1-1-2017

Analysis Of Cross-Layer Optimization Of Facial Recognition In Automated Video Surveillance

Loren Garavaglia
Wayne State University,

ANALYSIS OF CROSS-LAYER OPTIMIZATION OF FACIAL RECOGNITION IN AUTOMATED VIDEO SURVEILLANCE

by

LOREN GARAVAGLIA

THESIS

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

2017

MAJOR: Computer Engineering

Approved By:

_____

Advisor Date

DEDICATION

To all those that have been supportive of me in achieving my goals, especially my parents, Kimberly and Brian

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Nabil Sarhan. His guidance and expertise helped me greatly throughout my efforts of developing this thesis. I would like extend gratitude to my committee members: Dr. Mohammed Ismail Elnaggar and Dr. Loren Schwiebert. They worked on very short notice to provide valuable feedback. I would like to thank my friends and family for being supportive of me throughout the process of completing this thesis. Finally, I would like to thank Sina Davani and Hayder Hamandi, who also work under the guidance of Dr. Sarhan. Their work provided valuable assistance in the development of this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Introduction

In recent times, interest in video surveillance systems has grown dramatically and with that so too has research on the topic. It is projected that the video surveillance market value will reach USD $75.64 Billion by 2022, up from USD $30.37 Billion in 2016 [1]. In addition, adoption of newer technologies, such as IP cameras and wireless video transmission, have created ground for rapid expansion of the market [2]. Thus far, much of the research that has been performed has focused on providing accurate and precise detection of security threats in a timely manner, but a larger focus is now being placed on advancing these topics to improve the recognition of threats in addition to detection [2]. Advances in technology have allowed the development of efficient approaches to automatically detecting and monitoring the progress of threats in real-time. These systems, referred to as automated video surveillance (AVS) systems, are becoming increasingly prevalent, resulting in increased demand for improvements to threat detection accuracy.

Many studies have been performed that focus on developing better computer vision (CV) algorithms for improved detection, tracking, and classification of objects [3-8]. Other studies have focused

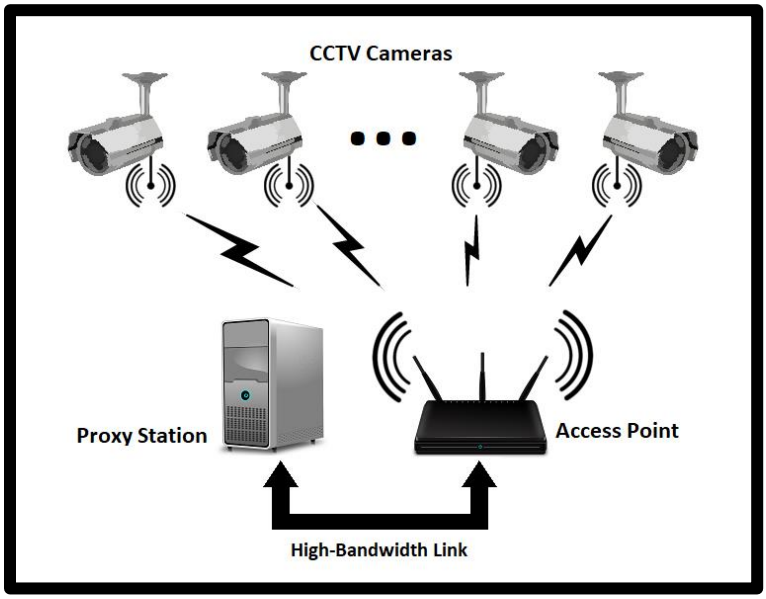

Figure 1.1: An Illustration of an Automated Video Surveillance System

on detecting and classifying unusual events [9-14]. All of these studies have focused on the CV aspect of the AVS systems, without considering the entire architecture. While these types of studies attempt to tackle the problem of object detection at the application level with CV algorithms, these increasingly complex algorithms have done little to address other issues that come with the expansion of surveillance systems. The complexity of these proposed algorithms is acceptable with smaller network sizes, as current processing technology is able to handle the level of computational load required, however, as the scale of AVS networks increases these algorithms will have difficultly running in real-time due to their high computational requirements. Therefore, new methods for increasing AVS efficiency must be explored to address the cost and scalability issue arising in this field.

The scalability-cost problem becomes a concern as additional video sources are added to a system in an effort to increase coverage. In these situations, a larger strain is placed on the network as the total bandwidth required by the video sources increases. Additionally, increasing the number of video sources also requires increased computational capability in order to process all of the transmitted video streams. This computational cost can even become a concern in distributed processing architectures because video encoding, decoding, and CV algorithms are some of the most computationally intensive activities a processor can be tasked to perform. Power consumption is another major concern as surveillance systems begin to evolve, especially as wireless, battery-powered video sources become more commonplace. Even in systems that do not use wireless or battery powered technologies there is financial incentive to reduce energy usage in always-on systems, such as surveillance equipment.

A few studies [15-17] have considered using image distortion as the basis for bandwidth optimizations in general streaming systems. While not all of these studies focused on the scalability-cost issue, their findings have shown promise in addressing the matter. Though optimizations based off of distortion may show some benefit in terms of face detection, network load, and power consumption, it is not the best method for use in AVS systems. The ultimate goal in AVS systems is to have high accuracy in recognition and thus threat detection. While distortion does have some effect on detection accuracy, it makes more sense to characterize an optimization based off the desired output (accuracy). To our

knowledge there has only been one Study, [18], that has addressed accuracy-based cross-layer bandwidth optimization (ACBO). This study used face detection accuracy as the basis for bandwidth adjustments in an AVS network, which directly correlates the desired output to the input values of the model. This study showed promising results in improving face detection accuracy while also reducing network load and power consumption; however, it does not address the topic of facial recognition accuracy. While face detection is an important step in the progress of AVS systems, the goal is for such systems to be able to detect and recognize, from a database, threats in an accurate and precise manner. With recognition, in addition to detection, increased confidence in threat handling is achieved.

The ACBO solution considers a system in which conditions in multiple network layers are monitored. In the ACBO approach, three layers are considered: physical, link, and application. Using data gathered across several network layers provides the ability to perform more accurate and timely calculations of the effective airtime of a medium, thus providing more robust control over bandwidth. By utilizing the data collected from parameters in each of these layers the optimization solution intends to adjust bandwidth, and consequently video quality, in a way that maintains a high level of face recognition accuracy, while also reducing overall network load and power consumption. Logically, face recognition accuracy of a CV algorithm directly correlates with video quality, which we intend to show through our experimentation. However, as video quality increases, so does the bitrate and thus the required bandwidth. Although increasing the video quality does increase accuracy of face recognitions, Study [19] (and references within) demonstrates that the sensitivity of a CV algorithm to video quality is much less than that of a person. Therefore, by characterizing the accuracy of a database at different levels of video quality (i.e. bitrate) the optimization solution should be able to adjust the bitrate to achieve the best possible accuracy.

Previous work done on the topic of accuracy-based cross-layer optimization [18] (and references within) provided a limited scope into the effectiveness of such solutions. In previous work, face detection accuracy was optimized rather than face recognition accuracy. In addition, adjustments to the sending rate were done in an indirect fashion, making calculations and controls cumbersome. The prior work on this topic did not use a complete implementation of a video streaming system, instead choosing to simplify the

system. While face detection is a useful first step in an automated video surveillance system, it ultimately falls short of the ideal functionality of automated video surveillance. With face detection only, the system is not able to narrow down and identify threats. Instead, every detected face is reported and the managers of the AVS system must sift through the data to find the threats and with no identification provided, system managers must manually cross-reference a threat database if an identity is needed. The amount of data will vary depending on the environment in which the system is placed, however, with face recognition the need to examine data is no longer present and threats can be handled in a much quicker manner. With accurate face recognition threat identifications can be handled automatically. Face recognition is a much more complex and computationally intensive task than face detection. With face recognition, in addition to detecting faces, the algorithm must compare the detected face to a database of faces in order to determine an identification. Typically the detected face is processed to reduce data dimensionality before comparing to the database. Depending on the size of the database and the complexity of the dimensionality reduction algorithm, facial recognition can be a very computationally intensive and time consuming task. Despite shortcomings in current face recognition algorithms, optimizations can be made in a system in order to achieve the best possible level of recognition accuracy with current technologies. In addition, the benefits of facial recognition to AVS are so great that it has become a necessary member of such systems.

This thesis analyzes the effectiveness of the ACBO solution to the scalability-cost problem proposed in [18] when applied to face recognition. None of the earlier work investigated the efficacy of accuracy or distortion-based solutions when applied to face recognition. We perform extensive work to integrate face recognition into the ACBO solution. Using rate-accuracy characterization functions, the previously proposed solution intended to optimize face detection accuracy through adjustments to sending rates on a per video source basis. In this thesis, we show that the developed model provides an accurate characterization of the relationship between video source sending rate and face recognition accuracy in addition to face detection accuracy. This thesis intends to address the shortcomings of prior work by characterizing the rate-accuracy model for facial recognition accuracy of our selected video set and developing a full streaming client to send real video data over the network. This allows enhanced control

over sending rate and more closely resembles a real-world AVS implementation than previous work. Furthermore, past work used older, less efficient codecs to compress the data sent over the network. This thesis delves into the effectiveness of the ACBO solution when used in conjunction with H.264. In addition, we analyze the effectiveness of weighting in the ACBO solution, implementing modifications and performing experimentation that examines the effect of weighting on several performance metrics.

This thesis considers a specific AVS system in which multiple video sources, in various locations, capture and send video feeds to a central proxy station over a single-hop IEEE 802.11 wireless local area network (WLAN). Figure 1.1 shows an example of an automated video surveillance network. In the system, the medium can be shared by both battery-powered and non-battery-powered wireless video sources. A high-bandwidth link connects the proxy station to the access point (AP); this connection is assumed to have a high enough bandwidth that it is not a bottleneck. The proxy station runs CV algorithms which generate automated alerts whenever suspicious events/objects are detected in the monitored site. Large systems may contain multiple of such networks in order to distribute some of the video processing load.

In addition to the analysis of the ACBO solution, we propose and test two enhancements to the solution to provide better bandwidth utilization. The first enhancement proposes limits to bandwidth at smaller network sizes to stop the system from choosing an unnecessarily high sending rate. As discussed previously, studies have shown that face detection and recognition algorithms are tolerant to changes in video quality. While the optimization solution attempts to adjust the bitrate to minimize the network load and power consumption, at smaller network sizes where resources are in surplus the solution tends to choose a higher sending rate than is necessary. The second enhancement involves partially distributing the less intensive CV tasks to the video sources in order to reduce the computational load at the proxy station as well as performing manipulations on the sent video frames to reduce the amount of data sent.

The **main unique contributions** of this thesis are as follows: (*i*) analyzing the effectiveness of the ACBO solution from [18] when applied to face recognition, (*ii*) modifying the system to perform training and recognition on real-time video, (*iii*) developing and analyzing a bandwidth capping system to enhance

the effectiveness of the ACBO solution, and (*iv*) developing and analyzing a distributed CV implementation for ACBO to provide further reductions to proxy station load and data sent over the network.

The results are based on extensive simulations using our chosen video dataset. These simulations allow us to assess the effectiveness of the ACBO solution when applied to face recognition. We perform simulations using the OPNET network modeler as it offers robust features for simulating and adjusting parameters for the type of network considered in this study. One major feature of the work done in this thesis is the sending of real video frame data over the simulated network, mirroring real-world implementations exactly, rather than sending abstract bit streams over the network to mimic video streams like previous work has done. To implement this system, we created a full video streaming client using FFmpeg to encode and decode frames as well as developing a full Real-Time Transport protocol (RTP) implementation for transferring video data over the network, effectively creating a streaming server implementation. While this is a much more complex implementation, mimicking real-world systems exactly with the simulated system gives confidence that our results are valid. Our results show that the ACBO solution is effective in improving facial recognition accuracy in the streamed video feeds as well as significantly reducing the power consumed by the video sources through considerable reductions in the sending rate. We show that the effective airtime algorithm provides an accurate estimation allowing convergence to occur quickly even with a high compression video codec. We also show that the proposed enhancements to the ACBO solution offer significant and meaningful benefits.

Chapter 2 Background Information and Related Work

This chapter provides a detailed look at topics necessary for understanding the work done in this thesis as well as providing a better understanding of the progression of work leading up to this point. The topics involved with AVS are vast in their breadth; the sections in this chapter summarize the main topics covered in order to provide the scope necessary to sufficiently comprehend the work done for this thesis. In the following sections, information is provided that details face recognition, including three algorithms that were considered for this work, the benefits of face recognition in AVS systems, and information on cross-layer bandwidth optimization solutions.

2.1 Face Recognition

The task of recognizing and identifying faces is a common and simple routine for humans, however, for machines this is a much more complicated task. Face recognition has become a major area of research in CV algorithms since it was first introduced. The importance of this technology can be seen through its use in many different applications. One area in particular that benefits greatly from automated face recognition is video surveillance. Utilizing CV algorithms, security systems incorporate the ability to automatically recognize faces of people in the video feeds of cameras. This has the benefit of being a passive system, where no direct input required by the target of the face recognition, this is unlike other security system strategies [20]. In addition, with sufficient accuracy of the system, minimal input is required from managers of the surveillance system to provide corrections. In its beginnings, face recognition started as an extension of face detection, attempting to provide a way to characterize facial features in order to determine the identifying characteristics of a person [21]. Studies [21-24] (and references within), show some of the early methods that were used to identify faces reliably and automatically using CV algorithms. These methods have provided a foundation for the field of face recognition and have been expanded upon considerably over time.

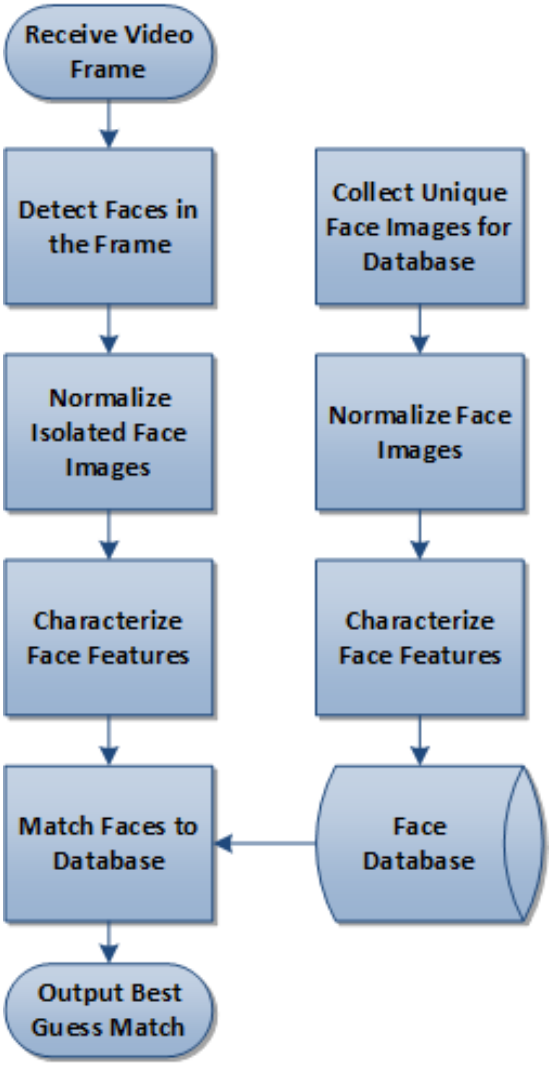

Figure 2.1: Process for Performing Face Recognition

Many methodologies exist for performing face recognition, where most differ is in the specific details of classification and feature extraction of faces. However, all systems that we are aware of follow the same general steps to perform a recognition. Those tasks are face detection, face normalization, face feature extraction, and face matching. Figure 2.1 shows the full process for face recognition. Face detection, as has been covered in numerous other studies, is the ability for a CV algorithm to determine if a face exists in an image and if so to isolate the face. Face normalization is necessary for face recognition as it standardizes faces based off pose and illumination to match the images in the training database. Face feature extraction is a technique used to find distinguishing features in a detected face to match to the features

found in the trained images. Finally, face matching is taking all of the data collected thus far in the process, comparing it to images in the training database, and finding its best match [20].

Image quality is an important concern when performing any CV task. Many studies have been performed to show the effects of image/video quality on face detection and recognition. Studies [19, 25, 26] go into detail of the effects that image quality has on these tasks. In [19] signal-to-noise ratio quality was varied for several common image databases using a JPEG compression algorithm. The results show that the image quality can be decreased to 20% of the original quality without having a negative effect on the accuracy of facial recognition, thus showing that CV algorithms are very tolerant to changes in image quality.

The computer vision library, OpenCV [27], provides three algorithms for face recognition. The following sections will detail the general approach of these algorithms.

2.1.1 Principal Component Analysis

Early methods of face recognition used relative locations and sizes of facial features to perform recognition [28]. These systems were found to be inaccurate and difficult to expand upon for further improvements. It was not until Kirby and Sirovich [24] completed their study of human face characterization based on the Karhunen-Loève procedure and principal component analysis (PCA) that the field began to experience exponential growth. This study provided one of the first methods of applying a pattern recognition methodology to faces in order to create a compressed approximation for computer algorithms to work with. The method effectively translated a three-dimensional (3-D) object into a two-dimensional (2-D) representation. The approach utilized the mathematical concept of eigenvectors to create representations of images referred to as Eigenpictures. This study showed that by encoding a face using mathematical concepts, patterns can emerge that allow the data to be compressed into a more convenient format. This allowed for computational speed ups through simplification of data and reductions in memory usage. Thus, allowing computers to be more capable of performing these types of CV operations. There

were several limitations to this study as it only considered small image collections and did not apply its concepts to face recognition.

