Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Estado de México

School of Engineering and Sciences

Closing the Gap on Affordable Real-Time Very Low Resolution Face Recognition for Automated Video Surveillance

A thesis presented by

Luis Santiago Luévano García

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

Mexico City, December, 2022

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Estado de México

School of Engineering and Sciences

The committee members, hereby, certify that have read the thesis presented by Luis Santiago Luévano García and that it is fully adequate in scope and quality as a partial requirement for the degree of Doctor of Philosophy in Computer Science.

Dr. Miguel González Mendoza
Tecnológico de Monterrey
Advisor

Dr. Leonardo Chang Fernández
Tecnológico de Monterrey
Advisor

Dr. Heydi Méndez Vázquez
Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV)
Committee Member

Dr. Gilberto Ochoa Ruiz
Tecnológico de Monterrey

Committee Member

Dr. Yoanna Martínez Díaz
Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV)
Committee Member

Dr. Rubén Morales Menéndez
Dean of Graduate Studies
School of Engineering and Sciences

Mexico City, December 5th, 2022

Declaration of Authorship

I, Luis Santiago Luévano García, declare that this thesis titled, ”Closing the Gap on Affordable Real-Time Very Low Resolution Face Recognition for Automated Video Surveillance” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this dissertation is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Luis Santiago Luévano García
Mexico City, December, 2022

Dedication

To my closest family Luis Felipe, Dora Elizabeth, and Liz for their unconditional confidence, love, and support throughout my life's endeavors. This dissertation would not have been completed without their unwavering support.

Acknowledgements

I would like to express my deep gratitude towards my advisors at Tecnológico de Monterrey, Dr. Leonardo Chang, and Dr. Miguel González Mendoza, and our colleagues at CENATAV, Dr. Heydi Méndez Vázquez and Dr. Yoanna Martínez Díaz for their tremendously good guide and insights throughout this project. I also acknowledge the tuition support from Tecnológico de Monterrey and the coverage of living expenses by CONACyT, making this research project possible.

Closing the Gap on Affordable Real-Time Very Low Resolution Face Recognition for Automated Video Surveillance

by

Luis Santiago Luévano García

Abstract

Public and private security is a worldwide problem where efficient and automated video-surveillance technologies have a lot of potential. In an emerging country like Mexico, a functional real-time automated video-surveillance system will have a very positive social, economic, and technological impact. Proposing an open framework for face recognition at very low resolutions which public and private institutions could implement and take advantage of, will ultimately benefit our society and contribute to the state of the art in terms of efficacy and efficiency. Currently, efficient face recognition for automated video surveillance is not present within the reach of public institutions and much less so for the smallest business establishments, such as convenience stores and small offices. To make an impact in this area, the scientific problem that we are focusing on solving is the one of effectively and efficiently extracting robust facial features from Very Low Resolution face images from surveillance footage, to perform the appropriate subspace projection, and perform the posterior face identification using a dataset reference, in order to improve in efficiency terms. In this thesis, we propose solving this problem using our novel method, BinaryFaceNet, with state-of-the-art training methodology and advancements in the Binary Neural Network (BNN) and Lightweight Convolutional Neural Network (CNN) literature. The implementation of our method makes accurate and real-time face recognition available for affordable ARM-based embedded devices, with limited identification and verification performance penalties while achieving an inference performance of less than 90% latency against state-of-the-art BNNs. We finally discuss the feasibility of implementing BNN technology on extremely limited hardware, the compromises made to achieve maximum efficiency, training stable ultra-compact binarized models, and provide future work directions to complement this proposal. Finally, in our concluding remarks, we summarize the research work done and the research outcomes during the tenure of this thesis project.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Automated face recognition (FR) using computers is a problem that has been of the scientific community's interest for almost six decades now. Applications for facial recognition include surveillance tracking and biometrics authentication [68], most importantly. Advancements in computer hardware now allow us to process millions of images using a personal consumer-grade computer, giving us the ability to research more robust and accurate pattern recognition models. Today, many face recognition solutions can run in real-time on a smartphone. However, there are still major challenges in this area including recognition in uncontrolled environments, image resolution, image artifacts, and the overall robustness, reliability, and inference runtime for a real-time working surveillance application.

