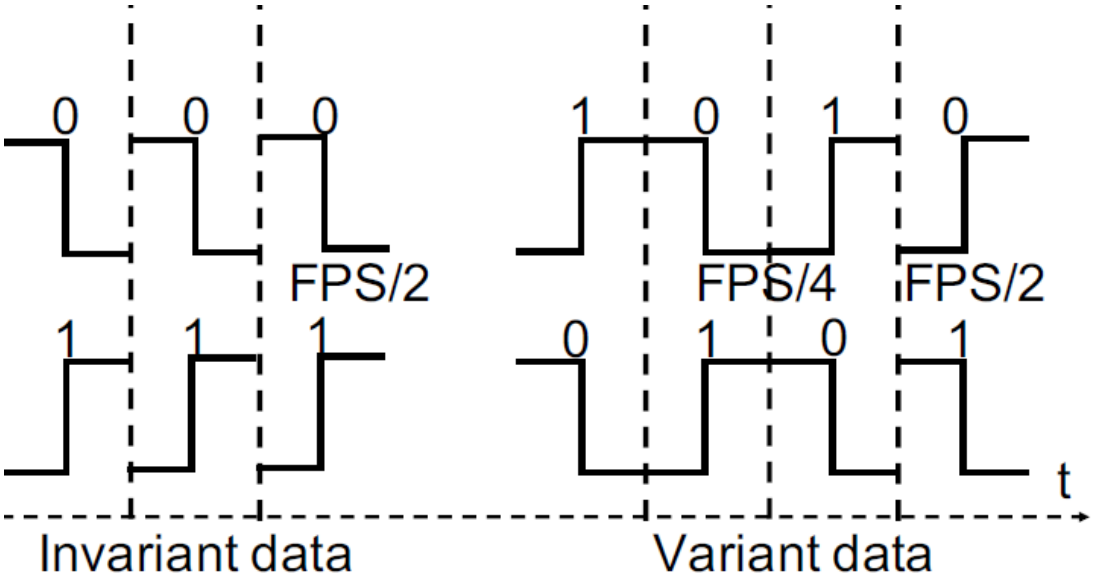

Figure 3.3 Manchester coding with invariant (static) and variant (dynamic) data

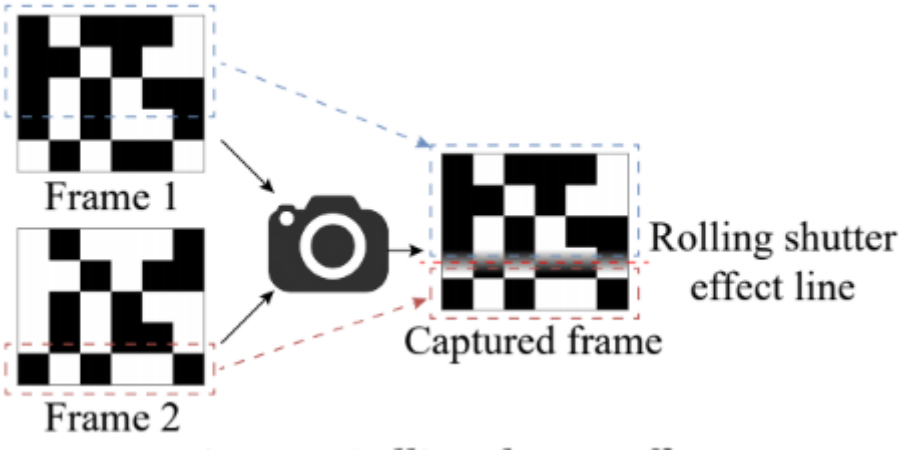

Figure 3.4 Illustrating the Problem of Transition Frames

High flicker perception at low frame rates

Prior work (e.g., Chromacode) assumes that flicker can be eliminated by using very high screen frame rates, in tandem with Manchester encoding. However, we explain, and empirically validate, that flicker can remain a problem even at high frame rates, *when the communication content (i.e., the encoded bits) vary with time.* Accordingly, Manchester coding with moderately high Δ values is imperceptible only if the content is time-invariant (e.g., if the bits are communicating a static URL) and can present problems for more dynamic transmissions (e.g., the audio feed transmission example described earlier). Figure 3.3 illustrates the underlying issue. If the communicated bit (in a specific grid) is unchanged across frames, the grid effectively transmits a regular square wave at *half* the screen frame rate. However, for dynamic data, the square wave frequency can drop to (in the worst case) 1/4th of the screen frame rate (e.g., when a `0' is followed by a `1'). Moreover, for dynamic content, the time period of the square wave can vary intermittently (based on the underlying transmission) between $\frac{1}{2}$ and $\frac{1}{4}$ of R_S; such sharp changes in the frequency of coded signals is known to give rise to stronger flicker effects. Based on the fact that human eyes are unusually unable to perceive changes above 50Hz, Manchester-coded content can be imperceptible only if the display refresh rate $R_S > 200$ Hz. *This is clearly a problem, given that the vast majority of commodity displays in public venues have refresh rates of 50 or 60 Hz.*

To address the flicker issue without Manchester coding, HiLight instead encodes the bits in the longer time-scale variations in the screen brightness, with Δ set to a very low 0.8%. However, even in this case, our empirical studies show the occurrence of perceptible flicker on *larger screens*. (Consistent with the results reported in Hilight, flicker was imperceptible on a smaller 9" tablet device.) Figure 3.4 shows

the flicker perception score using different existing methods. We conclude that the flicker may be imperceptible only when individual cells (grids) are small, as the human eye will then be unable to spatially resolve individual grids. However, such small cells (i.e., high values of N and M) are also impractical, as they become prone to even small offsets/errors in the screen extraction technique.