Further expanding upon the PCA method of face approximation, Turk and Pentland [22, 23] came to the realization that the concepts from [24] could be applied to face recognition. In this context, the eigenvector encoded faces are referred to as Eigenfaces. The reduction in data size resulting from this encoding allowed the use of large image data sets for recognition while taking up less memory than previous face recognition implementations. The process begins by translating the training set to arrays of Eigenfaces. When recognizing a face the input image is transformed into an Eigenface and compared to the translated data in the training set. The use of facial patterns for recognition, rather than relative face information allowed for quicker, more accurate recognition over previous methods. Figure 2.2 shows some example Eigenfaces generated from the Honda/UCSD dataset [29, 30] along with the resulting image reconstruction. The differences in color in the Eigenfaces result from changes in lighting in the original video. We can see that the reconstructed image provides a good representation of a face, with only minor errors.

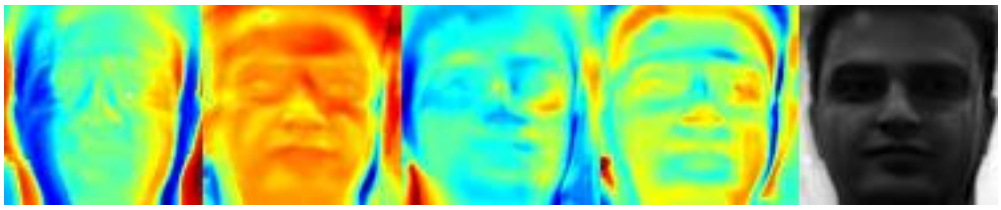

Figure 2.2: Example Eigenfaces

The PCA method allowed for significant advances in automated face recognition, however, there were several drawbacks that limited its applications. Changes in lighting and facial expressions caused accuracy to drop significantly [31, 32]. Lighting changes on a face are exceedingly common in real-world systems, especially in AVS systems. In such systems, a person may be moving between light sources, causing cameras to record images with varying lighting directions and intensities. In such a scenario, the accuracy of a PCA reliant system would drop dramatically.

2.1.2 Linear Discriminant Analysis

Advancing the concepts of the PCA Eigenfaces method, a new face recognition algorithm was developed utilizing linear discriminant analysis (LDA) techniques. The main concept behind both the PCA and LDA techniques is that a reduction in dimensionality of image data can allow CV algorithms to utilize the data more efficiently. However, the way in which the data is projected can have drastic effects on the resulting ability to recognize faces. With PCA, the projection does not eliminate variations due to lighting or facial expression. The LDA method seeks to utilize Fisher's Linear Discriminant to reduce the dimensionality of face data while also reducing the effects of lighting and expression on face recognition. In [31], reduction of dimensionality is done by first using PCA to perform an initial reduction in dimensionality, LDA is then utilized to reduce the dimensionality further and eliminate scatter based on lighting and facial expression. The resulting projections are referred to as Fisherfaces. Study [32] proposes a method in which only LDA is used to reduce the dimensionality of an image that face recognition is to be performed on. Both studies show significant increases in accuracy over PCA methods of face recognition. Figure 2.3 shows example Fisherfaces generated from the Honda/UCSD dataset with the resulting reconstructed face. We can see that lighting is completely removed from the variations in the image representations, as all of the representations have similar illumination levels. Interestingly, although LDA is typically associated with higher recognition accuracy than PCA, the reconstructed image for LDA has much more blur than for PCA.

Figure 2.3: Example Fisherfaces

2.1.3 Local Binary Patterns Histograms

While the PCA and LDA methods follow similar approaches to face characterization, new paradigms have arisen, challenging the way in which faces are described computationally. One such

method, utilized by OpenCV, is referred to as local binary patterns histograms. Unlike the PCA and LDA methods, which attempt to describe patterns of a face through dimensionality reduction utilizing vector mathematics, the LBPH approach to face description uses texture analysis to translate the face data into a usable format for recognition. This method is computationally more efficient and is more capable of handling variance in lighting that previous methodologies [33, 34].

Study [33] details, to our knowledge, the first implementation of a local binary pattern (LBP) based automated face recognition system. The basic principle of this LBPH face recognition method is to assign a binary mask to all of the pixels in an image, upon completion these masks can be reassembled into a representation of the original image, which can then be compared to a training database. The first step, finding the LBP, is done by dividing an image into N-number of cells, each with a predefined number of pixels. The algorithm then sequentially moves through every pixel in a cell and compares its intensity to the intensities of each of its nearest neighbors. Simple implementations use a 3 x 3 grid with the pixel under test being the center of the grid. Neighbors with intensities greater than or equal to the intensity of the test pixel are assigned a value of 1 and 0 otherwise. The eight values are then assembled into a descriptor byte for the test pixel. To perform recognition, a histogram is created from this data, which creates an overall description of the image. The generated histogram is compared to the histograms of the images in the training database, with the closest match being returned as the prediction. The results from [33] show that the recognition accuracy for the LBPH method is significantly higher than for a PCA based method. The results also show that small offsets in face detection locality do not have as large of an impact on LBPH as they do with PCA face recognition. Figure 2.4 shows several frames of a video from the Honda/UCSD dataset represented as local binary patterns. The figure shows that although the frames had varying lighting, the LBP images are unaffected. We can also see that the LBP images retain a large amount of detail of the faces, while reducing the total data needed to represent those images. The accuracy of LBPH-based facial recognition is studied further in [34], corroborating the findings of the previous study.

Figure 2.4: Local Binary Pattern Image Representation Examples

2.2 Automated Facial Recognition for Video Surveillance

Facial recognition still poses a technological challenge for most computer systems. Modern processors have allowed for increases in image processing speeds, however, under various conditions bottlenecks are still encountered. While original systems focused on recognition of still-images, where the time it takes for recognition is only of minor importance, there has been an increasing push for real-time facial recognition applications, especially in the field of AVS. Numerous studies [35-37] have been conducted to find ways to speed up face recognition. Study [36] proposed a hardware-based real-time face recognition system, as hardware-based approaches generally execute quicker than software-based approaches. The study develops a face classification system referred to as the frequency distribution curve (FDC) technique. In this method, image data is translated to the frequency domain and compared against training data using a standard variance vector. The FDC algorithm is then implemented using a field-programmable gate array (FPGA), allowing some customization to the application. The results showed that face recognition on a single image could be completed within 0.6 μsec and have an accuracy of up to 98.3%, allowing the possibility of highly accurate real-time face recognition. There are several drawbacks to the method proposed in [36], most obvious is that hardware implementations are more expensive and less

adaptable than software-based solutions. In addition, the method of recognition provided is only tested with one image database and recognition with more difficult databases was not explored. Furthermore, although the execution time is significantly decreased, the achieved accuracy is worse in most cases than more widely adopted face recognition algorithms.

Many studies [29, 30, 38] (and references within) have been conducted to determine the best method of face recognition on video sources. In [38] a method is proposed for increasing face recognition accuracy in video-based systems. Rather than using image sets for training databases and treating input video frames as unrelated images, this study places temporal importance on the sequence of video frames to provide a more confident match for the test video. This is done by modeling the training videos as a linear dynamical system [38]. The implementation of this system was able to achieve 90% accuracy for recognitions while using video databases that contain significant 2-D and 3-D variations in subject pose. In Study [29] a proposal is made for a face recognition system based on what are referred to as appearance manifolds, low-dimensional representations of a person's appearance. Video frames are compared to this manifold with the closest match being assigned the identity of the person in the frame. Temporal information from the video is used to adjust recognitions for pose and occlusion variations. While this method does not propose the ability to perform face recognitions in real-time, it does offer significant improvements to accuracy over previous methods, especially in the case of occlusion. Study [30] expands on the previous study by proposing an online training system for the appearance manifold method. An initial training database is created and as video sources are input for training the system uses machine-learning methods to incrementally adjust the database to provide better recognition accuracy. An online training approach was tested for the facial recognition work proposed in this thesis. Although this type of training approach is a feature that provides benefits to surveillance systems, it was ultimately decided that for the scenarios being tested an online training database would not be beneficial.

Video surveillance adds to the challenge of face recognition as such systems are intolerant to many of the shortcomings in face recognition. With surveillance systems, dropping frames is extremely undesired behavior. In scenarios where frame dropping occurs, possible threats are at risk of going unnoticed. For

this reason, if face recognition is to be continuously run in an AVS system, it must be performed in real-time. Facial recognition in AVS systems also faces the challenge of uncooperative test subjects. This is an obvious conclusion, but as most people are unaware or unconcerned with being under surveillance they do not show up in the video frames under optimal conditions for face recognition algorithms. There will be significant variances in position, pose, and illumination of the scene when a recognition needs to occur [20].

Studies [39-43] (and references within) have developed solutions to specific challenges for face recognition in video surveillance. The approach proposed in [39] details a method in which both the histogram of oriented gradients (HOG) and local binary pattern (LBP) methods are run in parallel on surveillance video. By combining both methods of face recognition, shortcomings in each algorithm (e.g. pose, lighting, and emotion variations) are covered by the complementary algorithm and overall accuracy increases substantially. However, this study does not run face recognition in real-time as the test system is not capable of running both algorithms in parallel in real-time. Instead, the detected faces are saved and recognition is performed later. This can have significant impact on the time to detect a threat. For these reasons, the work done in this thesis focuses on only using one recognition algorithm, LBP.

In [43] a full smart camera system is proposed for surveillance networks. In the study, FPGAs are paired with high-resolution image sensors to create custom smart cameras. These cameras are capable of performing CV tasks, including face detection. With face detection implemented at the camera side of the system, it is only necessary to send the cropped face data over the network. The central node is then able to perform face recognition on the cropped image data. In this configuration, the processing of images is distributed and the amount of data being sent over the network medium is significantly reduced. While this study makes use of custom-made smart cameras using FPGAs, many modern security cameras have the computational capability to perform face detection.

2.3 Accuracy-Based Cross-Layer Optimization for Video Stream Systems

Many studies [15-17, 44-47] have been conducted on the topic of cross-layer optimization of video streaming in wireless networks. Most work in this area has taken a distortion-based approach, relying on

relative distortions of the video streams to calculate adjustments in bandwidth allocation. Study [18] proposes an ACBO solution, with the intent to show that accuracy-based solutions are more effective than distortion-based solutions for bandwidth optimization. The study attempts to find the optimal fraction of a medium's effective airtime for each video source in a network such that the sum of weighted detection accuracy error is minimized. This sum is given by $\sum_{s=1}^{S} w_s \times accuracyError_s(r_s)$, where S is the network size, w_s is the importance for video source s, $accuracyError_s$ is the face detection accuracy error for video source s, and r_s is the transfer rate for video source s. Additionally, the optimization solution is constrained by the following conditions: the sum of airtimes of all video sources must not be greater than the effective airtime of the medium (A_{eff}), the transfer rate for each video source s is equal to the product of its physical rate (y_s) and fraction of airtime (f_s), and the fraction of airtime for each source must be in the range from 0 to 1. In order to obtain a solution for this formulated problem, Study [18] proposes two actions: (1) The characterization of a rate-accuracy function and (2) accurate estimation of the effective airtime of the medium.

In implementing an optimization solution for the problem formulation, first face detection rate-accuracy curves were developed for each of the face databases utilized. The rate-accuracy characterization provides a model for the relationship between video frame size and accuracy error of the face detection algorithm, therefore it is imperative that these curves be tailored to each database individually. The characterization allows the cross-layer optimization algorithm to calculate the expected error for a requested bitrate in the network, which in turn makes it possible to adjust the airtime for each node to improve their expected accuracy. The face detection accuracy error for video source s was characterized as $a_s(\frac{f_s y_s}{\tau_s})^{b_s} + c_s$, where f_s is the fraction of airtime for video source s, y_s is the physical rate for video source s, τ_s is the video frame rate for video source s. The rate-accuracy constants a_s, b_s, and c_s are assumed to be equal between all video sources as they share the same dataset.

Effective airtime (EA) estimation is an important part of the study, as the total airtime for all nodes cannot exceed the EA of the network. Estimating the EA with a high degree of accuracy allows better

optimization of the fraction of airtime given to each video source. The study implements a proportional-integral-derivative (PID) controller-based method for calculating the EA, allowing adjustments to EA in real-time using the packet dropping rate as the input. The study found that the proposed PID-based EA calculation converged quicker on network startup and reacted faster to network disturbances than other methods.

After formulating the rate-accuracy function and finding the effective airtime of the medium, (A_{eff}), the optimization problem was shown to be a convex programming problem and thus can be solved as follows:

$$f_s^* = \left(\frac{-\lambda^* \tau_s}{w_s a_s b_s y_s (y_s/\tau_s)^{(b_s-1)}}\right)^{(1/(b_s-1)}, \quad (1)$$

and

$$\lambda^* = \left(\frac{A_{eff}}{\sum_{s=1}^{S}\left(\frac{-\tau_s}{w_s a_s b_s y_s (y_s/\tau_s)^{(b_s-1)}}\right)^{(1/(b_s-1))}}\right)^{(b_s-1)}. \quad (2)$$

Study [18] uses Equations (1) and (2) to determine the fraction of airtime for each video source in the network. In the implementation the value for λ^* is calculated by the server node. This value is then sent to the corresponding video source to allow it to determine its fraction of the effective airtime, thus providing the necessary information for adjusting the bitrate of the sent video. In conjunction with the fraction of airtime calculation, the allocation algorithm also uses the link-layer parameter transmission opportunity duration limit (TXOP limit) to adjust timing of sent packets. The TXOP limit is defined as the time required to send all of the packets belonging to a single frame over the network, taking into account all of associated overhead involved. Study [15] proposes the model for TXOP limit. The formulation of TXOP limit takes into account the MAC and physical layer parameters for a more accurate calculation.

The study also proposed a bandwidth pruning method that is used to reduce network load and power consumption in networks. Due to the slope of the rate accuracy curves, at higher frame sizes significant reductions in bandwidth can be expected with only small increases in accuracy error. The study experiments with several different percentages to show the effects on face detection accuracy, network load,

and power consumption. The study found that the ACBO solution outperformed prior distortion-based methods in every tested metric (accuracy, network load, and power consumption). The proposed bandwidth pruning method was found to achieve similar accuracy to distortion-based solutions, but with a 45% reduction in network load and power consumption.

A great deal of work is currently ongoing to advance the topics covered in Study [18]. In addition to this thesis, Study [48] developed an experimental AVS system utilizing the cross-layer optimization solution from [18].

Chapter 3 Proposed Work

This chapter details the work proposed by this thesis. It covers in detail all of the work that was completed to build upon prior studies and provides a discussion on proposed enhancements. The chapter covers the motivation behind this thesis as well as the work done to fit the rate-accuracy characterization to the data collected for face recognition. The chapter will discuss the details of the system shown in Figure 3.1. The chapter also covers, in detail, the implementation of the proposed FFmpeg based streaming solution as well as the justification for the chosen video codec. Finally, this chapter describes proposals for two enhancements to the ACBO solution and the details of the work done to implement these enhancements.

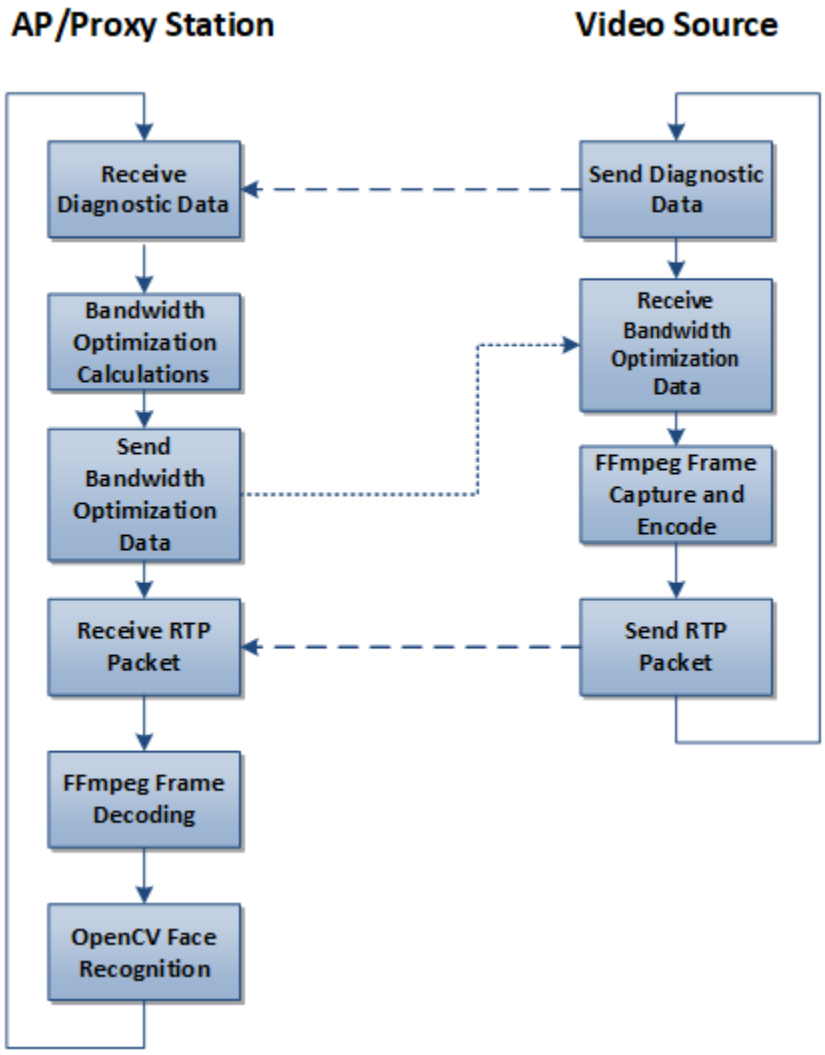

Figure 3.1: Overview of an ACBO Solution Implementation

3.1 Motivation and System Description

3.1.1 Motivation

Although prior work on the topic of cross-layer optimization provided substantial improvements to network load and power consumption, the topic of face recognition was never discussed. All of the previous studies focused making improvements to various other CV tasks, most prominently being face detection.