In this thesis work, we analyze the problem of real-time face recognition over Very Low Resolution face images from unconstrained surveillance scenarios using affordable limited hardware and we propose a working approach for this specific scenario. This problem is important to analyze due to the fact that modern approaches for this scenario require expensive hardware to perform face recognition and some are virtually unfeasible to run in real-time. The use of heavier methods translates to more computation time, energy, budget, and limits the amount of matches that can be performed at the same time. Our efficient approach suited for real-time applications, named BinaryFaceNet, is a Binary Neural Network (BNN) designed for maximizing efficiency performance, while having competent face identification and verification performance on state-of-the-art benchmark datasets for the Very Low Resolution Face Recognition problem, compared to state-of-the-art BNNs. Finally, we provide the concluding remarks based on our research results.

1.1 Traditional face recognition

Traditional face recognition methods have been successful in environments with controlled illumination and no occlusion, where cameras of High Resolutions are employed with a corresponding High Resolution facial region of interest. Face recognition in uncontrolled scenarios is especially challenging Robustness has been added to face recognition algorithms by doing face pose estimation, face frontalization, and simulating conditions via data augmentation (random occlusion, cropping, and rotating). In general, High Resolution in the face region allows us to get more discriminant features from the face and as such, dealing with artifacts like Very Low Resolution images is extremely challenging [75]. Recognition performance starts to heavily degrade in scenarios with face region areas of 32×32 pixels and less as [12].

The face recognition pipeline, depicted in Figure 1.1, models different aspects of the face recognition problem as its own sub-problem. It consists of the following steps [92, 68]:

1. Face detection: extract the face(s) from an image.
2. Face alignment: estimate face orientation and warp the face image to a frontal face template.
3. Feature extraction: extract the most discriminant features from a face. Select global and/or local features.
4. Face representation: select an appropriate descriptor for the face image and features.
5. Face recognition step or model training: use the descriptor either to compare against the trained model or use the image for training. Match images by employing similarity measurements to score the query image with the images saved in the model.

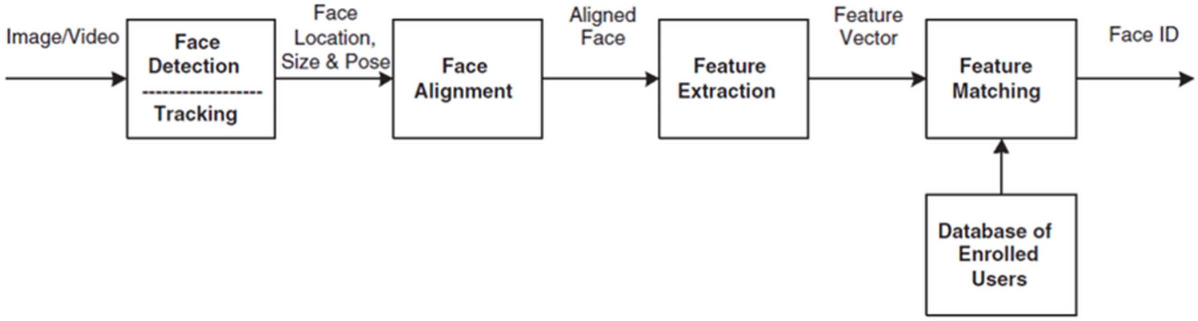


Figure 1.1: Face recognition pipeline as illustrated in [68]. It consists on detecting the face image, normalizing it, extracting relevant features and use them compare against a previously enrolled biometric face template.

For the face detection step, classic algorithms, such as the Boosting approach proposed by Viola and Jones [119], are accurate and very efficient for runtime for real-time applications. More modern Deep Learning-based approaches include Retinaface [30] and MTCNN [144], where the latter detects and aligns faces in a single step. Standard methods for face recognition are the ones based on holistic representations and human-engineered features. Holistic representation approaches include Eigenfaces [116] and Fisherfaces [7], aiming to mathematically represent an identity using whole facial features. Later, feature-based face recognition was performed using extracting Local Binary Pattern Histograms [34] and Gabor [24], SIFT [73], and SURF [6] features. Feature matching using Principal Component Analysis (PCA) and Linear Discriminant Analysis-based matching [118] is also common. Works with learned descriptors such as [17] also started to emerge. However, these methods struggle with capturing the non-linearity of face appearances in unconstrained scenarios. The face recognition task has four variants: open or close set identification (one-to-many probe and gallery matching) or verification (one-to-one probe and gallery matching). Open-set refers to the task where the probes can either appear or not in the gallery, while closed-set refers to the task where probes always appear in the gallery [123].