Need for Explicit In-Screen Markers

The use of mobile cameras to capture the screen content gives rises to another problem: the *rolling shutter effect*. As has been well-documented, to reduce hardware costs, such cameras effectively generate images whose rows/columns are not captured instantaneously, but in sequential fashion. Note that, to avoid *aliasing* problems, the camera sampling frequency (or frame rate), R_C has to be at least twice that of the screen display rate R_S. Even with this higher value of R_C, Figure 3.5 illustrates the problem of spurious or *transition* camera images, which effectively capture composite signals from multiple consecutive display frames. As illustrated in Figure 3.5, the second camera frame $F_{C(2)}$ includes some pixels from the prior display frame $F_{D(1)}$ and some pixels from the current display frame $F_{D(2)}$. Unless such transition frames are recognized and filtered out, the decoding process would generate spurious values, as it is possible that a particular `grid'(bit) would have a mix of pixel values from two consecutive communication bits.

In current work, this problem is tackled via the addition of explicit and well-known ``markers", embedded in each display frame. In particular, Chromacode embeds black and white border lines in each frame, and the decoding software then filters out all captured images which do not have corresponding unbroken black/white border lines. On the other hand, HiLight addresses this problem from a synchronization perspective and uses specific preamble bit patterns that are arranged spatially along the border pixels.

In essence, both, the border lines and specific bit patterns can be considered in-frame marker approaches. This assumption of explicit, in-screen markers is problematic in the real world for two reasons. First, such markers do not just consume screen real estate, they are obviously observable by humans, thereby negating the goal of subliminal communication. Second, different venues/operators may utilize different markers, and it is impractical to assume that the decoding system is implicitly aware of the specific marker pattern in each venue.

3.3. DESIGN GOALS

The practical limitations of SCC technologies discussed above motivate us to develop DeepLight, a novel ML-based system with the following goals:

- **Robust Screen Extraction:** We require a technology that will be able to robustly extract the display screen from the camera-captured image, even in the face of real-world impairments and challenges such as non-static & dynamic backgrounds, different & dynamically changing viewing angles, variable lighting conditions and absence of explicit ``boundary markers'' on the screen.
- **Operate with Very LowΔ:** Because it appears to be difficult to eliminate flicker at moderateΔ, at commonly used display rates, without resulting in dramatic reduction in throughput, DeepLight should be able to achieve reliable decoding while using only low values of Δ.
- **Not Rely on In-Display Markers:** Given the reality of rolling shutter-based camera sensors, DeepLight should be able to identify and eliminate *transition* frames, across a range of display rates and without assuming the existence of any screen-embedded explicit markers or patterns.
- **Reduced Complexity ML:** This is not an SCC-specific goal, but rather based on our strategy of tackling these challenges using ML-based techniques. Given that state-of-the-art DNN techniques are heavyweight, we should try to develop techniques that can provide satisfactory performance with significantly lower model complexity.

Chapter 04

DESIGN AND IMPLEMENTATION

4.1. INTRODUCTION

In this section, we describe the design of the 3 key and novel components that underpin the DeepLight approach to robust, unobtrusive and high bit-rate SCC communication. Our main novelty is in re-imagining the decoding pipeline by replacing the previous mechanisms (based on statistical signal processing) with multiple DNN-based steps. This re-imagining is based on the observation that the decoding process involves several *vision-driven* steps, and is motivated by the unprecedented success of DNN pipelines in tackling various vision-related problems.

We use Figure 4.1 to provide an overview of the DeepLight decoding pipeline. The decoder works by first identifying the screen/display boundaries in captured images, then performing appropriate cropping/resizing (to extract the screen content at highest possible fidelity), followed by identification of the correct sequence of Manchester-encoded frames, with the final step involving the deciphering of the Manchester-coded bits. We next describe how distinct DNN pipelines are used to perform various decoding tasks on the camera-captured sequence of images, such as

(i) robust, accurate screen extraction,

(ii) elimination of spurious, transient frames and selection of appropriate coded frames, and

(iii) decoding of communicated bits, even with inaccurate screen extraction.