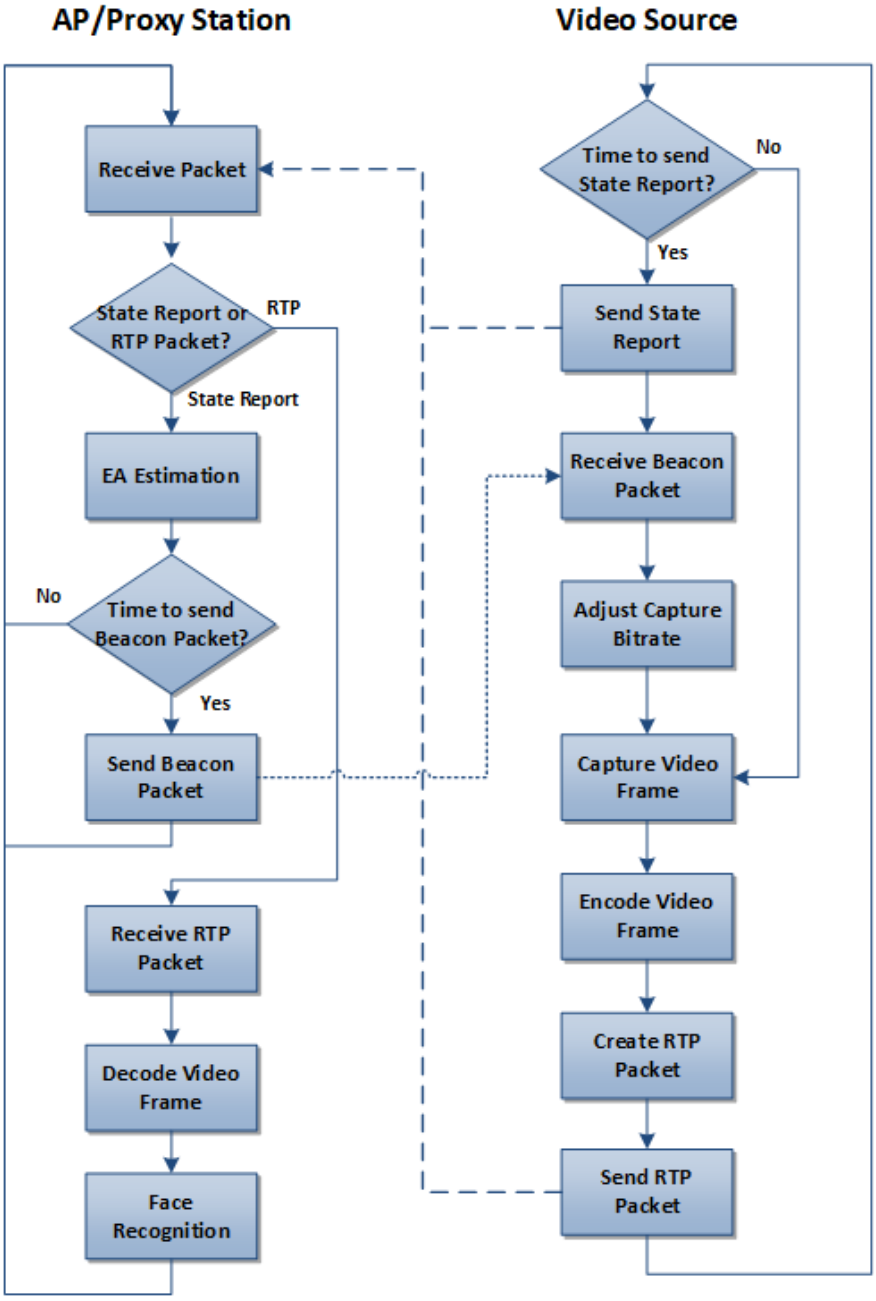

Figure 3.2: Detailed Overview of an ACBO Solution Implementation

While face detection is a valuable technology to implement within AVS systems, it does not allow the identification of threats. The motivation for this thesis comes from the desire to implement and analyze the effectiveness of the ACBO solution when applied to face recognition. We also felt that further improvements could be made in order to increase the effectiveness of the ACBO solution.

3.1.2 System Description

Using the same mathematical model applied in [18], we develop a system that focuses on maintaining face recognition accuracy in AVS networks. As such the only changes we make to model are the values of the equation constants, which are primarily dependent on the rate-accuracy curve. While the basis of the ACBO implementation remains the same, significant changes were made to allow the use of a full video streaming client and facial recognition. In addition, extensive work was done to verify that the model proposed in previous work was valid for the system tested in this thesis. Figure 3.2 shows the details of our implementation of the ACBO solution when applied to facial recognition. In the implementation, the tasks are split between the AP/proxy station and the video sources. While the system is in steady-state operation the video sources monitor network conditions and send this information to the proxy station in the form of a state report packet. The proxy station collects data from state reports, which it uses in the calculation of the effective airtime. The effective airtime estimation is sent to the video sources as a beacon packet. Each source calculates its share of the airtime and then translates it into a transmission bitrate. If this calculated bitrate differs by more than 100 bits per second (bps) from the previous bitrate, the video encoding is updated with the new bitrate. The differential of 100 bps was selected as it was small enough as to not affect the accuracy of the face recognition, but large enough to stop constant adjustments to the encoding rate. Making changes to the encoding rate is a time-consuming task that is best to be avoided unless completely necessary. The video sources then encode the captured video frames at the desired rate and send them over the network in RTP packets. The AP/proxy station receives the RTP packets, decodes them, and performs face recognition. This process repeats as long as the network continues running, with beacon and state report packets sent at predefined periodic rates. The previous work on this topic did not

consider a rate-accuracy characterization that directly relates bitrate to accuracy; it instead focused on the relation between frame size and face detection accuracy. While frame size does affect the bitrate of a stream, it is an unnecessary abstraction when the ability to directly control bitrate exists. Given that we are directly controlling bitrate, rather than frame size, and are focusing on face recognition, opposed to face detection, we detail our method for calculating the new rate-accuracy curve in the below section.

3.2 Rate-Accuracy Characterization

The general optimization problem for accuracy has been developed by previous studies [15, 18] and is used in its entirety for this study; since the rate-accuracy formulation is the same as previous work we only have to obtain accuracy values for our chosen dataset in order to acquire meaningful constants to use when calculating the expected rate-accuracy values in our simulated network. The methodology for rate-accuracy characterization is covered in more detail in Section 3.2.3. This model characterizes the relationship between streaming bitrate and accuracy error in the facial recognition algorithm utilized by the proxy station in our network. As prior work has characterized detection algorithms [18], we build upon this work to characterize recognition accuracy. We also consider the work done in [49] when performing our rate-accuracy characterization, in the study the Honda/UCSD database is utilized in full to create a rate-accuracy curve for dataset. We build upon this by testing the accuracy error for a larger set of bitrates in

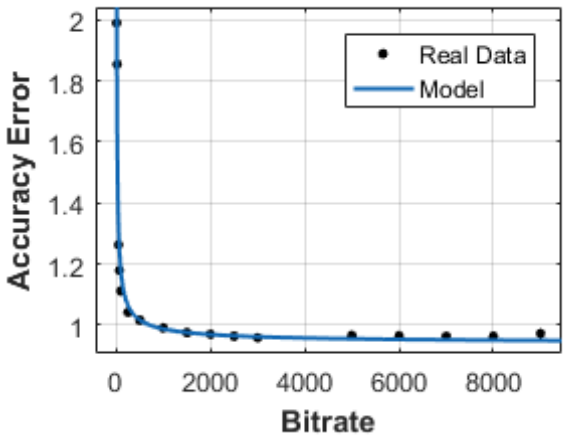

Figure 3.3: Rate-Accuracy Characterization Model

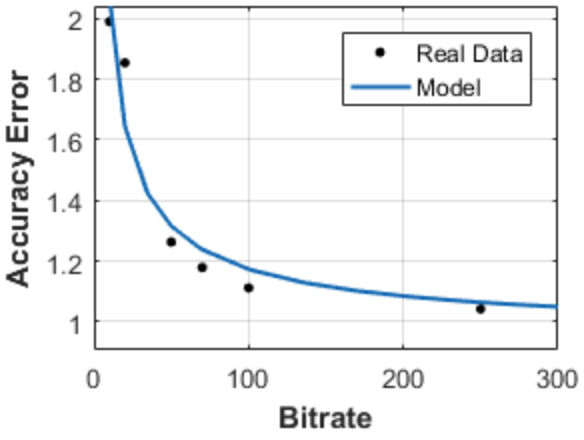

Figure 3.4: Rate-Accuracy Characterization Model Zoomed

addition to verifying the previous results. We have chosen not to use the full Honda/UCSD dataset for this thesis; we remove the videos that have outlying accuracy values.

3.2.1 Facial Recognition Algorithm Analysis

This thesis is not focused on the variances in accuracy between different recognition algorithms; we only seek to optimize bandwidth in an AVS network, thus any algorithm that provides sufficiently reliable accuracy is acceptable. For this reason, we decided to only characterize one of the face recognition algorithms offered by OpenCV, rather than all three. All of the facial recognition algorithms use the Viola-Jones algorithm [50, 51] to detect the faces for recognition. We perform analysis on the OpenCV facial recognition algorithms in order to determine which one best fit for our test system. We consider the three face recognition algorithms implemented in OpenCV: Eigenfaces, Fisherfaces, and Local Binary Patterns Histograms (LBPH). We consider two metrics for our tests of the algorithms: recognition accuracy and training time. Table 3.1 summarizes the results of our analysis on the three algorithms. For our purposes, the best algorithm to use is one that provides the highest accuracy possible for our chosen dataset while also performing training in a reasonable amount of time. Higher overall accuracy will provide the most consistent results when testing the ACBO solution as we can see larger variations in accuracy due to the changes in sending rate. In addition, although real-world AVS system implementations may be tolerant to long training times since training is only run occasionally, for our simulated method we ran tests frequently, each of which required the training to be run. In order to complete the testing in a timely manner we selected an algorithm that could quickly generate its training data.

Of the three algorithms tested, our analysis shows that the Eigenfaces method provided the lowest accuracy only achieving 35.57% accuracy with the test dataset. The Fisherfaces algorithm offers the best accuracy at 52.42%, with LBPH achieving slightly lower accuracy at 52.03%. However, our results show that although the Fisherfaces algorithm provided slightly better accuracy overall, the LBPH method ran significantly faster in both training and recognition. The training took only 59 seconds with LBPH compared to approximately eight and a half hours for Fisherfaces. With the accuracy difference between

LBPH and Fisherfaces being less than 1% and the training time difference being so great, the LBPH algorithm is the best fit in our scenario for characterization and simulation.

Table 3.1: Comparison of OpenCV Facial Recognition Algorithms

Algorithm	Correct Recognitions	Total Faces	Accuracy	Training Time (sec)
Eigenfaces	4106	11544	35.57%	24407
Fisherfaces	6051	11544	52.42%	29468
LBPH	6007	11544	52.03%	59

3.2.2 Codec Choice

With earlier work a limited set of video codecs were utilized due to the types of face databases used. The previous work utilized image file databases, rather than databases that included video files, which made it difficult to use modern video codecs. For this study, we use the Honda/UCSD database, which contains sets of RAW video files. This allows us to utilize modern codecs for encoding the video data transmitted over the AVS network. We considered two different codecs, H.264 and H.265. Although H.265 is a newer standard and provides better compression than H.264, it is much more computationally intensive and its development within FFmpeg is still ongoing, meaning we would be unable to test with a full implementation of the codec. In addition, adoption of H.265 is much lower at this point than H.264. For these reasons, we believe H.264 was the best choice for use as the video codec in our AVS system.

3.2.3 Characterization Methodology

We perform our analysis of the rate-accuracy relationship using the Honda/UCSD video database. Utilizing the full resolution of the video (640 x 480), we achieve a large range of bitrates to test. To collect the rate-accuracy error data we first create a method for properly training our face recognition algorithm. As the OpenCV algorithms do not allow for video as an input to the training session, we devise a strategy in which we first run the training videos through the Viola-Jones face detection algorithm. From here we get the isolated frames from which we can detect faces. We then crop these images so only the detected faces remain; these images are used as inputs to the training algorithm. One benefit to using the

Honda/UCSD database is that all video frames should contain at least one face. This gives us many faces to train with, even though we are unable to detect every face due to the amount of pose and lighting variation of the subjects in the video. We run face recognition on all of the test videos in the database at their original bitrate. Using the accuracy results from this test, we acquire the best possible accuracy for each video. We remove any of the videos with an outlying accuracy in order to achieve more consistent accuracy results at the varying bitrates.

Table 3.2: Face Recognition Accuracy at Various Bitrates

Bitrate (Kbps)	Total Faces	Faces Recognized	Positive Index	Negative Index	Accuracy Error
10	11544	60	0.0052	0.9948	1.9896
20	11544	841	0.0729	0.9271	1.8543
50	11544	4251	0.3682	0.6318	1.2635
70	11544	4736	0.4103	0.5897	1.1795
100	11544	5124	0.4439	0.5561	1.1123
250	11544	5525	0.4786	0.5214	1.0428
500	11544	5675	0.4916	0.5084	1.0168
1000	11544	5824	0.5045	0.4955	0.9910
1500	11544	5911	0.5120	0.4880	0.9759
2000	11544	5942	0.5147	0.4853	0.9705
2500	11544	5980	0.5180	0.4820	0.9640
3000	11544	6007	0.5204	0.4796	0.9593
5000	11544	5963	0.5165	0.4835	0.9669
6000	11544	5972	0.5173	0.4827	0.9653
7000	11544	5978	0.5178	0.4822	0.9643
8000	11544	5982	0.5182	0.4818	0.9636
9000	11544	5929	0.5136	0.4864	0.9728

Utilizing FFmpeg, we encode the videos using the H.264 codec at varying bitrates ranging from 10 kbps to 9 Mbps. We chose this range as it covers the range of bitrates that are achievable by the video sources in the simulated network. We use the same method from studies [15, 18] for determining the rate-accuracy relationship. In these studies two metrics are used to calculate the accuracy error, the *positive index*, which is the number of correctly recognized faces divided by the total number of faces, and the *negative index*, which is the total number of incorrectly recognized faces divided by the total number of faces. Table 3.2 shows the collected data for face recognition at various bitrates along with the positive and negative indexes and the calculated accuracy error. As expected, with increases in bitrate the accuracy error

decreases. We can see from the data that accuracy error does not decrease linearly with increases in bitrate. The accuracy error decreases dramatically at smaller bitrates and then at about 500 Kbps the error begins to level off, with further increases to bitrate having little effect on the error.

From our accuracy results for each video in the dataset, we are able to determine the average accuracy, the positive index, and the negative index. Using this we can calculate the accuracy error as follows: *accuracyError* = (1-*positiveIndex*) + *negativeIndex*. Table 3.2 shows the results of our accuracy testing. This data is curve fit in order to create a formulaic representation of *accuracyError*, the rate-accuracy model is represented as follows:

$$accuracyError = a \times Z^b + c, \tag{3}$$

In Equation (3), Z represents the bitrate of the video and a, b, and c are constants determined through the curve fitting process. These constants will vary depending on the codec and dataset used. The constants used to fit Equation (3) to our real data are shown in Table 3.3.

Table 3.3: Rate-Accuracy Model Constants

Constant	Value
a	5.48
b	-0.6845
c	0.0306

3.2.4 Confirming the Rate-Accuracy Model

To corroborate the rate-accuracy characterization we measure two more statistics at varying bitrates: peak signal-to-noise ratio (PSNR) and mean structural similarity (MSSIM). We expect that varying the bitrate of a video feed will directly affect video quality and recognition accuracy, as it is the basis of the ACBO solution. While our primary concern is the effect of changes to the bitrate on face recognition accuracy, we can use its relationship with video quality to check our rate-accuracy model. As we know from [19], the relationship between video quality and face recognition accuracy follows a logarithmic curve, with accuracy initially increasing rapidly and then leveling off after quality reaches a critical point. To check that our data follows this same trend we use OpenCV to determine the quality of

our videos over the same range of bitrates used in the calculation of our rate-accuracy curve. We collect data for the two quality metrics (PSNR and MMSIM) with the Honda/UCSD dataset. Figure 3.5 shows the relationship between PSNR and bitrate. The figure shows that while the PSNR does not level off in the same way as face recognition, the behavior below 1000 Kbps is identical, with quality dropping significantly below that point. MSSIM behaves the same, as is shown in Figure 3.6. All three color-channels differ significantly in quality from the original video below 1000 Kbps. Above 1000 Kbps the quality remains level. The differences in quality between the red, green, and blue color channels were not a concern as the system converts the video frames to grayscale before performing recognition. Thus, our only concern is that the qualities of each channel follow the same trend. With this data, we can confirm that the rate-accuracy model developed for the Honda/UCSD dataset is valid.