Benchmark	Mean Height	# Identity	# Image
LFW [56]	119	5,749	13,233
VGGFace [88]	138	2,622	2.6M
MegaFace [60]	352	530	1M
CASIA [136]	153	10,575	494,414
IJB-A [61]	307	500	5,712
IJB-B [61]	$>$100	1,845	21,798*
IJB-C [82]	$>$100	3,531	31,334*
CelebA [71]	212	10,177	202,599
UMDFaces [5]	$>$100	8,277	367,888
MS-Celeb-1M [45]	$>$100	99,892	8,456,240
MegaFace2 [85]	252	672,057	4,753,320
YouTube Faces [129]	100	1,595	621,126

Table 1.1: Summary of popular face recognition benchmark datasets, expanded from [22]. We note that all of them have a mean height of more than 100 pixels, considered High Resolution. *Only accounts for still-image data.

Since the introduction and explosive growth of Deep Learning in 2011, starting with AlexNet [62] and VGG Convolutional Neural Networks (CNNs) [106] for general-purpose Computer Vision recognition tasks, modifications of various Convolutional Neural Network concepts have been made to tackle other specific Computer Vision problems [39, 36, 89] as well as for the face recognition problem, using millions of face images for training. Some of these approaches include VGGFace [88], DeepFace [111], and FaceNet [101], making a tremendous leap in face recognition performance in benchmark datasets including Labeled Faces on the Wild (LFW) [56], CelebA [71], and YouTube Faces [129]. These methods are particularly successful with train and test cases in uncontrolled scenarios [123]. Table 1.1 contains a summary of popular benchmark datasets for face recognition purposes, detailing the mean height of the facial region of interest, the number of identities and the number of images. All of the referenced datasets have a region of interest with a mean height of more than 100 pixels.

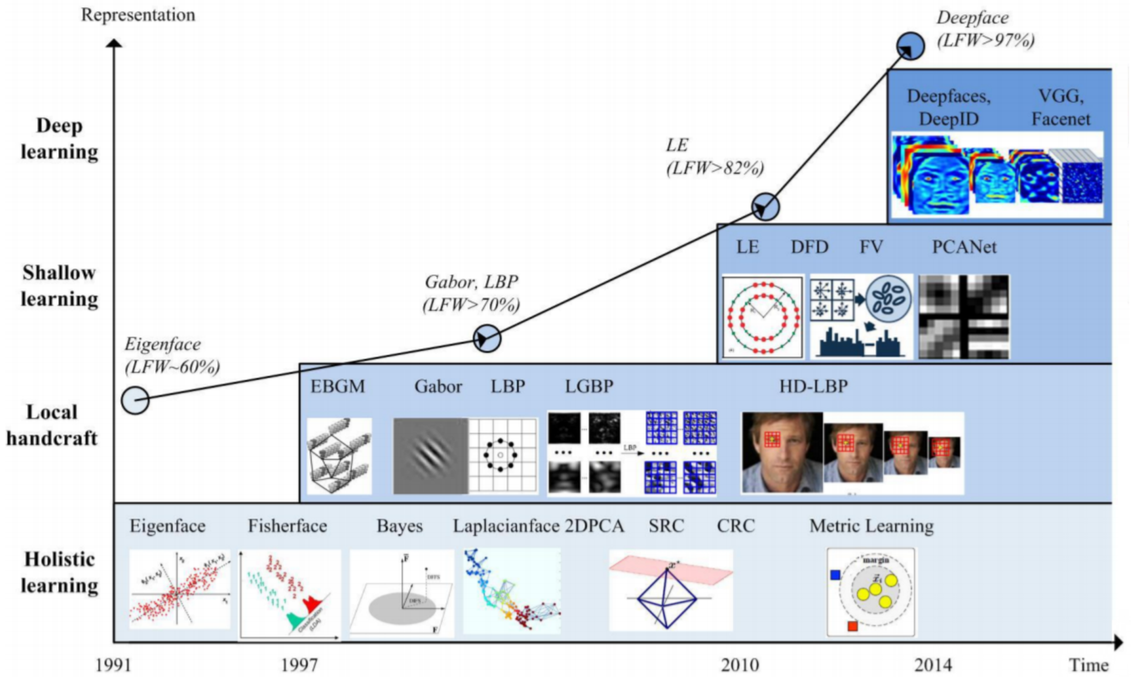


Figure 1.2: Face recognition accuracy performance on the Labeled Faces on the Wild (LFW) dataset and method type proposals over the years, taken from [123]. Earlier holistic approaches struggled with unconstrained data, while Deep Learning-based methods capture image condition non-linearity better, at the cost of computational complexity.