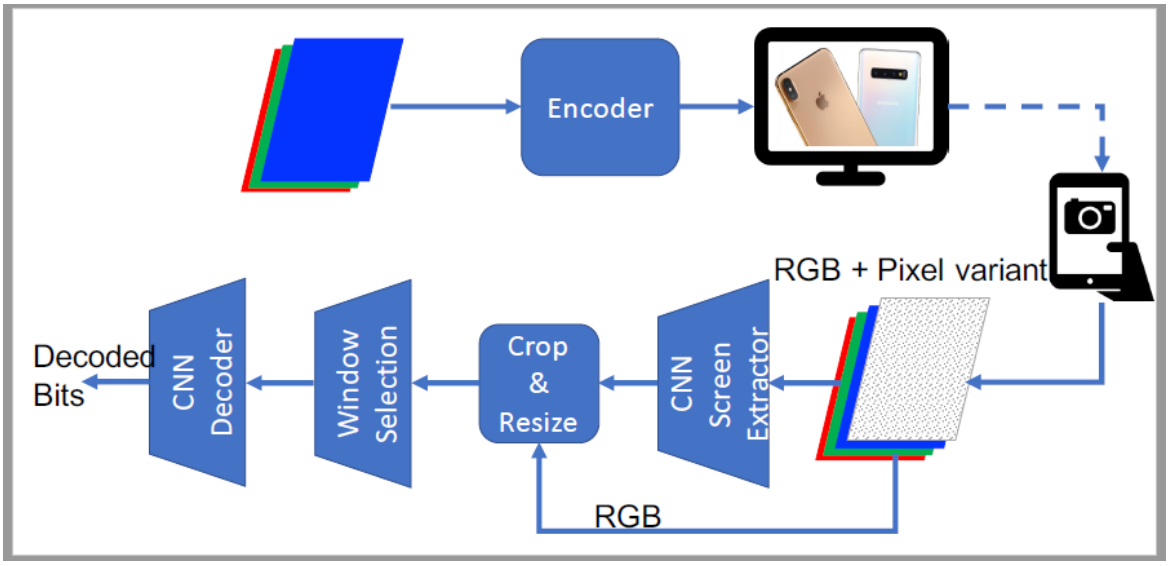

Figure 4.1 Overview of DeepLight decoding pipeline

4.2. SYSTEM DEISGN

Screen detection: ScreenNNet

As previously demonstrated, simple edge-detection based algorithms fail to identify the boundaries of the display screen (within individual camera images) under real-world artefacts such as non-frontal viewing angles and changes in background scenery. Inspired by DNN-based object detection pipelines, we develop and implement our own Convolutional Neural Network (CNN) model to estimate the 4 corners (coordinates) of the screen. Figure 4.2 shows the model of *ScreeNNet*, our DNN-based screen extractor.