3.3 Face Recognition Implementation

Paramount to the work done in this thesis is the implementation of an effective face recognition system. We develop a system by utilizing the OpenCV library, as it has already has robust implementations of several face recognition algorithms. After consideration of the three algorithms offered by OpenCV, we have chosen the LBPH algorithm as the best fit for the system we use in this thesis. To perform face recognition we first develop a method for creating a training database to compare our input faces against. The method for training we use in our simulated AVS system is identical to the approach discussed in

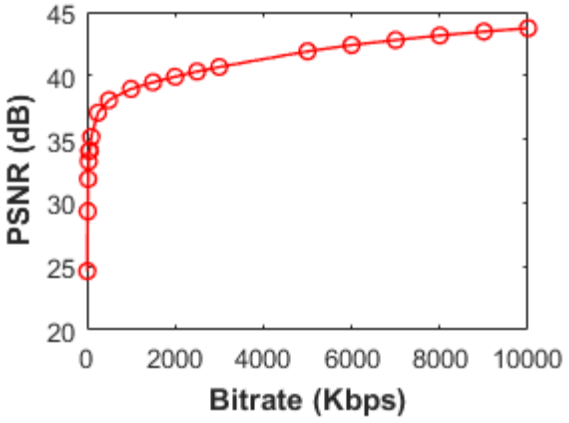

Figure 3.5: Relationship between PSNR and Bitrate for Tested Datasets

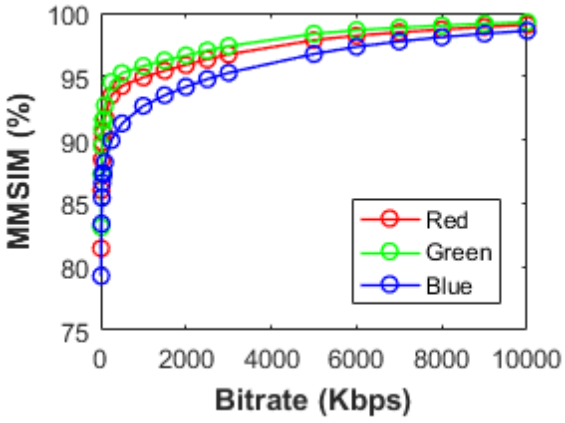

Figure 3.6: Relationship between MSSIM and Bitrate for Tested Datasets

Section 3.2.3. As discussed in Section 2.1 there are four main steps for performing face recognition: face detection, face normalization, face characterization, and prediction. We perform each of these discrete steps manually using OpenCV library functions. Within our implementation, after a frame is sent to the proxy station, we perform face detection using the OpenCV implementation of the Viola-Jones algorithm. This gives us unprocessed isolated faces with which we then normalize. The normalization process is important to achieving consistent recognition results. With the normalization process we first convert all face images to grayscale. Although it is not necessary to perform this step before using the LBPH recognition algorithm, we have found that performing this step results in more consistent recognition results as it removes any inconsistencies due to color variances. To further normalize the face images we resize all images to a consistent size, in our case we have found that a size of 75 x 75 pixels provides high enough accuracy while also executing quick enough that recognition is still able to run in real-time. The face characterization process is run using the OpenCV local binary patterns algorithm. The algorithm determines the binary patterns and creates a histogram of the data. The predictor then compares the histogram to the histograms in the training database. The best match in the training database is found and the identifier the match is returned as the prediction for the input face.

3.4 Video Streaming System

3.4.1 FFmpeg Implementation

One focus of this research was to fit prior work into a system that more closely resembles a real AVS system. While previous studies mimic the basic networking functionality of an AVS system, they did not address real-world conditions for the functionality of the video sources and proxy station. While this is useful in testing certain aspects of a network, without a full implementation of an AVS system it is not possible to know that the solution works as we expect in all scenarios. While this study does not test on a real AVS system, we do create a simulated network that matches real-world systems as close as possible. Utilizing FFmpeg, we implement a full video encoding and decoding system in the simulated AVS network. With the FFmpeg developer libraries, we are able to encode videos in sim-time at the desired bitrate

specified by the network. Table 3.4 summarizes the FFmpeg parameters used for encoding. In a real AVS system, the input video feed would come from the video sensor of the associated camera. In our simulated network, we did not have the ability to have a direct input from a sensor; however, to overcome this obstacle we use RAW format video from the Honda/UCSD dataset input directly to the video sources in the simulated network. From there we can encode the frames in the simulated time domain, place the data in packets, and send it to the AP/proxy station. We fully implement the real-time transport protocol in the application layer of the video sources and proxy station in order to transmit the video data. In the proxy station, we again utilize FFmpeg to decode the data as it is received from each video source before passing the decoded video frames to the facial recognition algorithm.

Table 3.4: Summary of FFmpeg Parameters

Parameter	**Value**
Codec Standard	H.264
Bitrate	Adjustable based on optimization algorithm
Bitrate Tolerance	1000 bps
Frame Dimensions	640 x 480 pixels
Frame Rate	20 frames/sec
Group of Pictures (GOP)	40 frames
Maximum B Frames	0
Pixel Format	YUV420P
Encoding Preset	Fast

3.4.2 Bitrate Control

As mentioned, prior studies did not perform the encoding at run-time; this meant that discrete bitrate levels had to be created prior to simulations. In order to achieve the best performance as defined by the rate-accuracy characterization, it is necessary to have the ability to perform fine control of bitrate. What can be seen in Figure 3.3 is that a change in bitrate at the lower end of the curve can have drastic effects on the achieved accuracy. The prior study had 100 discrete bitrate levels spread evenly across the full range of tested bitrates (10 kbps to 9 Mbps). This coarse level of granularity causes problems when the optimization algorithm requests lower bitrates. At the lower end of the rate-accuracy curve, small changes

in bitrate can have substantial effects on accuracy. Evenly distributing the levels also has the problem of skewing too many levels at the higher bitrates where the need for fine control is not as necessary. Using FFmpeg makes it is possible to achieve any bitrate in our range. While FFmpeg does have a tolerance associated with its bitrate setting capabilities, it is much more precise in its bitrate adjustments than the methods of prior work.

Table 3.5: FFmpeg Requested Bitrate vs. Actual

Requested (Kbps)	Actual (Kbps)	Difference
10	13.4	34.00%
20	22	10.00%
30	30.5	1.67%
40	41	2.50%
50	49.5	-1.00%
70	69.1	-1.29%
100	98.1	-1.90%
250	245.2	-1.92%
500	497.8	-0.44%
1000	1005	0.50%
1500	1523	1.53%
2000	2030	1.50%
2500	2580	3.20%
3000	2993	-0.23%
5000	4990	-0.20%
6000	5997	-0.05%
7000	6985	-0.21%
8000	7995	-0.06%
9000	8968	-0.36%
10000	9965	-0.35%

Table 3.5 shows a comparison of requested bitrate values to the actual values that FFmpeg achieves. The table shows that even though our bitrate tolerance value is set to 1 Kbps, FFmpeg is unable to achieve target values that precise. Although this variance is relatively small at higher bitrates, we can see from the table that at lower bitrates the difference between requested and actual is comparatively high. What can also be seen in Table 3.5 is that except for very low bitrates FFmpeg tends to achieve a lower bitrate than requested; we consider this behavior in our tests, as there is the potential to affect face recognition accuracy

if the requested bitrate based on the rate-accuracy curve is not achieved. Even though we see some issues with the capability of FFmpeg to achieve desired bitrates, this method is significantly more precise than the method used in previous work. The advantages of finer control over bitrate outweigh the negatives of increased complexity, as it is very important that the system be able to provide as precise a bitrate as possible.

3.5 Codec Implementation

With the implementation of full encoding and decoding at the video sources and proxy station, we were also afforded the ability to easily utilize available codec implementations in FFmpeg. As H.264 has become the de facto standard by which video is encoded in a multitude of applications, it made sense to use it as our test codec. Previous work focused on sending MJPEG video over the network due to the limitation inherent to choosing image datasets over video datasets. While MJPEG has been a commonly used encoding standard for video surveillance systems, this method provides little in the way of compression compared to other codec standards. With advancement in wireless technologies for video surveillance, it is becoming increasingly important to utilize modern encoding standards that allow video quality to remain the same with a greater reduction in bitrate compared to older standards. In addition, with the advancement of processors, the computational overhead associated with H.264 over MJPEG has become less of a concern. It is for these reasons that we utilize H.264 as the codec for video data transmitted over our simulated network. The change to a more modern codec requires careful consideration of the effects on the ACBO solution. As the proposed method for rate-accuracy characterization from [15] is dependent on the relationship between accuracy and bitrate, we recalculate the curve fitting constant values due to the increased quality of H.264 encoded video at lower bitrates. The rate-accuracy model forms the basis of the ACBO solution, for this reason we confirm that the rate-accuracy curve does not resolve in such a way that the optimization would no longer provide any benefit in the network. With video quality increased at lower bitrates compared to MJPEG, we were unsure if the data would follow the same characterization as previously tested encoding methods. What we observe from testing is that although H.264 encoded video

has a smaller data size than previous codecs, this does not have an effect on the trend that the rate-accuracy curve follows.

3.6 Proposed Bandwidth Capping Method

In this thesis, we propose several enhancements to the ACBO solution. The first enhancement is similar in principle to the bandwidth pruning method, referred to as the bandwidth capping method. We develop this method by following the hypotheses tested in studies [25, 26]. These studies focus on the relationship between video quality and face recognition and detection. The results from these studies shows that a point exists where increases to quality no longer have a significant effect on face detection and recognition accuracy. As we have shown in Section 3.2.4 that the quality-accuracy relationship follows a similar trend to the rate-accuracy relationship we can deduce that there is also a point on the rate-accuracy curve beyond which no significant increases to face recognition accuracy are observed. From the rate-accuracy curve shown in Figure 3.3 we see this exact scenario, as bitrate increases the accuracy error rapidly levels off. Beyond a certain bitrate improvements to error are minimal. This suggests that there is an upper limit in terms of video quality when performing face recognition. Beyond this limit, all increases in bandwidth are superfluous.

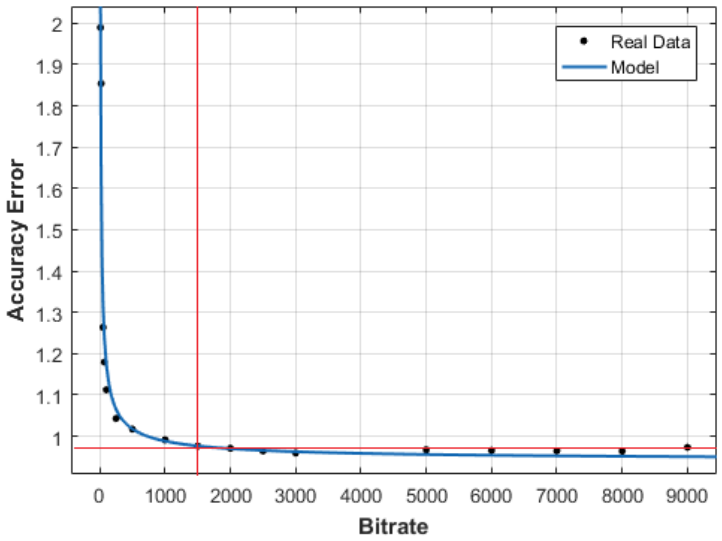

Figure 3.7: Bandwidth Cap Bitrate Determination

Significant reductions can be made in network load and power consumption by imposing a cap on the bandwidth of each video source. It is important to base this value on the characteristics of the rate-accuracy curve; setting the value too low negatively affects the face recognition accuracy of the system, potentially causing threats to go undetected. Setting the limit too high results in unnecessary network load and power consumption. Figure 3.7 shows the way in which the capping bitrate is determined for the system utilized in this thesis. The figure shows that the rate-accuracy curve deviates from the real data at higher bitrates. The data follows a horizontal trend, whereas the rate-accuracy curve continues trending toward zero. Based off of the collected data, we know that the accuracy error does not approach zero as bitrate increases. To determine the best limit to place on the bitrate without affecting the accuracy, we draw the horizontal trend line that the data follows at high bitrates. We then observe where the rate-accuracy curve crosses that line; the vertical red line in Figure 3.7 denotes this point. For the system under test in this thesis, we observe that the rate-accuracy curve crosses the horizontal trend line at 1.5 Mbps, we place our limit at this point. As the application rate has an inverse relationship with the number of sources in the network, this optimization is only effective with smaller network sizes. Beyond a certain network size the average application rate drops below the limit and thus the limit is no longer necessary.

3.7 Proposed Distributed Face Cropping Method

The other enhancement we implement in this study is an adaptation of the system detailed in [43]. In [43], rather than perform the face detection for all video sources at the proxy station, it is recognized that it is possible to distribute these tasks to the video sources. Face detection has relatively low computational requirements due to the increasing capability of embedded systems and the efficiency of the Viola-Jones algorithm. This allows the possibility of performing these tasks at video sources, which then creates the ability to send only the relevant image data (faces) over the network rather than sending the entire captured frame. As the system under test already requires the use of “smart” video cameras, sources that have advanced processing capability, the jump to performing face detection is not significant. Performing face detection at the video source allows for significant enhancement in terms of data being sent over the

network. As the AVS system is mainly interested in faces to recognize, it is not necessary to send an entire video frame to the proxy station. In the original implementation of the ACBO solution, the proxy station looks for faces in received frames and disregards the remaining information in the frame that had been sent. Study [43] suggests cropping the video frames so that the data sent to the centralized server only contains face images. The central server then performs face recognition on these cropped images. As face recognition is a computationally intensive task it is still necessary to perform at the server. Figure 3.8 outlines the implementation of this distributed face cropping system in the test network for this thesis. In the network, the video source (camera) captures a frame and performs face detection. If a face is found, the smart camera crops it from the original frame and encodes, packetizes, and sends the resulting image to the proxy station. After receiving all necessary packets, the proxy station reassembles the encoded frame, decodes it, and performs face recognition. The expectation of this type of system is that power consumption should remain similar to the non-optimized network, as the system still performs face detection, just at a different point in

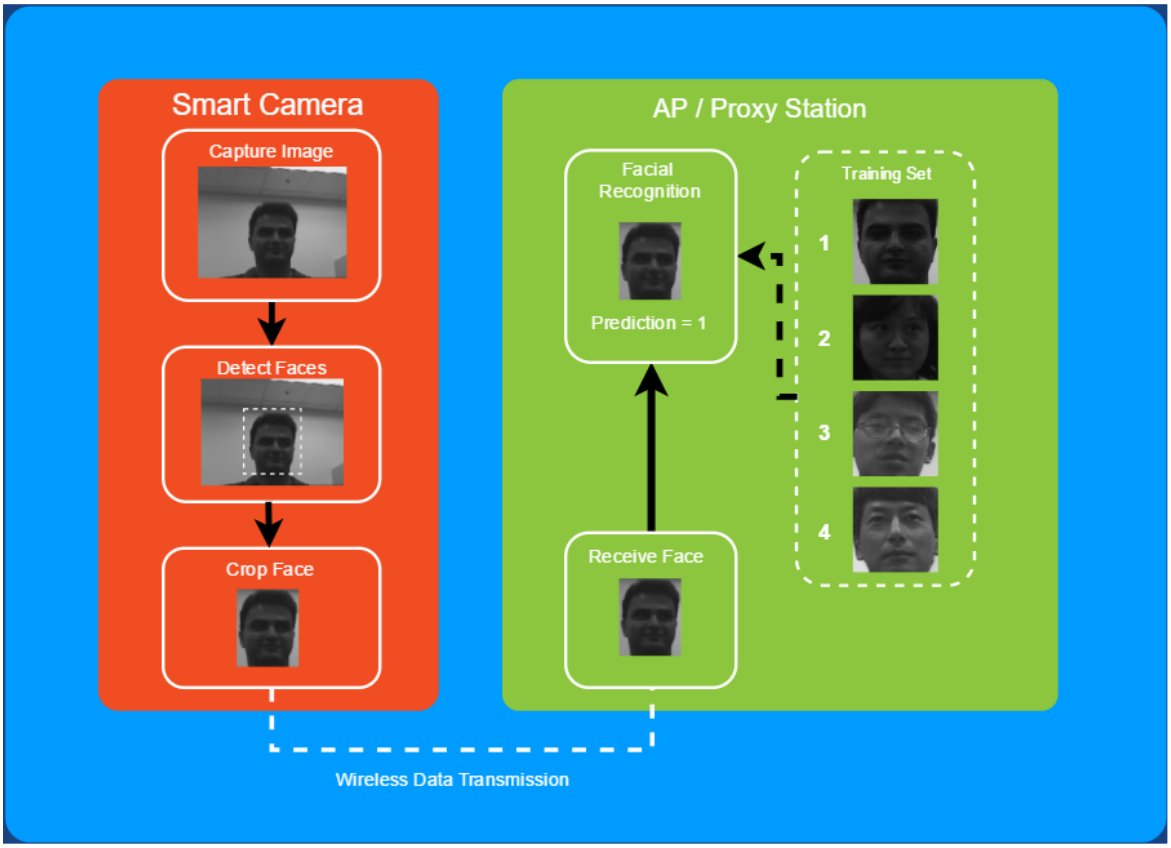

Figure 3.8: Distributed Face Cropping System

the process. As only the face image data are being sent over the network, the system maintains a higher overall image quality because even with the higher quality, the image cropping still results in less data being sent overall.

Chapter 4 Performance Evaluation Methodology

This chapter covers the work performed to evaluate the efficacy of the ACBO solution when applied to face recognition. It provides details of the evaluation of the effective airtime solution and the setup of the simulated test environment. This chapter also describes the development of the video streaming system, discussing the rationale behind the necessity of such a system in fully testing the work done. Weighted and non-weighted approaches to the ACBO solution are discussed, including how they are utilized in the test environment. Finally, this chapter covers the methodology used in choosing performance metrics for comparison against previous bandwidth allocation solutions.

Table 4.1: Test System Summary

Component	**Value**
Processor	Intel Core i5-3570k, 4-cores @ 3.40 GHz
Memory	16.0GB DDR3 1333 MHz
Motherboard	Gigabyte Z77X-UD3H
Storage	Samsung 840 Pro 256GB SSD
Operating System	Windows 8.1
Simulation Software	OPNET Modeler 14.5

4.1 System Setup

The evaluation of the ACBO solution is performed on the system summarized in Table 4.1. This system provides sufficient processing ability and memory to perform CV and video encoding tasks in conjunction with the modeled network simulations. A graphics card is not necessary for our testing as the tasks were run completely on the processor. A solid state drive (SSD) is used to reduce the amount of time spent accessing video files, since the files are too large to store in RAM they need to be located on another storage medium. Windows 8.1 is used as it was the latest release of the Windows operating system at the time testing and development of this thesis began. We use OPNET Modeler 14.5 as the tool for simulating our AVS network models, this version provides all of the functionality that is necessary for evaluating the ACBO solution.

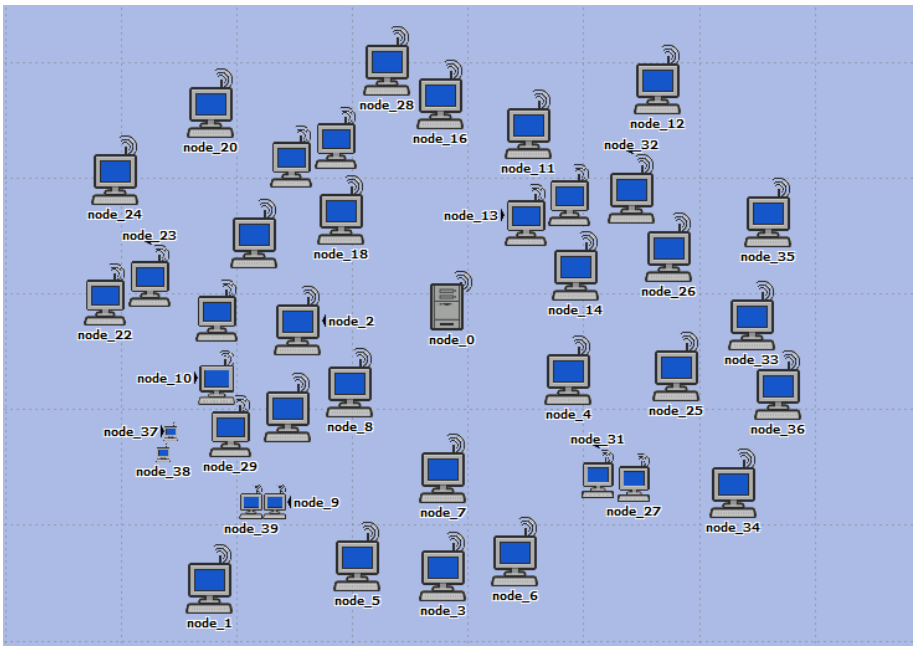

Figure 4.1: An Example of an OPNET Simulated Network

To evaluate all aspects of the ACBO solution when applied to face recognition effectively, our network environment is setup in the OPNET Network Modeler. OPNET allows us to assess the ACBO solution, the effective airtime algorithm, and optimization enhancements across multiple network configurations quickly and efficiently. This also allows us to control network conditions that are difficult to modify in a real-world network, allowing us the ability to fully test how our optimization solution reacts under varying circumstances. Figure 4.1 shows an example of an OPNET network model; it shows a network in which there are 40 video sources streaming to the AP/proxy station (node_0).

We distribute the implementation of the ACBO solution among both the video sources and the AP/proxy station. This allows monitoring of parameters from both the clients and server, resulting in a higher precision effective airtime calculation. This implementation methodology also provides additional benefits by distributing the computational load across many nodes. As mentioned previously, the H.264 standard is used to encode video traffic due to its current ubiquity. Previous work utilized image databases to create video streams to send over the simulated network. The method for interacting with video-based datasets is more complex to implement and execute than for image-based datasets. As our database for this study consists of actual video content we have to implement a method for interaction. Our investigation

considers several methods for interacting with video streams in order to determine the best approach. In one implementation, we attempted to utilize the FFmpeg implementation within OpenCV to create the video data for streaming; however, this approach did not give us fine enough control over the encoding parameters. We then attempted to utilize the FFmpeg command line externally to the simulation to encode videos as input using the system-in-the-loop functionality of OPNET. This method was not effective as simulations run much slower than real-time causing a disparity between a simulation and the FFmpeg process, resulting in the video sources being unable to receive all of the frames of the video file. This finally led to employing the FFmpeg developer library directly in our video sources to encode the video in simulation-time, this allows the system to sequentially encode and transmit of all frames from the video without the need to interact with multiple programming libraries. Utilizing the FFmpeg developer library was not initially chosen because of the complexity involved with integrating it with the simulated environment. As we were left with no other feasible options, we did extensively modify the OPNET simulated network to work with this library. Additionally, as part of application layer of the proxy station we implement a realistic video streaming client. The client is able to take RTP packets incoming from the video sources, reassemble these packets into video frames, and perform any error concealment necessary to mitigate the effects of packet loss.

4.2 Effective Airtime Estimation Evaluation

To fully determine if the ACBO solution was a valuable tool in assisting with face recognition we have decided to test the entire solution, including the effective airtime estimation. Major changes have been made to the implementation of ACBO in order to accommodate our chosen facial recognition algorithm. Face recognition algorithms are magnitudes more computationally intensive than face detection algorithms, we consider this when looking at effective airtime as computational limits have an effect on the amount of data able to be processed and thus the effective airtime. In our scenarios the concern was that the increased computational bandwidth usage would cause a larger number of missed packets, requiring the effective airtime to be adjusted in response. The previous work on this topic did not fully test scenarios in which high

computational loads exist. We test this twofold, in addition to the facial recognition algorithms that place a high computational load on the proxy station; we also encode and decode video within the simulated system itself. Previous work did not have to perform encoding as the data streamed over the network consisted of preselected JPEG images that were sent over the network to simulate a video feed. This scenario results in minimal computational load during the simulation, as the video sources need only to point to the correct image file and populate packets with the pre-encoded data. The decoding task was performed at the proxy station, however, decoding the JPEG images is trivial for a modern processor and this function is already built in to OpenCV which was being utilized for face detection, allowing optimizations to be made in the process of decoding the data into the correct format for face detection. We, however, have chosen to perform all of the encoding tasks during the simulation; this requires much more complex methods for adjusting bitrate and packetizing the data. Although our system is more complex there are major benefits, most importantly the simulated system now accurately models a real AVS system. In addition, we now have much finer control over the bitrate being sent by the video sources.

4.3 Video Streaming Implementation

With this thesis, we focus on creating a more realistic video streaming system compared to previous implementations. While previous systems utilize image databases to create streams of simulated video with MJPEG encoding, this study utilizes video databases and creates streams with H.264 encoded input videos. This decision has led to several challenges in implementing the system successfully in our simulated networks. As we are not using real cameras as inputs, we need a way to simulate the behavior of such devices. While encoding and streaming the video files is straightforward, we found that the method for assigning the videos to each source had effects on the consistency of the facial recognition accuracy even with sufficiently long simulations. For this reason, we needed to find an assignment method that produced repeatable facial recognition accuracy results.

Through experimentation, we found that the method outlined in Figure 4.2 provided consistent recognition accuracy results across multiple simulations. In this method, we begin by determining the

number of video sources in the network and reading the number of videos available, from here we determine how many times we can loop through the video file list with the number of sources present. Each video has an identifier associated with it, if there are more sources in the network than videos we will assign the videos sequentially, looping through the list of videos until we are unable to complete the full list. Once this occurs we divide the video list into steps greater than 1 using the remainder value from the division calculation. This allows the remaining sources to be evenly distributed across the entire list of videos. This situation also occurs when the number of video sources is less than the size of the video list. Once a video source finishes streaming a video it moves on to the next video in the sequence. Evenly distributing the videos over the entire list allows all videos to be streamed even with small networks, given a sufficiently long simulation.

```
READ Network_Size

READ Number_of_Videos

[LoopsNumber, Remainder] = Network_Size / Number_of_Videos

FOR i < LoopsNumber
    FOR j < Number_of_Videos
        ASSIGN video[j] to node(i*j)
    ENDFOR
ENDFOR

FOR k <= remaining unassigned nodes
    multiplier = k * Remainder
    ASSIGN video[multiplier] to node(k)
ENDFOR
```

Figure 4.2: Pseudocode for Video Assignment

In testing, experiments are conducted on networks that consisted of a combined AP and proxy station and a chosen number of video sources. At network initialization, each video source will randomly determine a time to begin sending video to the proxy station. The determined time to begin sending must be before the first second of network time has passed. The initial sending bitrate is equal to the physical rate of the proxy station divided by the number of video sources, irrespective of the physical rate of the

video source. Every second each video source sends a status update to the proxy station containing weight, physical rate, dropping rate, and local accuracy error information for that specific source. With the data from all of the sources, the AP is able to execute the effective airtime estimation algorithm and calculate the optimization solution terms.

```
WHILE RECEIVE Beacon_Pkt

    READ Beacon_Pkt.Fraction_of_Airtime

    CALCULATE Bitrate USING Beacon_Pkt.Fraction_of_Airtime

    IF Bitrate > PreviousBitrate + 100 OR Bitrate < PreviousBitrate – 100 THEN
        UPDATE Encoding Bitrate
    ENDIF

    PreviousBitrate = Bitrate;

ENDWHILE
```

Figure 4.3: Pseudocode for Encoding Bitrate Hysteresis

Choosing H.264 as our encoding standard allowed us to use a full video dataset as the content of our in-network video streams. As many face recognition datasets are comprised of individual images, it creates an added step in compiling these images into a video feed before encoding them with the proper codec. Using the Honda/UCSD dataset, we were able to eliminate this step in the process while also having the added benefit of having continuous video feeds in our network, as would exist in a real-world scenario. Each video source takes an input value for a frame rate and a desired bitrate. We set the frame rate to a constant 20 frames per second and the bitrate is adjustable based off the network conditions. We implement a simple hysteresis to the bitrate input to ensure that the video sources are not caught in a loop of changing the stream bitrate, this algorithm is shown in Figure 4.3. We view that in most cases relatively small changes to bitrate ($\leq$ 100 bps) do not affect the overall facial recognition accuracy.

4.4 Weighted vs. Non-Weighted Configurations

An additional approach was added to the testing to show the possible benefits of weighting. This was not explored in previous work, as it was assumed that weighting would always be performed. The weight values, or importance factors, are used to designate priority of video sources in the network. These values can be pre-assigned or dynamically assigned as the network operates. For the testing purposes in this thesis and since all videos streaming in the network contain similar content, there was no metric to base the weights on. Therefore, for this thesis the weighted optimization solution assigns randomly generated weights to each video source in the network before running. This approach is consistent across all network configurations and experiments. We test two variants of the ACBO solution in this thesis: unweighted and weighted. We refer to the unweighted variant as the *Accuracy Optimization without Weighting* (AO) method. For this solution, the weight values of all sources are equal. To determine the weight value, we divide 1 by the number of video sources in the network. This effectively ensures that there is no weighting as all video sources have the same importance when calculating the share of effective airtime. In the weighted variant, referred to as the *Weighted Accuracy Optimization* (WAO), five different levels of weight values exist; the weight of each video source is randomly chosen from this list. In both solutions the weight values range from 0 to 1, with the sum of all weights used being equal to 1. We assign the physical rates for each video source by creating an even distribution of the 6 possible physical rates of the 802.11g standard (12, 18, 24, 36, 48, and 54 Mb/s) [52] and randomly assigning the rates to each video source. Table 4.2**Error! Reference source not found.** summarizes the main simulation parameters.

4.5 Performance Metrics

We compare the two variants of the ACBO solution, AO and WAO, with the following solutions. (*1*) The enhanced distributed channel access (EDCA) solution, which provides no accuracy-based optimization, only giving benefits through reduction in network contention, and (*2*) Adaptive EDCA [15], a hypothetical method that utilizes the same framework as EDCA, but also takes into account the physical rate of each video source and the number of sources in the network when allocating bandwidth. In order to

Table 4.2: Simulation Parameters

Parameter	Values(s)
Number of Video Sources	2-76
Simulation Time	5-10 min
Packet Size	1024 bytes
Application Rate	Optimized by Bandwidth Allocation Solution, Default = Max Physical Rate / No. of Sources
Video Frame Rate	20 frames/sec
Physical Layer Characteristics	Extended Rate (802.11g)
Physical Layer Data Rate	Evenly Distributed, Randomly Assigned from Set: {12Mb/s, 18Mb/s, 24Mb/s, 36Mb/s, 48Mb/s, 54Mb/s}
Weights	Randomly Assigned from One of Fives Levels Between 0 and 1
Buffer Size	256 Kb
Video TXOP Limit	Optimized, Default = 3008 µs
Video CW_{min}	15
Video CW_{max}	31
Video AIFS	2
Short Retry Limit	7
Long Retry Limit	4
Beacon Interval	0.02 seconds
State Report Interval	1 second

provide useful comparisons to prior studies we select several main performance metrics. The primary performance metrics we analyze are facial recognition accuracy, overall network load, and power consumption. Facial recognition accuracy was an obvious choice as the ACBO solution intends to optimize bandwidth to improve or maintain accuracy. Network load and power consumption are important metrics to monitor, as they have implications on the cost and scalability of an AVS system. The facial recognition accuracy for AO, EDCA and Adaptive-EDCA is calculated as the average accuracy across all video sources in a network. For WAO the accuracy is determined as the sum of the weight-adjusted accuracy for each video source, shown in Equation (4), where S is the network size, w is the weight of a source, and A is the accuracy of the source. In each solution, the accuracy for each source is determined as the facial recognition accuracy for all received frames. The system assigns a value of zero for the accuracy of dropped frames. The overall network load is defined as the total load sent by the application layers of all video sources. Finally, power consumption is the average power consumption of the wireless interfaces of the video sources and is determined using the power consumption model in [53].

$$Total\ Facial\ Recognition\ Accuracy = \sum_{i=1}^{S} w_i \times A_i, \quad (4)$$

Chapter 5 Result Presentation and Analysis

This chapter will present and analyze the results of the tests performed on the OPNET simulated AVS networks focusing primarily on the performance metrics discussed in the previous chapter. It will discuss the efficiency of the effective airtime estimation algorithm when used in conjunction with H.264 encoded data packets. Furthermore, the chapter presents the results for our main performance metrics for testing of the ACBO solution applied to face recognition. The effectiveness of the enhancements to the ACBO solution (bandwidth pruning, bandwidth capping, and distributed face cropping) are analyzed thoroughly.

5.1 Analysis of Effective Airtime Estimation Method Applied to H.264 Encoding

A major part of the ACBO solution is the improvement to the effective airtime estimation through the use of a PID controller. Without a properly working PID controller we would not observe correct adjustments to the effective airtime, causing a loss in performance of the optimization. Utilizing the PID controller from [18], we perform experiments to determine the best values for the PID parameters, K_P, K_I, and K_D. Our findings show that as with the previous study, varying the value of A_{thresh} had the largest impact on the effective airtime. Table 5.1 shows the results of our experimentation.

Table 5.1: Summary of PID Parameters

Parameter	Value
A_{thresh}	0.01
K_P	6.25
K_I	5.25
K_D	0.75

These results deviate from those found in [18] and can be attributed to the differences in workloads between MJPEG and H.264 data. A main benefit to H.264 is being able to maintain the same video quality at a much lower bitrate than previously tested codecs. Consequently, when we utilize H.264 as our codec,

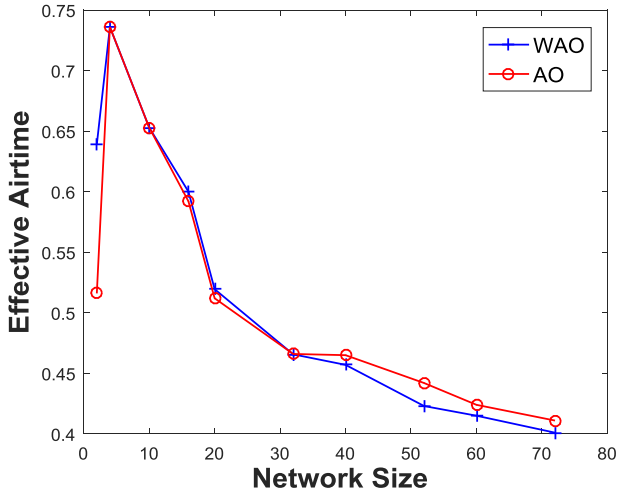

Figure 5.1: Average Effective Airtime for WAO and AO

the observed dropping rate remains low due to the lower necessary sending rate, thus allowing the use of more aggressive PID parameters. The previous implementation converges at around 70 seconds of simulated time, whereas with our parameters the convergence happens at around 40 seconds, with only minor adjustments due to network conditions after that. In this figure we use an A_{thresh} value of 0.005 in order to have a direct comparison, although for our main testing we do not use this value for A_{thresh}. Although this is not the value normally tested with, the convergence times are similar over multiple different values and as such are not shown in our results.