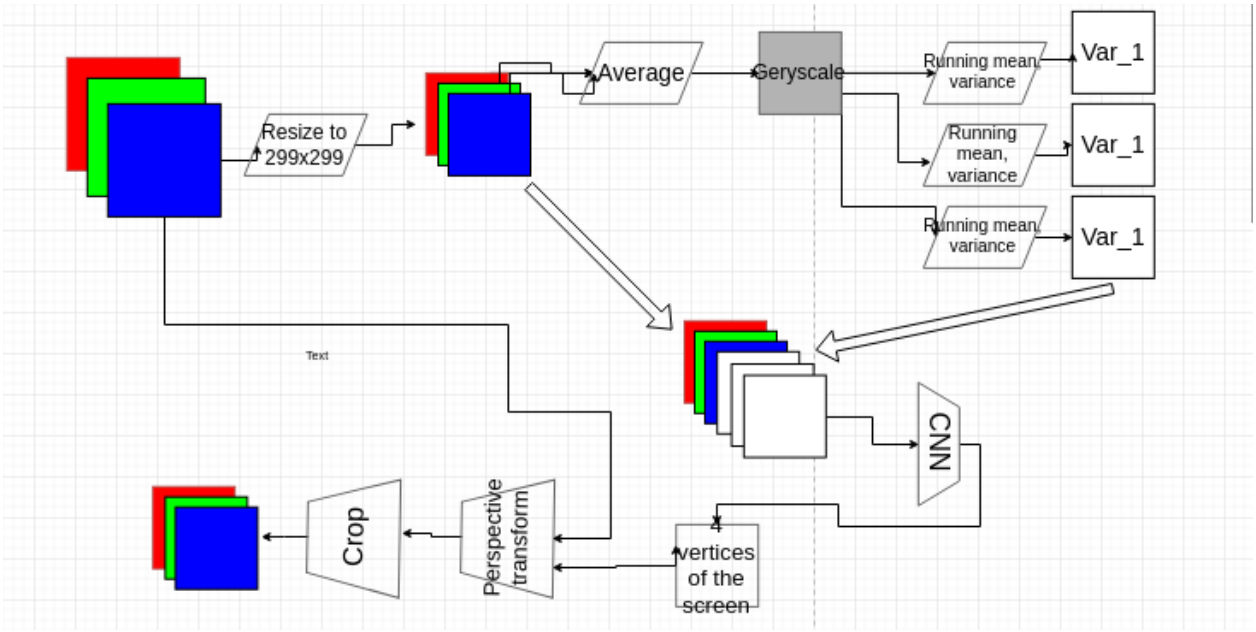

Figure 4.2 ScreenNNet Pipeline

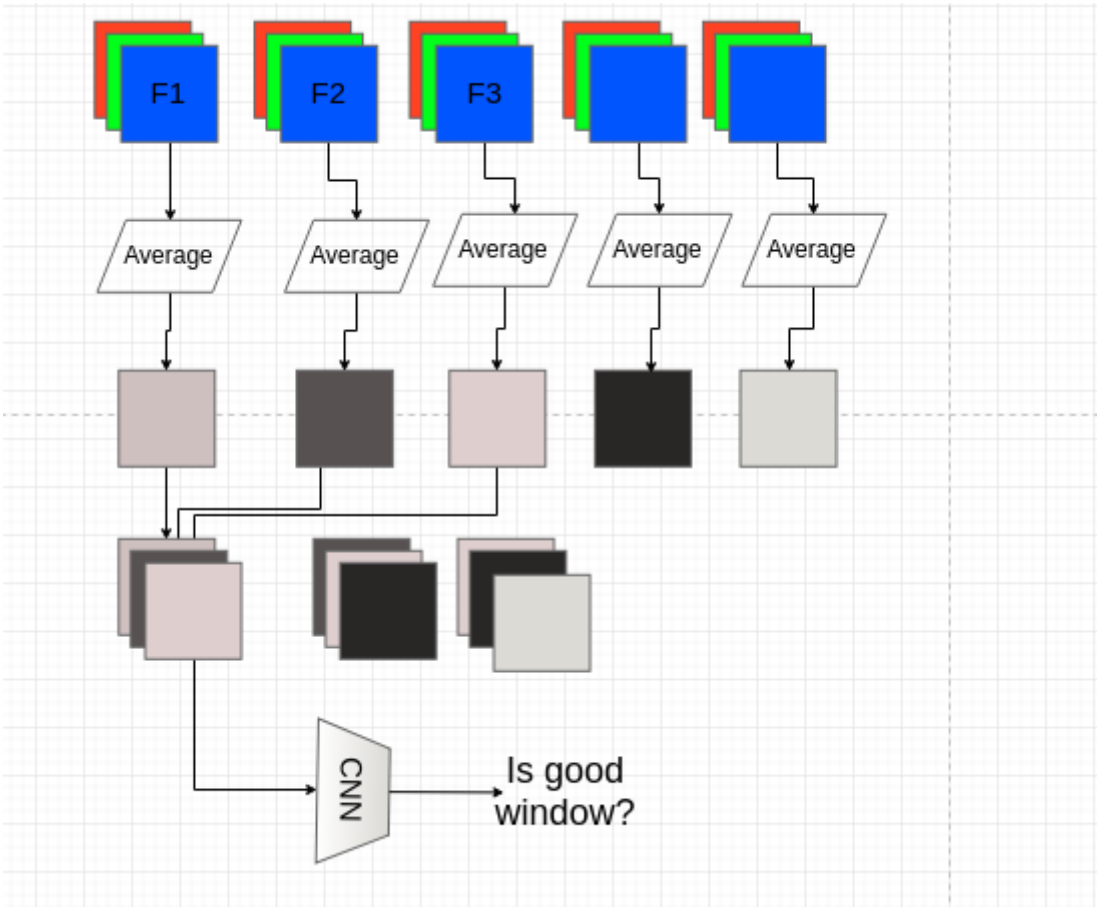

Figure 4.3 TransitioNNet Pipeline

The model is based on the hypothesis that, because of the underlying coding (intensity modulation) of the screen content, *the pixels within the screen are likely to exhibit a characteristic variation (with a frequency equal to that of the screen display rate)*, even if the screen is displaying purely static content--i.e., a still image. In contrast, changes in the non-screen background should have different temporal dynamics--most human movement has lower frequency, while any intensity changes are also likely to have a different variation. Accordingly, for ScreeNNet, we compute the variance in pixel intensity over multiple consecutive frames. The variance σ_t is calculated by equations below for different values of α = {0.01,0.02,0.05,0.1,0.2,0.5}.

$$\sigma_t^2 = \alpha\,(x_t - \mu_{t-1})^2 + (1-\alpha)\,\sigma_{t-1}^2$$

$$\mu_t = \alpha\, x_t + (1-\alpha)\,\mu_{t-1}$$

where μ_t denotes the mean and σ_t denotes the weighted average of the pixel variance.

Accordingly, the input to ScreeNNet includes the current RGB image (3 channels) and 6 channels corresponding to 6 different pixel variances. The ScreeNNet DNN outputs 4 different (x,y) coordinates, corresponding to each of the corners. The system then performs perspective transform, followed by cropping and resizing of the extracted image to a (299x299) pixel image. Note that ScreeNNet does not

attempt to use standard object detection DNNs (such as SSD or MobileNet), as those detectors are based purely on an object's shape--such detectors can potentially pick up other screens (e.g., in the background) even if those screens are not engaged in SCC. In contrast, while ScreeNNet does borrow the basic CNN pipeline, it modifies its input to explicit use temporal variations in pixel intensity that can be expected from code modulation.

Micro-Benchmark Results

We have conducted a fairly extensive set of experiments to quantify the performance of ScreenNNet. The model itself was trained with over 10,000 annotated images of 4 distinct screens (with screen refresh rates varying between 15-120 fps), captured from 2 different public venues, using a iPhone X camera. We evaluate the fidelity of screen extraction using two distinct measures:

(a) the IoU measure defined previously, and separately,

(b) SPAGE, the percentage of the true screen captured within the inferred coordinates.

The former measure is useful in capturing the amount of additional/spurious background captured, while SPAGE helps understand the potential loss in coding throughput (as parts of the screen outside the inferred coordinates are effectively excluded from subsequent decoding operations). The results are given in Figure 4.4. It is evident that screenNNet performance are independent of the screen framerate. But the performance cam drop by a small margin in different background conditions. Table 4.1 has the SPAGE values.

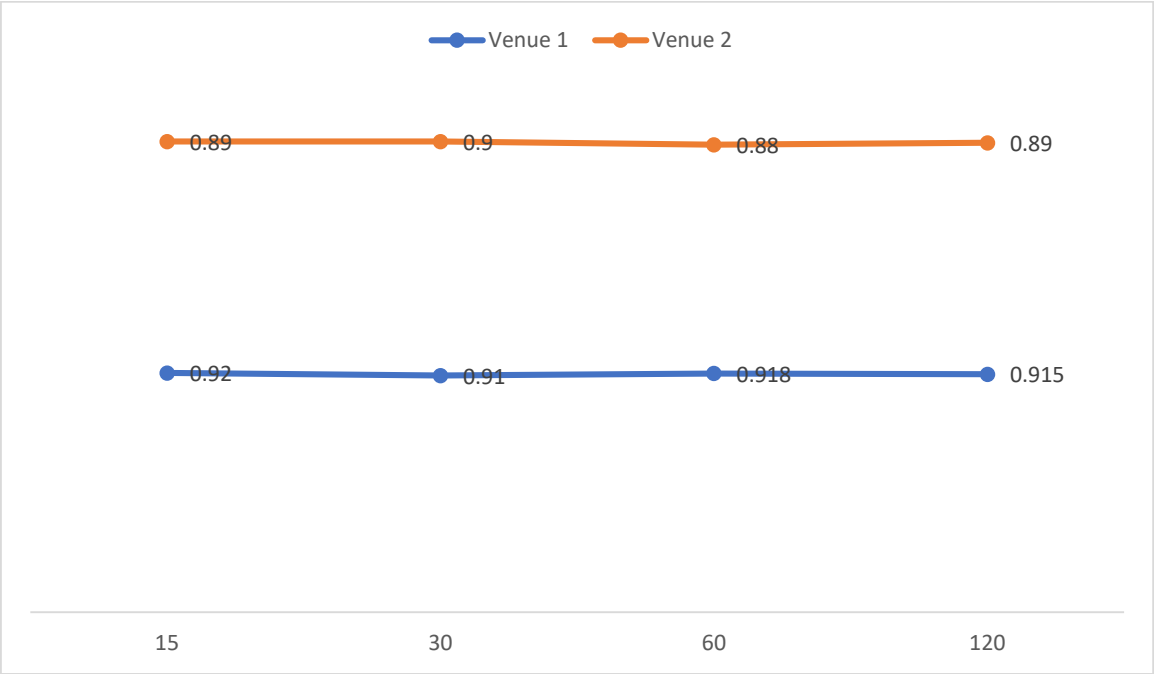

Figure 4.4. IoU performance of ScreenNNet on different venues against different screen frame rates.