Figure 5.1 shows the average effective airtime versus the number of video sources in the network for the two bandwidth allocation variants. From the results, we see that as the number of video sources in

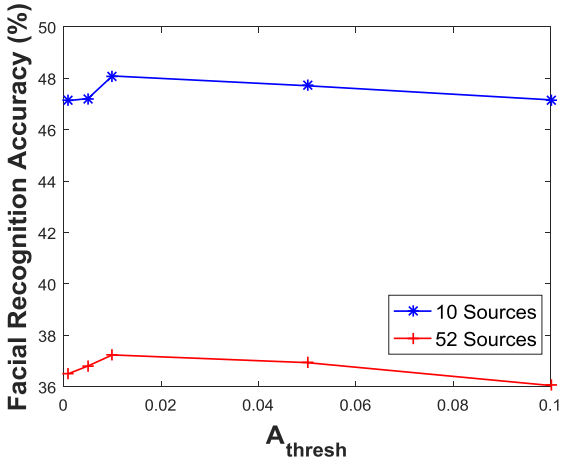

Figure 5.2: Facial Recognition Accuracy at Various Athresh Values

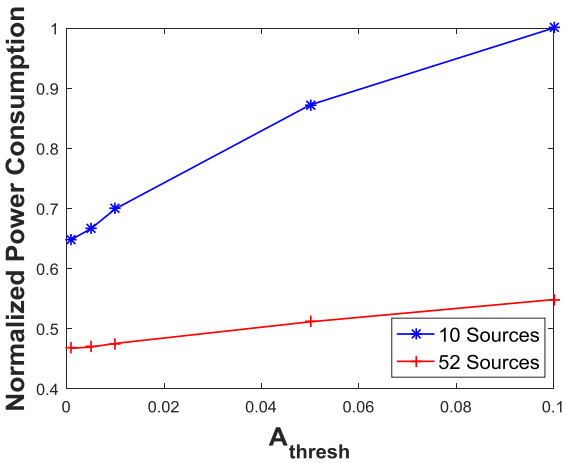

Figure 5.3: Power Consumption at Various A_{thresh} Values

the network increases so does the effective airtime up to a point, after which the effective airtime decreases gradually as source number increases. We observe that with H.264 encoded data packets the effective airtime peaks with a network size of four. This lines up with the hypothesis in [18], as the sending rate for the H.264 encoded video is much more aggressive than the sending rates for the tested encoding methods in that work. We observe that the average values for effective airtime are much larger in this study than in previous work. This has to do with the structure of the packets sent over the network. In previous studies, MJPEG data was sent over the network, with that encoding method all frames at any giving time are of similar sizes, making the traffic much more consistent. With H.264, the use of a mix of I, P, and B frames causes the size of the packets being sent over the network to be much more inconsistent at a specific time. This has a benefit though, with MJPEG this network is at its maximum stress at all times, however, with H.264 the stress on the network fluctuates but remains lower overall. While H.264 encoding would maintain the same overall bitrate, the different sizes of the three frame types makes the load on a network variable. There is rarely a time where only the largest frame type (I-frame) is sent over the network. This situation causes the largest stress on the network resulting in the highest observable dropping rate, but since this is a rare occurrence, dropping remains lower with H.264. This allows the bitrate to be increased in order to hit the level of allowable dropping defined by A_{thresh}, which is why a higher average effective airtime is observed with H.264 encoding. We also see in Figure 5.1 that the average effective airtime is very close

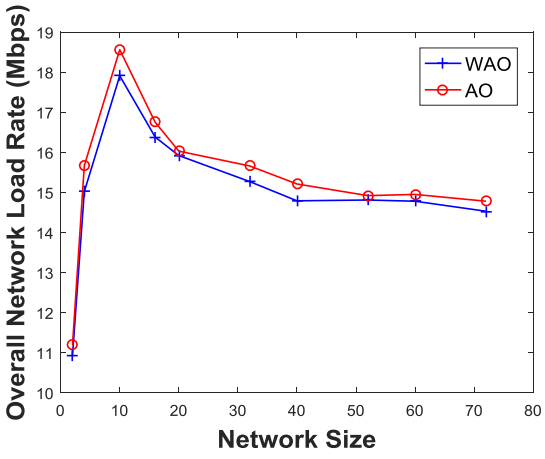

Figure 5.4: WAO vs. AO Comparison of Network Load

between the two ACBO variants; this is expected, as there are only small differences in overall network load between the two methods.

We will now look at the impact that A_{thresh} has on the optimization calculations. With the A_{thresh} value we control the amount of packet dropping that is allowable in the network, changing this value has a direct effect on the facial recognition accuracy and power consumption of the network. In Figure 5.2 we show the relationship between A_{thresh} and facial recognition accuracy. This figure shows that at very low A_{thresh} values there is a negative effect on accuracy. This can be attributed to the lower bitrate required to meet the dropping limit imposed by A_{thresh}. As we expect, as A_{thresh} increases we see an increase in accuracy to a point, after which we see accuracy decrease. This is due to the methodology chosen for calculating accuracy, in which a dropped packet is equal to an incorrect face recognition. With higher allowable dropping rates, the system is able to increase the bitrate for each video source; however, the effects of packet dropping on the accuracy offset any benefits that may have been provided. We see that in this system the accuracy peaks with an A_{thresh} value equal to 0.1, with smaller values there is a gradual drop-off in accuracy and with larger values there is a sharp decline in accuracy. Figure 5.3 shows that the power consumption increases with A_{thresh}. This is due to the higher achievable sending rate that occurs when more dropping is allowed. A_{thresh} selection depends on the application and should be chosen by analyzing the tradeoff between power consumption and desired accuracy.

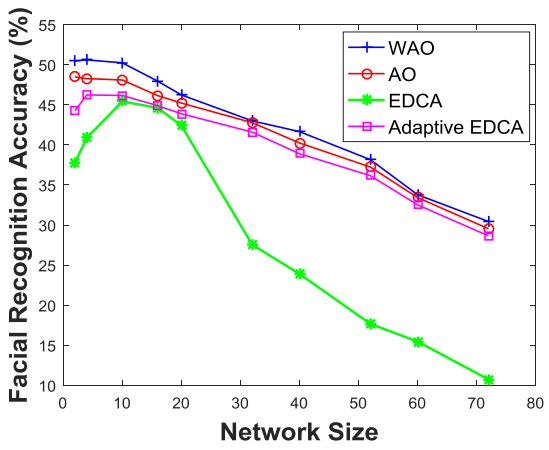

Figure 5.5: Comparison of Face Recognition Accuracy with Different Allocation Solutions

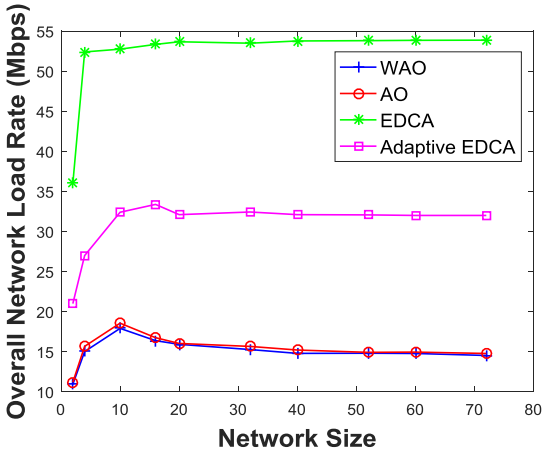

Figure 5.6: Comparison of Network Load with Different Allocation Solutions

5.2 Effectiveness of the Proposed Bandwidth Allocation Solution for Face Recognition

In Figure 5.5, Figure 5.6, and Figure 5.7 we show a comparison overall facial recognition accuracy, network load, and power consumption of several bandwidth allocation solutions using the Honda/UCSD dataset. The results show that with the AO solution we achieve significant increases in facial recognition accuracy over EDCA. Figure 5.5 shows that the accuracy ranges from a 5-65% increase when comparing AO to EDCA. With WAO, we see the range change to a 7-66% increase over EDCA. Comparing this to Adaptive EDCA, however, the increase in accuracy is not as significant. With AO, we see an increase in accuracy of 2-10% over Adaptive EDCA; with WAO, we see an increase of 3.5-13% over the Adaptive EDCA method. As we randomly assign the weighting in our testing, the implementation can be considered non-optimized. With optimization of the weights of the video sources based off individual conditions, we expect this range to further increase over EDCA.

We see in Figure 5.5 that with smaller network sizes AO and WAO both compare favorably to EDCA and Adaptive EDCA. The data shows that for the EDCA and Adaptive EDCA solutions there is an initial ramp up until a peak accuracy is reached. This peak occurs at a network size of 10 for EDCA and a network size of four for Adaptive EDCA. In smaller networks using EDCA, we attribute the lower accuracy to the requested sending rate reaching the maximum physical rate possible for those nodes, which results

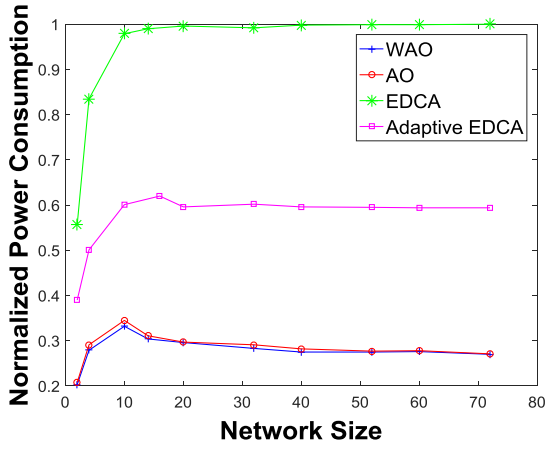

Figure 5.7: Comparison of Power Consumption with Different Allocation Solutions

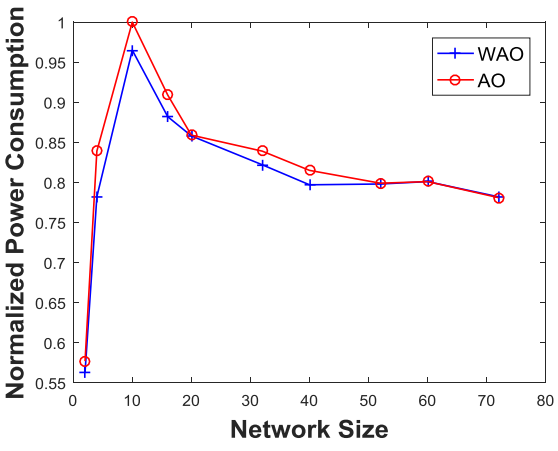

Figure 5.8: WAO vs. AO Comparison of Power Consumption

in a higher degree of packet loss. In our testing, we observe that although Adaptive EDCA is better overall at assigning bandwidth than EDCA, increased packet dropping still occurs with very small network sizes. This is interesting because by definition the assigned bandwidth should never exceed the physical rate of the connection when using the Adaptive EDCA solution. In our analysis, we observe that in these scenarios large amounts of contention in the network cause the elevated level of packet dropping. The sending time adjustments are not capable of overcoming the contention when few video sources exist in the network. This is also an issue with the standard EDCA implementation when used with smaller network sizes. As the WAO and AO methods are much better at distributing bandwidth, we do not observe the same accuracy trend for these methods, the smaller network sizes exhibit the peak accuracy and we see a downward trend as the network size increases. The initial positive slope seen with EDCA and Adaptive EDCA is not present.

With EDCA we observe a large drop-off in accuracy between networks sizes of 20 and 32 video sources. Observing the data for these simulations shows that there is a significant increase in the number of frames missed by the proxy station. As we can see in the network load data in Figure 5.6, the physical bandwidth limit of the medium is reached at 10 nodes with the EDCA solution, after which we observe increased contention in the network. Looking back at the accuracy data, we can see that the slope of the accuracy is much lower from 10 to 20 sources than from 32 to 72 sources. At 32 sources, we believe the network reaches a tipping point in the balance between data being sent over the network and data able to be processed by the proxy station. We do not observe this large drop-off with AO, WAO, or Adaptive EDCA. The data collected shows that none of these implementations reach the bandwidth limit of the medium, supporting our hypothesis.

In Figure 5.5, we can also see that the application of weights to each of the video sources provides the ability to tune the accuracy, allowing for further increases in recognition. In our testing, we randomly assign weights to each of the video sources. In these tests, we attain a 5% increase in facial recognition accuracy over the non-weighted solution; with more analysis of the activity from the video sources the weights could be adjusted to achieve even greater increases in accuracy.

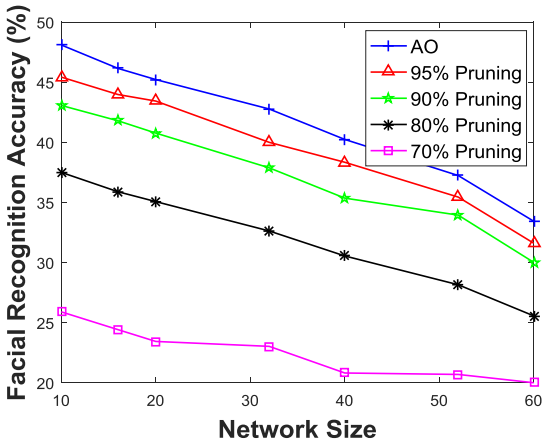

Figure 5.9: Comparison of Facial Recognition Accuracy at Various Pruning Levels

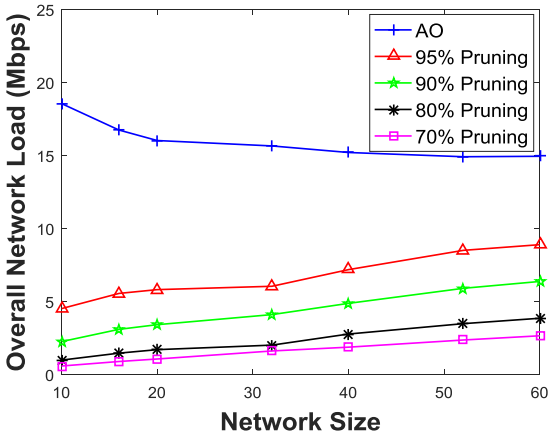

Figure 5.10: Comparison of Network Load at Various Pruning Levels

In addition to the facial recognition accuracy benefits from the AO and WAO bandwidth allocation variants, we also see a significant decrease in network load and power consumption in the tested networks, this data is shown in Figure 5.6 and Figure 5.7 respectively. The non-weighted optimization solution provides up to a 70% decrease in network load and power consumption from EDCA. Adding weights to the sources provides up to an additional 4% decrease in both metrics. Comparing AO to Adaptive EDCA, we see a reduction in network load and power consumption of up to 51%; with the weighted solution, we see up to an additional 2% reduction in both metrics. Figure 5.4 and Figure 5.8 show direct comparisons of the AO and WAO methods to more effectively display the differences in network load and power consumption between the two solutions. The figures show that the differences in network load and power consumption between the weighted and non-weighted variants are minimal in our implementation.

5.3 Analysis of the Bandwidth Pruning Mechanism

In this section, we discuss the effectiveness of the bandwidth pruning method when used with the AO bandwidth allocation variant. We analyze pruning at four different levels: 95%, 90%, 80%, and 70%. The pruning level specifies the expected percentage of the original accuracy achieved by the system when using the AO method. The decision to not test this mechanism with the WAO method was made because initial testing indicated that the results would closely match those of the AO method. The decreases in

sending rate in order to achieve the desired accuracy reductions matched closely between the two variants, as they did when testing the methods without pruning. The only differences seen in accuracy were related to the effects of weighting, which follows the same trend as the non-pruned tests. The observed network load and power consumption were nearly identical between the two variants when pruning was applied. The results of our pruning analysis with the AO method show that with the bandwidth pruning mechanism applied we are able to significantly reduce the network load and power consumption of a network with only relatively small decreases to accuracy.

Figure 5.9 shows the accuracy for all four levels of pruning plotted against the same results for the AO variant without pruning. From the figure, we see that the accuracy follows the expected trend and decreases as the number of video sources in the network increases. This is attributed to the per source sending rate decreasing as the source number increases which, as expected, causes a decrease in the face recognition accuracy. With the overall network load and power consumption, shown in **Figure 5.10** and **Figure 5.11** respectively, we see a different trend. For this data, we observe that as the number of video sources increases in the pruned networks the load and power consumption increases, unlike the non-pruned network where both of these parameters are inversely proportional to the number of sources in the network beyond a certain network size. What we observe in these scenarios is that the reduction in sending rate required to meet the desired accuracy is greater in the smaller networks. In small networks, individual

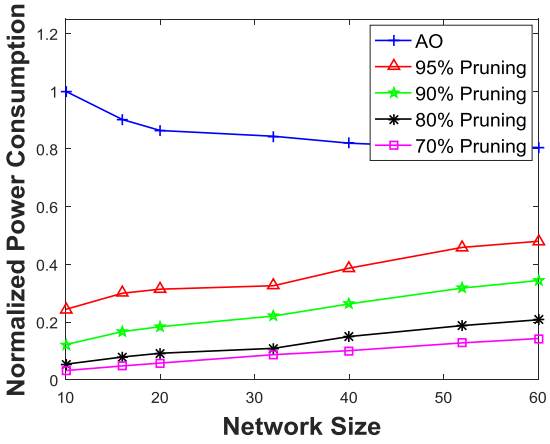

Figure 5.11: Comparison of Normalized Power Consumption at Various Pruning Levels

video sources start out at much higher sending rates than in larger networks. Due to the rate-accuracy curve leveling out at a relatively low bitrate, to achieve even a small reduction in accuracy the sending rate of the sources must be reduced significantly, thus providing a large benefit to smaller networks. This is not the case with the larger network sizes as the individual sending rate for each node is already at the low end of the rate-accuracy curve. In this situation the reduction in sending rate to achieve the desired accuracy is much less. This has the effect of normalizing the sending rates across differing network sizes and we consider this ideal behavior. The original sending rates for the video sources in smaller networks are significantly higher than they need to be to maximize the face recognition accuracy, creating unnecessary load on the network. With larger network sizes, we see the benefits decrease as the original sending rates for these video sources are already close to the transition point in the rate-accuracy curve where the accuracy drops off quickly. We can deduce that at a network size larger than those tested in this study the original sending rates will be such that the pruning provides little to no benefit to the network. However, this network size would be sufficiently large that computational limits of the proxy station would have become the main limiting factor. In addition, networks of this size with the configuration we examine in this thesis would be uncommon in real world scenarios due to cost of implementation and limits in the computational and network mediums. From the data shown in **Figure 5.11**, we see that the power consumption follows the same trend as the network load. This is expected, as power consumption is directly proportional to sending rate as previously mentioned.