Table 4.1. SPAGE values for ScreenNNet on different venues

	Venue 1	Venue 2
SPAGE value	0.98	0.95

Varying Illumination/Background:

Figure 4.4 and Table 4.1 shows the boxplots of both the IoU and SPAGE values, for test data involving 4 screens captured (frontally, with the camera directly facing the screen) at 2 venues, under 2 different lighting conditions (daytime and nighttime).

Varying Observation Angle:

Figure 4.5 provides boxplots of the IoU values for different *viewing angles* θ: θ=0 corresponds to the case where the camera is directly facing the screen. We see that ScreeNNet is fairly robust across variations in both viewing angles and background illumination. In contrast, Hilight and Chromacode are not that robust.

Varying Δ: We also investigate the performance of ScreeNNet with different values of Δ. Intuitively, as Δ diminishes, the coding-driven modulation effect becomes fainter, potentially degrading the performance of ScreenNNet.

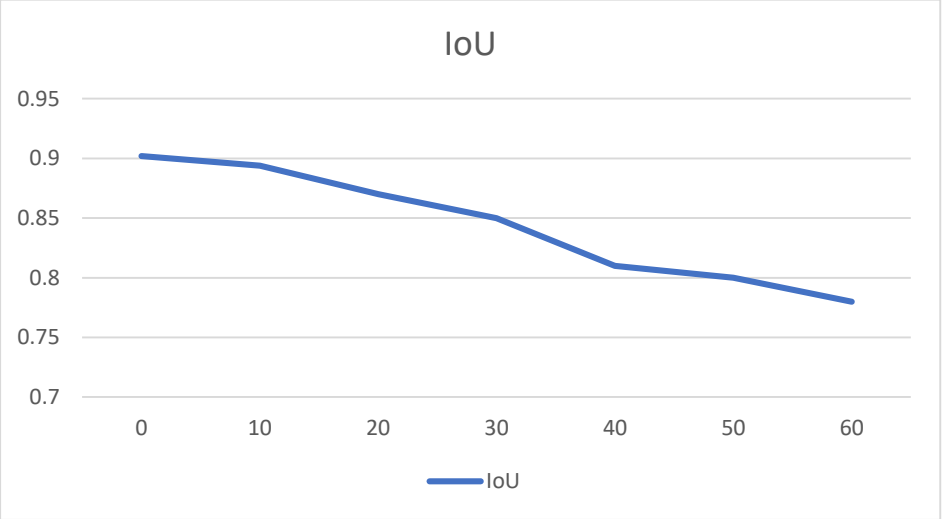

Figure 4.5 Screen detector accuracy when users hold the camera by hand.

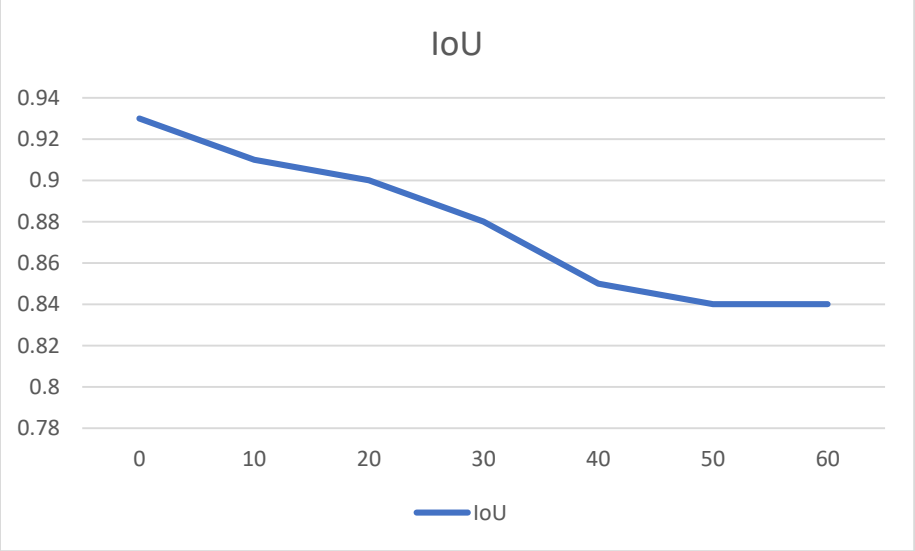

Figure 4.6 Screen detection performance at different viewing angles.

Filtering Transition Frames: TransitionNNet

As explained previously, the rolling shutter effect implies that the camera-captured images have a mix of properly coded *data* frames, interleaved with incorrect *transition* frames. We propose a modified CNN architecture, *TransitionNNet with temporal memory*, to tackle this problem. Figure 4.3 provides an overview of this CNN model.

For ease of explanation (this can be easily generalized), consider a sequence of frames F_1,F_2,.. captured by the camera. Moreover, assume that we employ a Manchester coding mechanism, whereby the transmitter encodes each bit into a (M+,M-) or (M-,M+) pair of successive display frames. A sliding-window protocol then takes K (for illustrative purposes, assume that K=3 corresponding to the case where the camera frame rate is twice that of the screen display rate, i.e., R_C=2 R_S) frames and applies TransitionNNet to determine if a given frame triple \{F_i,F_{i+1},F_{i+2} is *valid*. If so, DeepLight can assume that F_i and F_{i+2} are the encoded pair of frames and pass it along for decoding; if not, this frame triple is discarded and TransitionNNet is applied on the next triple F_{i+1}, F_{i+2}, F_{i+3}.

To make this determination, TransitionNNet first computes the average of the RGB values of each individual camera frame, creating a single gray-scale matrix for each frame. These K frames are then stacked together (creating a K-channel `virtual image') which is then fed into a binary CNN classifier (valid vs. invalid?). Note that TransitionNNet makes no assumptions whatsoever on the presence of any on-screen markers, and can be trained to support multiple display rates (which correspond to multiple possible transition frames between a valid Manchester-coded pair).

Robust Bit Decoding: DLNet and LightNet

As previously shown in Figure 4.5, the screen detection using our ML based models can localize the screen with a IOU value of more than 80%. However, there is still a significant amount of background scene in the bounding box. Existing methods, which explicitly partition the extracted `screen' into spatial grids, then exhibit poor perform, as they effectively attempt to decode bits from *misaligned* grids. To overcome this limitation, we instead propose a CNN-based approach, where the pipeline uses observed *intensity variations among neighboring regions*, at multiple spatial scales, to implicitly determine the grid boundaries.

Single-shot decoder

We first explore if a convolutional neural network can take advantage of its multi-scale spatial features to extract the grid boundaries from a single captured image. Intuitively, during the supervised training phase, the CNN will learn that:

(a) in any screen image, the grids/cells will be uniformly and regularly spaced, and

(b) that under `random' bit patterns, the pixel intensities should differ across, approximately, half of the neighboring cells. The decoding pipeline then effectively tries to compute the best offset values (i.e., the grid partitions) that best matched these learned properties.

More specifically, we apply the InceptionNet model (2 blocks only. This is shallow than the original paper), that represents the state-of-the-art for many vision applications. Figure 4.7 shows the block diagram of the decoder.

Experimental results show that this single-shot decoder is able to decode the embedded data accurately, when Δ >= 4%. However, the BER increases significantly when Δ=2%. Because prior results show that flicker is perceptible even at Δ=2%, it appears that the single-shot decoder is not very effective while simultaneously preserving the imperceptibility of encoding.

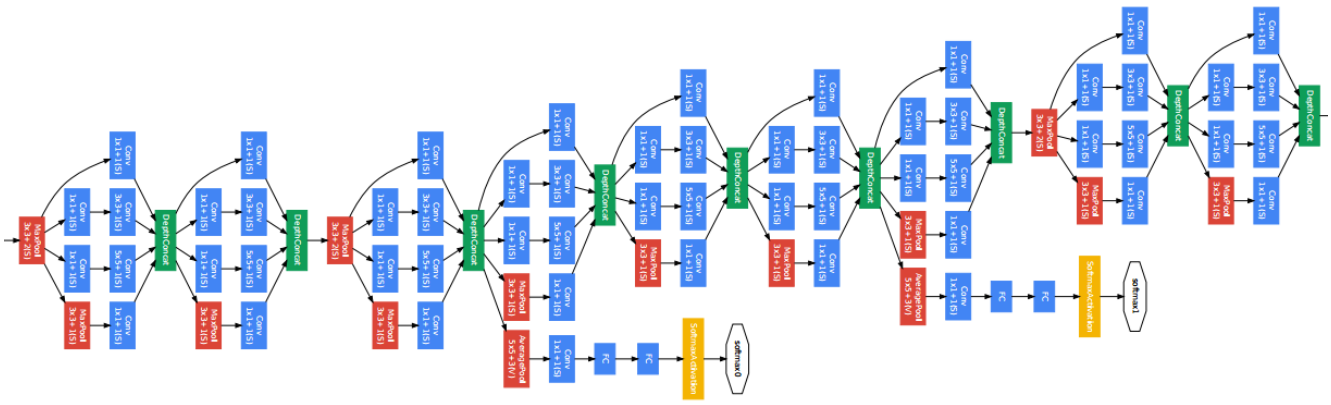

Figure 4.7. Inception network for single shot detection

Temporal decoder

To achieve robust decoding with even lower Δ, we need to develop a DNN model that is less dependent on the visual content within a frame, but instead uses the temporal variation of pixels *across frames*. To provide the decoder a clue of temporal variation, we need to feed it with consecutive frames, and we encode each "bit" as an amount of pixel variation over time using Manchester coding scheme.

Reduced Complexity DNN: LightNet

Though the Inception modules achieve fairly high bit accuracy, it is prohibitively complex: the total number of parameters for 299x299 pixel input image and a 10 x 10 grid is 8.76 million! And thus difficult to be scaled up to support higher resolution input image (e.g. full HD image for larger grid size ~ 80x60). Taking a closer look at a neuron, it can actually take a weighted sum (or subtraction) of the input frames

and thus eliminate the dependence of the original visual content. The elimination of visual content dependency leads to less complicated NN architecture. For example, it is more difficult to extract a grid of dimmer and brighter cells from an image with many vertical and horizontal line texture, and we need a complicated model to differentiate the features (horizontal/vertical edges) of data grid and ones of the original image. The situation is even more difficult if we have to reduce the brightness adjustment to eliminate the flickering effect.