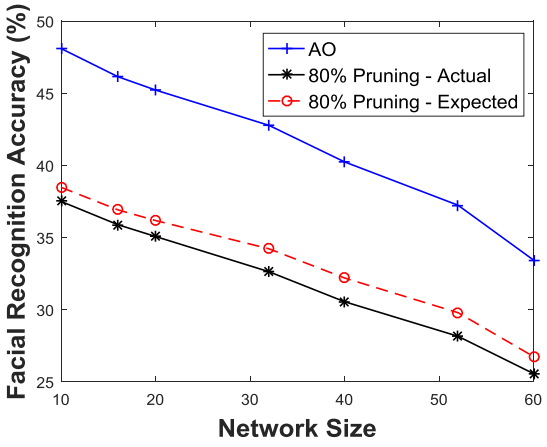

Figure 5.12: Expected vs. Actual Facial Recognition Accuracy at 80% Pruning

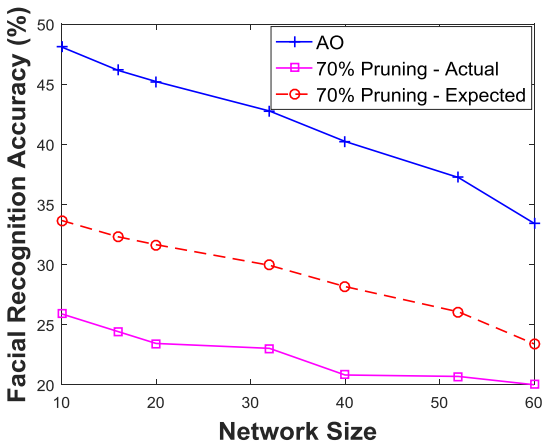

Figure 5.13: Expected vs. Actual Facial Recognition Accuracy at 70% Pruning

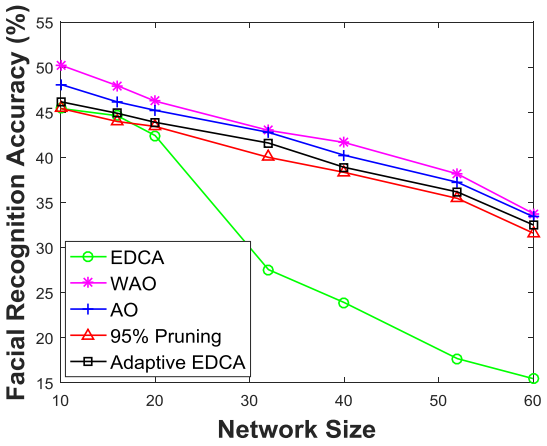

Figure 5.14: Facial Recognition Accuracy with 95% Pruning Compared to Other Solutions

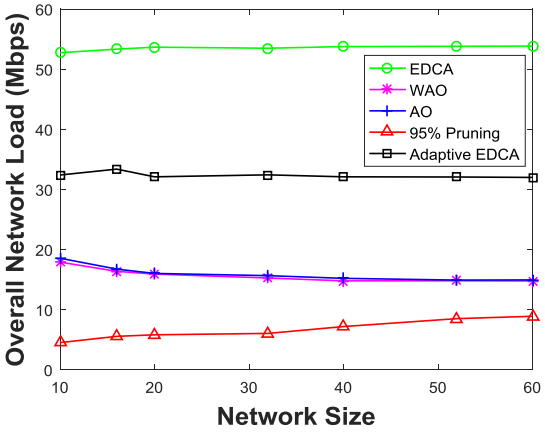

Figure 5.15: Network Load with 95% Pruning Compared to Other Solutions

From the data, we see that with 95% and 90% pruning the achieved accuracy matches the desired accuracy closely, once we drop to 80% and 70% pruning the achieved accuracy begins to vary largely from the desired accuracy, especially with the smaller network sizes. Figure 5.12 and Figure 5.13 show the expected accuracy after pruning against the actual value obtained for both 80% and 70% pruning respectively; the AO curve is included for reference. There are several causes for this type of behavior; the leading cause can be attributed to the tolerances of FFmpeg encoding when assigning bitrates. For the bitrate assignment with FFmpeg we observe that there is some variance between the achieved bitrate and the assigned bitrate. This is expected, since the achieved bitrate is highly dependent on the video source being encoded. We observe that in general with our selected video set the achieved bitrate tends to be lower

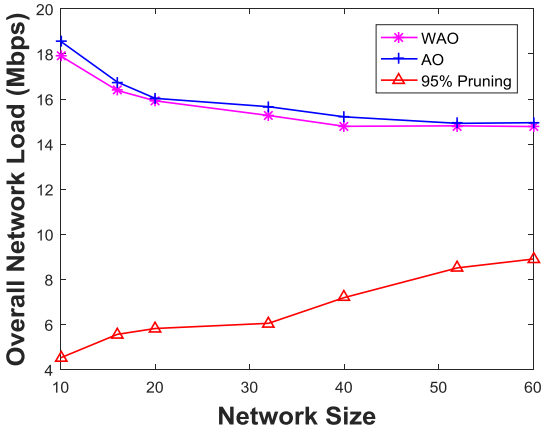

Figure 5.16: Comparison of Network Load with 95% Pruning vs. WAO vs. AO

than the specified bitrate. Further exacerbating the issue is that toward the lower end of our accuracy model small variances in bitrate result in large variances in face recognition accuracy. This can be seen in the zoomed view of the rate-accuracy model in Figure 3.4. When trying to prune a larger percentage of the original accuracy the desired bitrates begin to fall into this region of our rate-accuracy model and with FFmpeg unable to achieve an exact bitrate, it can lead to large differences between the achieved and desired accuracy. We see a larger discrepancy between the expected and achieved accuracy in smaller networks because they start out at a much higher initial sending rate, meaning there needs to be a large decrease in sending rate in order to achieve the desired pruning according to the rate-accuracy curve. In addition, with less video sources in the network, even a single source missing its sending rate target will have a large effect on the average accuracy of the network. In larger networks, this error is masked due to the number of video sources streaming as well as their lower initial starting rate.

Figure 5.14 shows the comparison of the average accuracy for AO with 95% pruning results with the results of the other bandwidth allocation methods tested. From the figure, we see that the accuracy of AO with 95% pruning follows the same trend as WAO, AO, and Adaptive EDCA. In addition, it shows that the accuracy achieved by the 95% pruning method is about 2% lower than the Adaptive EDCA method at all network sizes. With smaller numbers of video sources, the pruning method is also lower than EDCA; however, with larger network sizes we still see a significant increase in accuracy over EDCA. While these

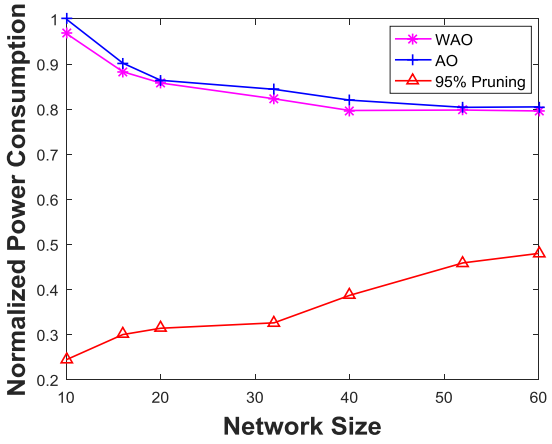

Figure 5.17: Comparison of Power Consumption with 95% Pruning vs. WAO vs. AO

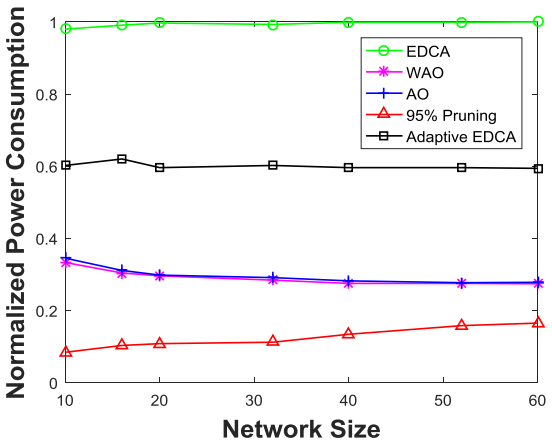

Figure 5.18: Comparison of Power Consumption with 95% Pruning vs. Other Solutions

accuracy data might not give optimal results in terms of face recognition accuracy, we see in the data shown in Figure 5.15 and Figure 5.18 that a substantial decrease in overall network load and power consumption is achieved with pruning. Figure 5.16 and Figure 5.17 show a direct comparison of the 95% pruning with AO and WAO to better show the behavior compared to the solutions under test. With the pruning method, we see up to a 95% reduction in network load and power consumption compared to EDCA. In addition, we see up to an 86% decrease in network load and power consumption when compared to Adaptive EDCA. As discussed previously, Adaptive EDCA performs exceptionally well in our scenarios while requiring little overhead, however, the network load and power consumption are still significantly higher than the ACBO-based methods. While there is a slight trade off in accuracy, the pruning method performs exceptionally well in regards to network load and power consumption. This method should be of strong consideration in situations where power consumption is an important factor.

5.4 Effectiveness of Proposed Bandwidth Capping Method

With the proposed bandwidth capping method, we expect to see a significant decrease in network load and power consumption in the simulated networks. It is also predicted that the effects of bandwidth capping would diminish as the network size grew, since it only makes sense to cap bandwidth to a value that does not adversely affect the performance of the face recognition accuracy. As network size grows the

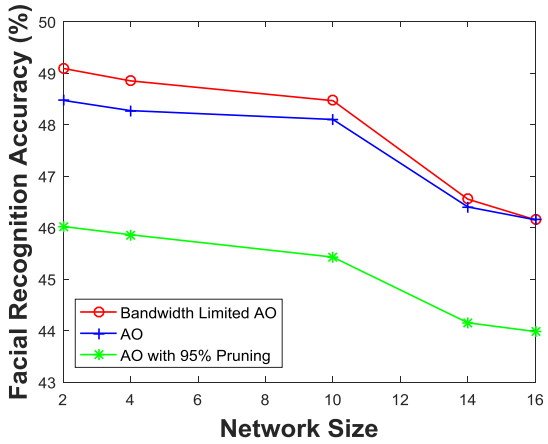

Figure 5.19: Bandwidth Capped vs. Non-Capped vs. Pruned Comparison of Facial Recognition Accuracy

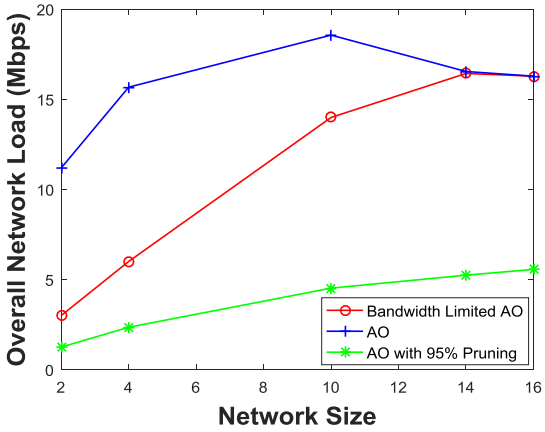

Figure 5.20: Bandwidth Capped vs. Non-Capped vs. Pruned Comparison of Network Load

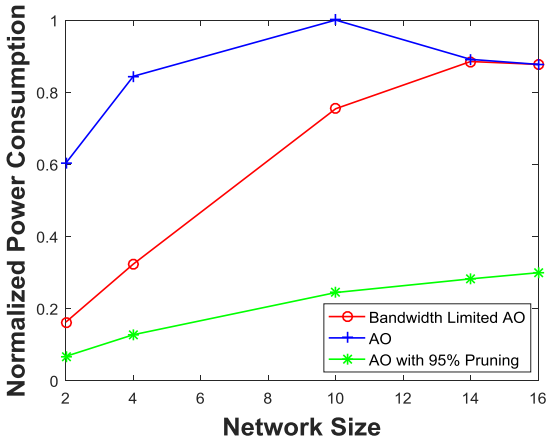

Figure 5.21: Bandwidth Capped vs. Non-Capped vs. Pruned Comparison of Power Consumption

average bitrate for each video source decreases gradually to values below the bandwidth cap, thus disregarding the cap. For the testing in this thesis a cap of 1.5 Mbps is chosen based off the rate-accuracy curve generated for the face recognition database under test. We see in Figure 5.20 and Figure 5.21 that the results follow the expected behavior for both network load and power consumption. For configurations of 2 and 4 video sources, we observe a reduction in network load and power consumption of 73% and 62% respectively. For the 10 video source network configuration, we observe a smaller, but still significant, decrease of 25%. Beyond 14 sources, the average bitrate of the video sources decreases below the bandwidth cap and the benefit is no longer present. As expected the bandwidth capping did not negatively affect the face recognition accuracy; in fact, as Figure 5.19 shows, there is a small increase in accuracy over all configurations tested in which the bandwidth capping was imposed. We observe a 1% increase in face recognition accuracy for the 2, 4, and 10 video source configurations. This result is similar in magnitude to the increase in accuracy observed when implementing a weighting system in the network. While this proposed enhancement only works with smaller network sizes, this method is much simpler to implement and requires much less input from outside sources than a weighting system.

Compared to the pruning method, the bandwidth capping method provides several benefits. In networks with up to 14 video sources we see an increase in accuracy over both the AO-only and AO with pruning variants even though we are still able to reduce the network load and power consumption. Although

the reductions in these two metrics are not as great as with the bandwidth pruning method, they are still significant. The bandwidth capping method is also less computationally intensive than the bandwidth pruning method. With the pruning method an initial bitrate is determined, then calculations are performed to determine the expected accuracy at that initial bitrate and the bitrate necessary to result in a specified percentage of that accuracy. The bandwidth capping method only requires a predetermined cap to be used during the initial bitrate calculation. The main disadvantage of the bandwidth capping method compared to the bandwidth pruning method is that once networks reach a certain size the capping no longer takes place, whereas the pruning method is able to work for all ranges of network sizes.

5.5 Effectiveness of Proposed Distributed Face Cropping Method

We evaluate the proposed distributed face cropping method using the same performance metrics as previous tests in this thesis. We monitor the facial recognition accuracy, overall network load, and power consumption. The results for accuracy are shown in Figure 5.22, with the results for network load and power consumption being shown in Figure 5.23 and Figure 5.24 respectively. What we observe is that there is a substantial increase in facial recognition accuracy compared to the AO bandwidth allocation variant. We observe that with smaller network sizes the facial recognition accuracy increases by about 5.5%, this attained accuracy comes within 1% of the maximum possible accuracy of 52% that we observe with the Honda/UCSD database. While this is a significant improvement over the non-enhanced solution, the

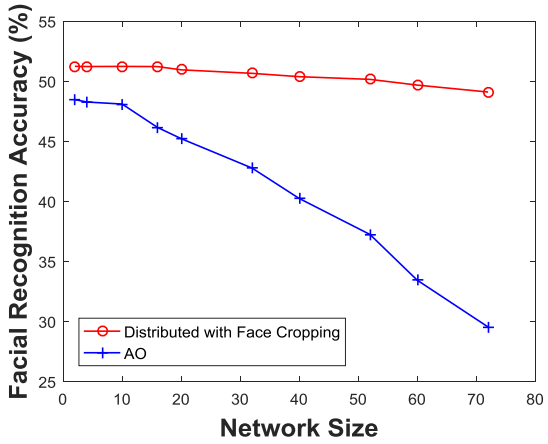

Figure 5.22: Distributed Face Cropping vs. AO Comparison of Facial Recognition Accuracy

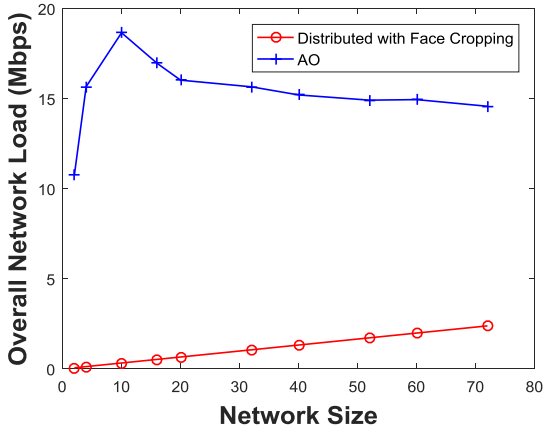

Figure 5.23: Distributed Face Cropping vs. AO Comparison of Network Load

enhancement really begins to show its effectiveness as the network size increases. As the number of video sources in the network increases, the accuracy value stays close to the observed maximum for much longer than without this enhancement. We see that with the AO solution, the accuracy begins to drop rapidly in networks with greater than 10 video sources. Over the range of network sizes tested in this thesis the accuracy drops 40% with only AO implemented. With the distributed face cropping method, the decrease in accuracy over that same range is only 4.2%. There are several reasons that can be attributed to the observed increase in accuracy. The main reason being the ability to use high quality video for all network sizes tested. In the tests run, the system was able to send face images at full quality, as even with that level of quality the reduction in data compared to sending the full frames results in the dropped packet rate staying close to zero. Figure 5.23 backs up this claim, as we see the network load at each network size is orders of magnitude lower than with just AO implemented. As the system counts a dropped packet as zero for the instantaneous accuracy, having effectively no packet dropping in the network helps significantly by reducing the likelihood of such an occurrence.

The proposed distributed face cropping method also has a large impact on the network load. In Figure 5.23, it can be seen that with smaller network sizes there is a 99% reduction in load, with larger network sizes this decreases to an 83.5% reduction in network load. The network load trend for the distributed face cropping method does not follow the same trend as AO by itself. This is because all video sources in each network configuration are effectively streaming at the same rate after cropping the faces,

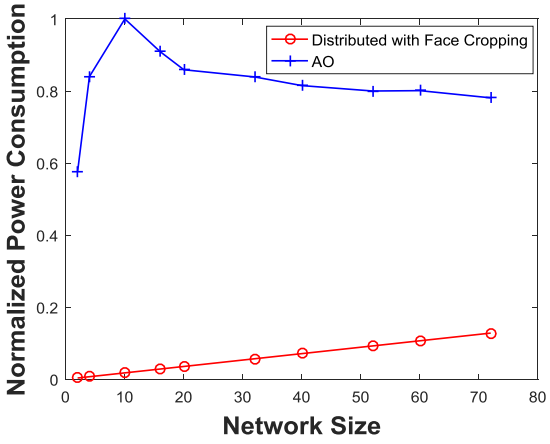

Figure 5.24: Distributed Face Cropping vs. AO Comparison of Normalized Transmission Power Consumption

due to there being no need to reduce the image quality. To stream at full quality with cropped videos the necessary bandwidth is only 23Kbps. As video sources are added the network load increases linearly. We expect that if video sources were to continue to increase the network would reach a saturation point and a reduction in the video quality would needed. However, network sizes large enough saturate the medium would have hit computational processing limits well before reaching that point. With 72 video sources we already observe limits due to the computation power of the proxy station, with the station missing packets due to the time needed to perform facial recognition for that number of video feeds.