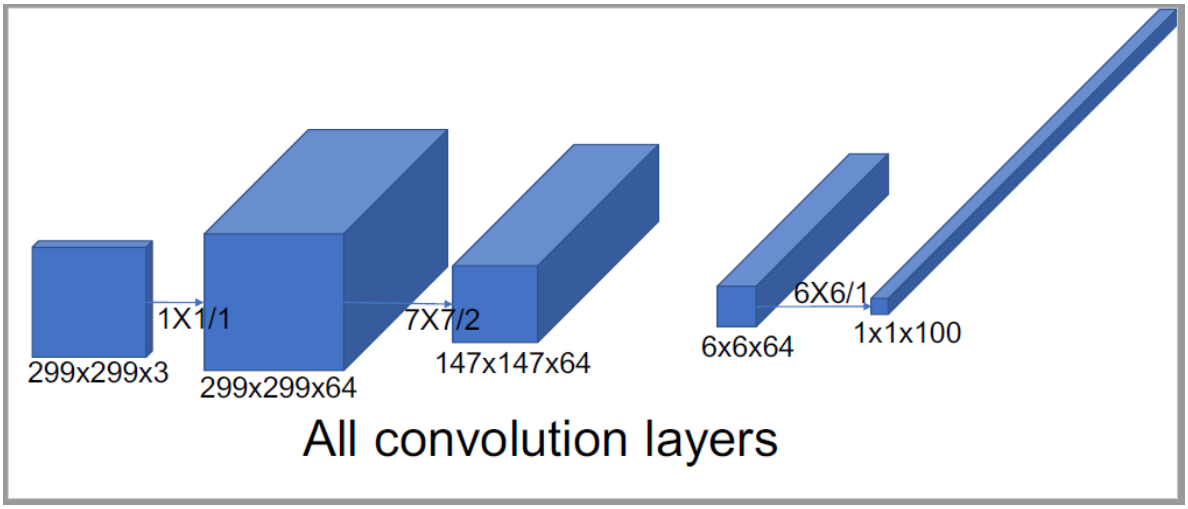

Figure 4.8 :LightNet

Another aspect that makes traditional image classification models (e.g. Inception) too big when applied to this type of image processing is the fully connected layer. To support high throughput, many bits are embedded into an image. The fully connected (FC) layer will link all extracted features to an output bit. Assumed that the extracted features have a dimension of WxHxD, and the total number of output bits is B, the total number of parameters of the FC layer is WxHxDxB. For example, the extracted feature tensor has dimension of 17x17x288 and the output size is 100, the total number parameters of FC layer is 17x17x288x100 = 8323200 parameters. However, for SCC application, each bit output is dependent of the others. We then can align the output bits in the depth channel of a convolutional layer. We will design the neural network so that the last convolutional layer has a dimension of 1x1xB.

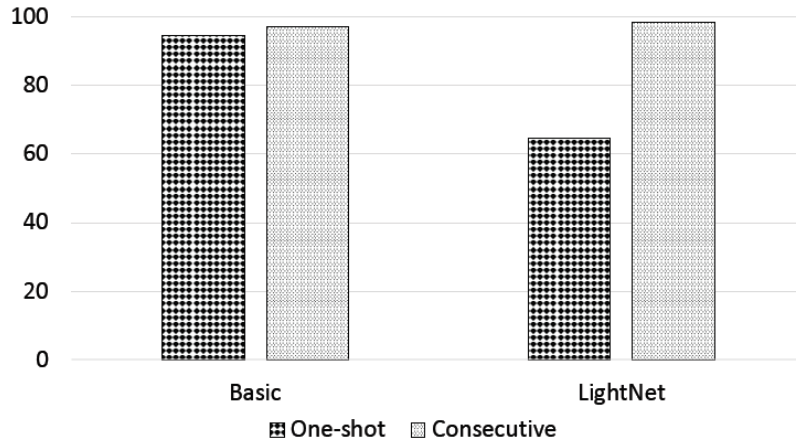

Figure 4.9. Accuracy comparison between Basic and LightNet DNN pipeline.

We also found that having the first convolutional layer size of 1x1 improve the output bit accuracy. The possible reason is that the 1x1 convolutional layer extract only temporal information, and thus prevent the visual content to flow to the next layers which might create visual content dependent. Figure 4.9 shows that the accuracy of LightNet is even slightly higher than the neural network using Inception modules. This holds with temporal encoding method only as we need a more complex model to extract data from a usual image.

4. 4. EXPERIMENTAL SETUP

The whole design was experimented on the following hardware

(1) Intel workstation
(2) iPhone X
(3) NVIDIA DGX server

The intel workstation consisted of a 8 core 16 thread CPU, 32 GBs of RAM and a 512 GB SSD drive. The display was running on a NVIDIA 1080X graphic card and the monitor allowed NVIDIA GSYNC for accurate frame display. The operating system used was Ubuntu 16.04.
The videos were recorded on a iPhone X with a custom written camera app (for constant exposure and focus). The setup is shown in Figure 4.10.

The training process of the neural network pipeline was run on a NVIDIA DGX server with 8 16GB V100 GPUs, 512GBs of RAM and 4 1.92 TB storage SSDs. This server is one of the most powerful servers used in academic research. A photograph is shown in Figure 4.11.

Figure 4.10. Screen-Camera setup

Figure 4.11. Nvidia DGX server (obtained from web)

Chapter 5

EVALUATION AND DISCUSSION

5.1. INTRODUCTION

This chapter elaborates the evaluation of the proposed system. It is followed by a discussion about the possible future directions of research.

5.2. EVALUATION

The system was evaluated for user convenience by holding a user study with 40 participants. The sample consisted of

- 20 students
- 10 faculty
- 10 non-technical personals

The participants were requested to evaluate the usability of DeepLight in comparison to Chromacode and Hilight. They had to answer multiple choice questions by assigning *opinion scores* regarding the usability of the systems. (1 being the least usable and 5 being most usable). MOS, mean opinion scores are given in Figure 5.1. It is evident that DeepLight passes the usability scores of Chromacode by a large margin and Hilight by a smaller margin. Nevertheless, DeepLight is the most usable solution.

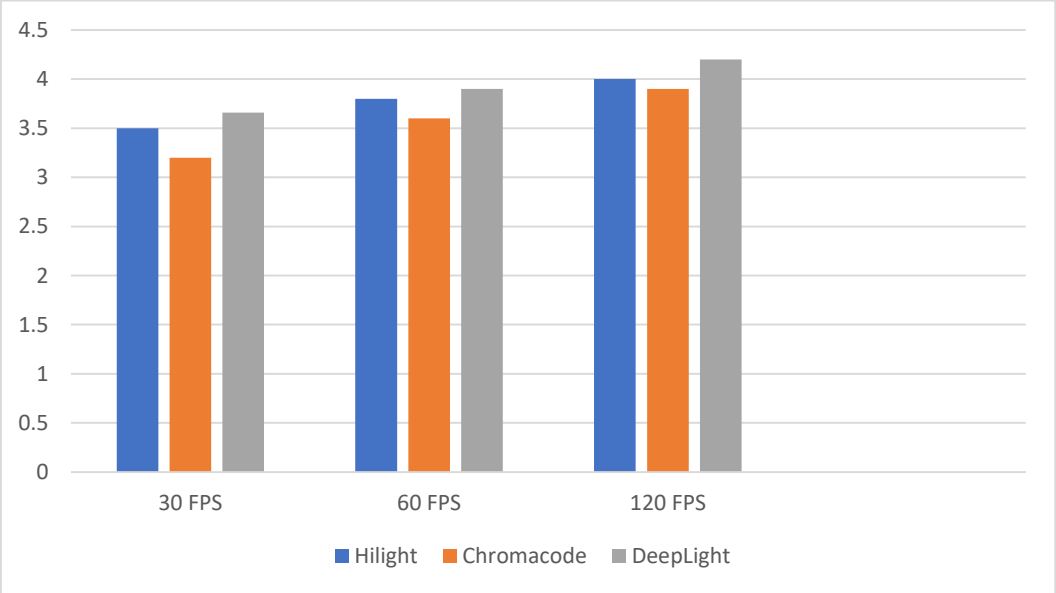

Figure 5.1. MOS of SCC systems against different screen framerates

5.3. DISCUSSION

The following points are worth noting,

Spatial Encoding: The current RS-code based encoder operates in row-based fashion: each symbol is encoded on to a row of the cells/grids. Empirical observations, however, suggest that the spatial distribution of bit error is not uniform or `random'--instead, the chances of incorrect decoding are higher around the edges of the screen. This is arguably not very surprising, given the inevitable errors in screen extraction. However, this phenomenon suggests that symbol decoding can be more robust if the bits of a symbol are *spatially distributed*--e.g., bits of different symbols are interleaved at the screen edges.

Automatic Grid Size Detection: While the current \name system does not assume a-priori knowledge of screen refresh rates or special in-screen markers, it does assume explicit knowledge of the grid size. More specifically, different grid choices lead to different DLNet models (with different MxN output labels), each of which must be separately trained. Such a restriction can possibly be overcome by training an *ensemble* model (with different underlying MxN models), and using well-known techniques (e.g., attentional weights) to automatically select the model that best matches the current screen content. However, ensemble models will further increase the computational overhead as multiple DNNs must now be executed in parallel. The development of such an *auto-tuning* DLNet architecture thus remains an open question.

Shape modulation: All of the conceptual explanation and experimental studies performed here have used square-shaped cells. While such cells permit easy tessellation within the typical rectangular display, the cells give rise to clear horizontal and vertical `edges' (transitions between two cells) and the well-known checkerboard pattern. Arguably, such edges are easier to perceive and potentially give rise to more noticeable flicker. Because DNNs have proven to be capable of recognizing objects with arbitrary shape, it is plausible that SCC functionality could leverage on this DNN property to further improve the imperceptibility vs. bit-rate tradeoff, by using alternative cell shapes, with smoother, non-continuous edges.

CONCLUSION

My internship was at Living Analytics Research Center, School of Information Systems, Singapore Management University where I worked with Vu Tran (a 5th year PhD student), was advised by Prof. Arhcan Misra (Dean of research, SMU) and Asst. Prof. Ashwin Ashok (Georgia Tech University). We conduct research on visible light communication between computer screens and cameras to develop **DeepLight**. DeepLight is the first screen-camera communication technology to use deep learning (a modern artificial intelligence technique). My group was able to surpass the performance in robustness and practicality of the previous screen-camera communication technologies. (put forward by Tsinghua and Maryland universities in 2018, Dartmouth in 2016 etc:).

During my internship, I was able to do original research with a good balance of guidance and independence – which boosted my drive to continue doing research through the remainder of my undergraduate studies as well as my future professional life. Working in a group with a senior professor from Singapore, a relatively new professor from USA and a PhD student aspiring to join the industry allowed me to look at research from a diverse set of angles – which I suppose to come in handy when I am making my career choices.

The education received during the first three years of the undergraduate program was instrumental in successfully conducting research during the internship. First and foremost, the computer science basic knowledge gathered through core courses came in handy throughout the 22 weeks. The freedom to choose technical elective courses was advantageous since I was able to build up the expertise in a subfield and do the internship work on the same subfield. The flexibility offered by the department to engage in research projects from the second year onwards and the willingness of the academic staff to advise the students on research projects came as a strong foundation to adapt to the research environment at SMU very quickly and deliver results.

Working on a solution to a technical problem of a very active field is a healthy challenge. I was able to utilize my time during the internship and the resources (human and other) effectively to work towards the completion of the project. I was able to learn new theory, techniques and skills during the period. In addition, I was able to interact with and make friends with a diverse community of researchers from all around the world in different steps of their careers. Furthermore, I was able to experience living in a new environment, indulge in their culture and travel.