Power consumption due to transmission sees similar gains as network load with the distributed face cropping method. Figure 5.24 shows that as with network load, with smaller network sizes we observe a 99% decrease in transmission power consumption and with larger network sizes the reduction is around 83.5%. There was no work done to re-characterize the power consumption model of the system, we used the same model from [53]. We assume that as the face detection is still being performed, although now at the video sources, the power consumption model would not change significantly. Therefore, we expect that the primary change in power usage is from the transmission of data from the video sources to the proxy station. With the large reduction in sending rate, the model shows that there should be a similarly large reduction in transmission power consumption. As the processor architectures of video sources may not have the same level of efficiency at performing CV tasks as the proxy station, more work needs to be done to characterize the power consumption of smart cameras when performing image manipulation tasks.

Chapter 6 Conclusions and Future Work

6.1 Conclusions

We have analyzed the accuracy-based cross-layer bandwidth optimization solution (ACBO) for automated video surveillance (AVS) systems when applied to face recognition. Through our testing we have shown that the ACBO solution can successfully manage bandwidth even when an efficient codec like H.264 is used. We have developed effective facial recognition and video streaming implementations for use with AVS systems addressing the shortcomings of previous work. We have proposed two effective enhancements to the ACBO solution: bandwidth capping and distributed face cropping. With bandwidth capping we have developed an enhancement that is able to reduce excessive bandwidth usage. The distributed face cropping enhancement provides improvements to network load through the distribution of computer vision (CV) tasks, allowing image manipulations, such as face detection, to be performed at video sources, resulting in a reduction of data sent over the network. The systems we have designed closely match real-world implementations and are able to be performed in real-time. Through our experimentation we have demonstrated the effectiveness of the framework under differing conditions. We have performed extensive work to implement a face recognition system into an OPNET simulated network, including the implementation of a full video transcoding system using FFmpeg and a full video training system for the recognition algorithm. Using OPNET-simulated wireless networks of varying sizes, we have extensively tested the application facial recognition to the ACBO solution. The main findings of our evaluations can be summarized as follows:

(1) Face recognition accuracy follows a similar trend to face detection, allowing the use of the same model for the rate-accuracy curve. Consequently, the ACBO solution requires only minor modifications to work with a face recognition system. We believe that this should hold true for most CV algorithms. As we have discussed, CV algorithms are tolerant to changes in video

quality and thus accuracy will remain at a consistent level until a critical point in quality is reached, after which the accuracy begins to decrease significantly.

(2) Face recognition accuracy, network load, and power consumption are significantly enhanced with ACBO-based solutions when compared to the enhanced distributed channel access (EDCA) and Adaptive-EDCA frameworks. Using the rate-accuracy model, the system is able to considerably reduce network load while also maximizing recognition accuracy. The reduction in power consumption and increase in recognition accuracy can be partially attributed to lower dropping rates in networks utilizing the ACBO framework.

(3) Even with a high compression codec like H.264, the ACBO solution is still able to provide significant benefits over EDCA and Adaptive-EDCA. No significant changes were needed in the calculation of the rate-accuracy curve when H.264 was used, as the data still follows the same rate-accuracy model function.

(4) The bandwidth pruning method is able to significantly reduce network load and power consumption across all tested network sizes. These results were viewed across several different pruning percentages. With the 95% pruning level the ACBO-based solutions achieve better face recognition accuracy than both the EDCA and Adaptive-EDCA solutions in the majority of situations while also reducing the network load and power consumption by up to 75%, demonstrating that bandwidth pruning is still effective in a system running facial recognition.

(5) The proposed bandwidth capping method results in significant reductions in network load and power consumption in small AVS networks. Interestingly, this enhancement also results in a slight increase in facial recognition accuracy. This 1% increase can be attributed to the reduction in packet loss exhibited when the network is not under full load. As this is a simple enhancement to implement and it does not negatively affect large networks, we recommend that it be used in all future implementations of the accuracy-based cross-layer bandwidth allocation solution.

(6) The proposed distributed face cropping method substantially reduces network load in the system under test while also resulting in an overall increase in facial recognition accuracy. Gains in facial recognition accuracy were up to 40% and the accuracy remained consistent as network size increased. Up to 99% reductions in network load were observed with this optimization. While this enhancement requires significant restructuring of the architecture of an AVS system, if the resources are available it provides dramatic gains to the performance of the system.

6.2 Future Work

For future work, we plan to test cross-layer optimization approaches over a variety of different network protocols. Due to limitations in our simulation software, we were only able to test using the 802.11g protocol. We plan to evaluate the ACBO solution in the newer 802.11n and 802.11ac protocols to see if we achieve the same benefits. As 802.11g is an older protocol, it would be beneficial to obtain results with a newer protocol. Our findings thus far indicate that the ACBO solution applied to face recognition would be beneficial with newer protocols as the largest difference between our tested protocol and newer technologies is the bandwidth available in the medium. We have already performed tests varying the physical rate of video sources, which have displayed the benefit of the allocation solutions. In addition, with more bandwidth available we plan to use simulations to test with larger network sizes to determine how well the solution scales. We also plan to test the effects of channel noise on the recognition accuracy-based optimizations, with our current model noise is left at the default setting in OPNET. With more work we will characterize different channel noise scenarios and further develop the solution so that it can react more effectively in these situations. It would also be beneficial to test the system on real hardware to understand if there are any intricacies present in a physical system. With H.265 in its nascence, it would be proactive to evaluate an accuracy model as well as testing the behavior of the ACBO solution when utilizing this protocol. As shown with the enhancements proposed in this thesis, there is still much work to be done that can result in further reductions in network usage and power consumption. Additional work is planned

to be done to characterize the power consumption when the face detection task is distributed among the network. With future work we intend to explore the proposed enhancements further.

REFERENCES

[1] MarketsandMarkets, "Video Surveillance Market by System (Analog, IP, Biometrics), Hardware (Camera, Monitors, Servers, Storage Devices), Software (Video Analytics, VMS), and Service (VSaaS, Installation & Maintenance), Vertical, and Region - Global Forecast to 2022," marketsandmarkets.com, Pune, 2017.

[2] T. A. Scally, "State of the Market: Video Surveillance 2017," BNP Media, 3 February 2017. [Online]. Available: http://www.sdmmag.com/articles/93511-state-of-the-market-video-surveillance. [Accessed 20 June 2017].

[3] L. Zhang, S. Z. Li, X. Yuan and S. Xiang, "Real-time object classification in video surveillance based on appearance learning," in *Proceedings of the 2007 IEEE Converence on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, MN, 2007, pp. 1-8.

[4] A. Oza, L. Mihaylova, D. Bull and N. Canagarajah, "Structural similarity-based object tracking in multimodality surveillance videos," *Machine Vision and Applications*, vol. 20, pp. 71-83, January 2009.

[5] Y. Jia and C. Zhang, "Front-view vehicle detection by markov chain monte carlo method," *Pattern Recognition*, vol. 42, pp. 313-321, 2009.

[6] L. Bourdev, S. Maji and J. Malik, "Describing people: poselet-based attribute classification," in *Proceedings of the 2011 International Conference on Computer Vision (ICCV)*, Barcelona, Spain, November 2011. pp. 1543-1550.

[7] R. Collins *et al.*, "A system for video surveillance and monitoring," Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-00-12, May 2000.

[8] O. Javed and M. Shah, "Tracking and object classification for automated surveillance," in *Proceedings of the 7th European Conference on Computer Vision-Part IV*, Copenhagen, Denmark, May 2002. doi: 10.1007/3-540-47979-1_23.

[9] R. Singh, S. Vishwakarma, A. Agrawal and M. D. Tiwari, "Unusual activity detection for video surveillance," in *Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia (ITTM)*, New York, NY, USA, 2010, pp. 297-305.

[10] J. C. Niebles, C.-W. Chen and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *Proceedings of the 12th European Conference of Computer Vision: Part II (ECCV)*, Crete, Greece, 2010. pp. 392-405.

[11] L. Fei-Fei and L.-J. Li, "What, where and who? telling the story of an image by activity classification, scene recognition and object categorization," *Studies in Computational Intelligence- Computer Vision*, pp. 157-171, 2010.

[12] V. Nair and J. Clark, "Automated visual surveillance using hidden markov models," in *Proceedings of the 15th International Conference on Vision Interface*, Calgary, Canada, 2002. pp. 88-93.

[13] W. Niu and J. Long, "Human activity detection and recognition for video surveillance," in *Proceedings of the IEEE Multimedia and Expo Conference (ICME)*, Taipei, Taiwan, 2004. pp. 719-722.

[14] Y. Dedeoğlu, B. U. Töreyin, U. Güdükbay and A. E. Çetin, "Silhouette-based method for object classification and human action recognition in video," in *Proceedings of the 2006 International Conference on Computer Vision in Human-Computer Interaction*, Berlin, Germany, 2006. pp. 777-780.

[15] M. A. Alsmirat and N. J. Sarhan, "Cross-layer optimization and effective airtime estimation for wireless video streaming," in *Proceedings of the 21st International Conference on Computer Communications and Networks (ICCCN)*, Munich, Germany, 2012. pp. 1-7.

[16] C.H. Hsu and M. Hefeeda, "A framework for cross-layer optimization of video streaming in wireless networks," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, February 2011. pp. 5:1-5:28.

[17] J. Huang, Z. Li, M. Chiang and A. K. Katsaggelos, "Pricing-based rate control and joint packet scheduling for multi-user wireless uplink video streaming," in *Proceedings of the 15th International Packet Video Workshop (PV2006)*, 2006.

[18] M. Alsmirat and N. J. Sarhan, "Cross-layer optimization for automated video surveillance," in *2016 IEEE International Symposium on Multimedia (ISM)*, San Jose, CA, 2016. pp. 243-246.

[19] P. Korshunov and W. T. Ooi, "Video quality for face detection, recognition, and tracking," *ACM Trans. Multimedia Computer Communications Appl.*, vol. 7, pp. 14:1-14:21, September, 2011.

[20] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*, 2nd ed., Springer Publishing Company, Incorporated, 2011.

[21] K. Sutherland, D. Renshaw and P. B. Denyer, "Automatic face recognition," in *Proceedings of the First International Conference on Intelligent Systems Engineering*, Edinburgh, Scotland, 1992. pp. 29-34.

[22] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, pp. 71-86, January 1991.

[23] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings of the 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1991. pp. 586-591.

[24] M. Kirby and L. Sirovich, "Application of the karhunen-loeve procedure for the characterization of human faces," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 103-108, January 1990.

[25] P. Korshunov, "Rate-accuracy tradeoff in automated, distributed video surveillance systems," in *Proceedings of the 14th annual ACM international conference on Multimedia*, Santa Barbara, CA 2006. pp. 887-889.

[26] P. Korshunov and W. T. Ooi, "Critical video quality for distributed automated video surveillance," in *Proceedings of the 13th annual ACM International Conference on Multimedia*, Hilton, Singapore, 2005. pp. 151-160.

[27] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[28] FBI, *Face Recognition - FBI at https://www.fbi.gov/file-repository/about-us-cjis-fingerprints\textunderscorebiometrics-biometric-center-of-excellences-face-recognition.pdf.*

[29] K.-C. Lee, J. Ho, M.-H. Yang and D. Kriegman, "Video-based face recognition using probabilistic appearance manifolds," in *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003. pp. I-313-I-320 vol.1.

[30] K.-C. Lee and D. Kriegman, "Online learning of probabilistic appearance manifolds for video-based recognition and tracking," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, 2005. pp. 852-589.

[31] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, pp. 711-720, July 1997.

[32] K. Etemad and R. Chellappa, "Face recognition using discriminant eigenvectors," in *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference*, Atlanta, GA, 1996. pp. 2148-2151

[33] T. Ahonen, A. Hadid and M. Pietikainen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 2037-2041, Dec 2006.

[34] Y. Zhang, T. Chai and C.-C. Hung, "Local binary patterns for face recognition under varying variations," in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW)*, New York, NY, USA, 2010.

[35] H. R. Farhan, M. H. Al-Muifraje and T. R. Saeed, "Using only two states of discrete HMM for high-speed face recognition," in *2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA)*, 2016. pp. 39:1-39:4.

[36] I. Sajid, S. G. Ziavras and M. M. Ahmed, "Hardware-based speed up of face recognition towards real-time performance," in *Proceedings of the 2010 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools*, Lille, France, 2010. pp. 763-770.

[37] M. J. Er, W. Chen and S. Wu, "High-speed face recognition based on discrete cosine transform and RBF neural networks," *IEEE Transactions on Neural Networks*, vol. 16, pp. 679-691, May 2005.

[38] G. Aggarwal, A. K. R. Chowdhury and R. Chellappa, "A system identification approach for video-based face recognition," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Cambridge, UK, 2004. pp. 175-178 Vol. 4.

[39] S. Kokila and B. Yogameena, "Face recognition based person specific identification for video surveillance applications," in *Proceedings of the Third International Symposium on Women in Computing and Informatics (WCI '15)*, New York, NY, 2015. pp. 143-148.

[40] R. Min and J.-L. Dugelay, "Cap detection for moving people in entrance surveillance," in *Proceedings of the 19th ACM International Conference on Multimedia (MM '11)*, New York, NY, USA, 2011. pp. 1253-1256.

[41] S. Banerjee, S. Samanta and S. Das, "Face recognition in surveillance conditions with bag-of-words, using unsupervised domain adaptation," in *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing (ICVGIP '14)*, New York, NY, USA, 2014. pp. 50:1-50.8.

[42] D. O. Gorodnichy, D. Bissessar, E. Granger and R. Laganiére, "Recognizing people and their activities in surveillance video: technology state of readiness and roadmap," in *Proceedings of 2016 13th Conference on Computer and Robot Vision (CRV)*, Victoria, BC, Canada, 2016. pp. 250-259.

[43] Y. M. Mustafah, A. W. Azman, A. Bigdeli and B. C. Lovell, "An automated face recognition system for intelligence surveillance: smart camera recognizing faces in the crowd," in *Proceedings of 2007 First ACM/IEEE International Conference on Distributed Smart Cameras*, Vienna, Austria, 2007. pp. 147-152.

[44] Y. Andreopoulos, N. Mastronarde and M. van der Schaar, "Cross-layer optimized video streaming over wireless multihop mesh networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 2104-2115, 2006.

[45] H. Zhang, Y. Zheng, M. A. Khojastepour and S. Rangarajan, "Cross-layer optimization for streaming scalable video over fading wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 344-353, 2010.

[46] S. Khan, J. Brehmer, W. Kellerer, W. Utschick and E. Steinbach, "Application-driven cross-layer optimization for video streaming over wireless networks," *IEEE Communications Magazine*, vol. 44, pp. 122-130, 2006.

[47] Z. He and D. Wu, "Resource allocation and performance analysis of wireless video sensors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, pp. 590-599, 2006.

[48] S. Davani, N. Sarhan, "Experimental Analysis of Bandwidth Allocation in Automated Video Surveillance Systems," *ACM Multimedia (To Appear)*, October 2017.

[49] H. R. Hamandi and N. J. Sarhan, "Rate-Accuracy Characterization for face recognition with the Honda/UCSD database," Wayne State Univ., Detroit, MI, Tech. Rep., 2016.

[50] P. Viola and M. Jones, "Robust real-time face detection," in *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137-154, May 2004.

[51] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features,", in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001).*, Kauai, HI, 2001. pp. I-511-I-518 vol. 1.

[52] "IEEE Standard for Information technology-Telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," pp. C1-1184, 12 2007.

[53] Y. Sharrab, "Video Stream Adaptation In Computer Vision Systems," Ph.D. dissertation, Dept. Elec.and Comp. Eng., Wayne State Univ., Detroit, MI, 2017

ABSTRACT

ANALYSIS OF CROSS-LAYER OPTIMIZATION OF FACIAL RECOGNITION IN AUTOMATED VIDEO SURVEILLANCE

by

LOREN GARAVAGLIA

August 2017

Advisor: Dr. Nabil Sarhan

Major: Computer Engineering

Degree: Master of Science

Interest in automated video surveillance systems has grown dramatically and with that so too has research on the topic. Recent approaches have begun addressing the issues of scalability and cost. One method aimed to utilize cross-layer information for adjusting bandwidth allocated to each video source. Work on this topic focused on using distortion and accuracy for face detection as an adjustment metric, utilizing older, less efficient codecs. The framework was shown to increase accuracy in face detection by interpreting dynamic network conditions in order to manage application rates and transmission opportunities for video sources with the added benefit of reducing overall network load and power consumption.

In this thesis, we analyze the effectiveness of an accuracy-based cross-layer bandwidth allocation solution when used in conjunction with facial recognition tasks. In addition, we consider the effectiveness of the optimization when combined with H.264. We perform analysis of the Honda/UCSD face database to characterize the relationship between facial recognition accuracy and bitrate. Utilizing OPNET, we develop a realistic automated video surveillance system that includes a full video streaming and facial recognition implementation. We conduct extensive experimentation that examines the effectiveness of the framework to maximize facial recognition accuracy while utilizing the H.264 video codec. In addition, network load

and power consumption characteristics are examined to observe what benefits may exist when using a codec that maintains video quality at lower bitrates more effectively than previously tested codecs. We propose two enhancements to the accuracy-based cross-layer bandwidth optimization solution. In the first enhancement we evaluate the effectiveness of placing a cap on bandwidth to reduce excessive bandwidth usage. The second enhancement explores the effectiveness of distributing computer vision tasks to smart cameras in order to reduce network load.

The results show that cross-layer optimization of facial recognition is effective in reducing load and power consumption in automated video surveillance networks. Furthermore, the analysis shows that the solution is effective when using H.264. Additionally, the proposed enhancements demonstrate further reductions to network load and power consumption while also maintaining facial recognition accuracy across larger network sizes.

AUTOBIOGRAPHICAL STATEMENT

Loren Garavaglia is an Embedded Powertrain Software Engineer at FCA US LLC. While working in the automotive industry he has been tasked with developing controls and software architecture for engine, transmission, and instrument cluster controllers. He has also performed extensive work in process and code generation automation. He received his B.S. degree in Electrical Engineering from Wayne State University in 2012. His main research interests are bandwidth adaptations for video streaming systems, computer vision, machine learning, and parallel and distributed system design.