As expected, many of the solutions proposed nowadays are Deep Learning-based approaches, since more datasets and benchmarks are available as well as the computational power means to train and test these heavier methods. Figure 1.2 shows the improvement from classic FR approaches to these newer CNN-based approaches. However, training and testing on these very deep networks pose challenges for run-time on less-capable devices such as laptops, mobile phones, and embedded systems and the need for fast and accurate face recognition systems. In recent years, efficient CNN architectures for genera-purpose Computer Vision tasks have emerged, with the most prominent being MobileNets [55, 99, 54], ShuffleNets [147, 76], EfficientNet [113], and VarGNet [146]. This approach type is commonly called Lightweight Convolutional Neural Networks, commonly aimed to run under model size and complexity constraints, such as less than 20 MegaBytes and 1 Giga FLOps (Floating-point Operations per Second). Lightweight CNNs for face recognition include variations on these architectures. MobileFaceNet [20], ShuffleFaceNet [79], and [132] obtain a good face recognition accuracy and efficiency trade-off in traditional face recognition surveillance scenarios [32, 81] and also in masked face recognition scenarios [29].

1.2 Very Low Resolution Face Recognition (VLR FR)

High resolution face recognition, as discussed in the previous section, has been successful in unconstrained FR benchmarks [123, 32]. However, the challenge of achieving fast and accurate Very Low Resolution Face Recognition over unconstrained scenarios still remain. Consistent with previous literature [75, 22, 124], we define *Very Low Resolution (VLR)* as the face regions with a resolution area of 32×32 pixels or less. Classic approaches for VLR FR include evaluations in datasets such as CEMU-PIE [44] and FERET [90], which are good benchmarks for a Very Low Resolution Face Recognition in controlled environments, where hand-crafted methods perform very well in both recognition performance and run-time [65]. Additionally, naïve data augmentation methods, mainly bilinear interpolation, are extensively used to generate synthetic image pairs in low resolution and High Resolution and still get an accurate FR performance. Nevertheless, a real-world uncontrolled setting is still very challenging due to occlusion, pose differences, and distortion artifacts, where these classic methods perform very poorly.

Factors that can affect recognition performance include the type of camera being used, the distance between the object and the camera, the domain disparity between gallery images in controlled environments against the Very Low Resolution captured face from the camera feed (when the gallery image is available), the training and fine-tuning datasets, and the degradation of features at this extreme low resolutions [47]. Experiments have been done to assess trained human performance against automatic systems, where -in those specific experimental settings- human face recognizers achieved 93% of accuracy [96].

1.2.1 Datasets

Currently, efficient Very Low Resolution Face Recognition under surveillance scenarios is a very niche research area, with limited datasets available. In recent efforts for expanding these types of studies, unconstrained datasets with native VLR imagery have been proposed. Such datasets are the QMUL-SurvFace[23], QMUL-TinyFace[22], and IJB-S[59]. Other challenging benchmark datasets for the Very Low Resolution Face Recognition under surveillance scenarios include the following: SCface[42], Point and Shoot Challenge[9], and UCCS Face[100].

Database name	Source	Quality	Static image/video	# subjects	# images
Point and Shoot[9]	Manually Collected	HR + blur	static + video	558	12,178
SCFace[42]	Surveillance	HR + LR	static	130	4,160
QMUL-Survface[23]	Surveillance	VLR	static + video	15,573	463,507
QMUL-TinyFace[22]	Web	VLR	static	5,139	169,403
UCCSface[100]	Surveillance	HR + blur	static	308	6,337
IJB-S[59]	Surveillance	HR + VLR	static + video	202	3 million+

Table 1.2: Summary of unconstrained surveillance datasets of interest. Taken from [75]. Most datasets contain HR and VLR image pairs, however, the largest-scale datasets include native VLR images with their subject labels only. This further complicates the native unconstrained VLR face recognition scenario. LR refers to images below 100×100 pixels of face area resolution but not below 32×32, as opposed to VLR images below 32×32.

Table 1.2 summarizes the datasets available for the Very Low Resolution Face Recognition specific task. The source column indicates from which scenario the data was obtained,

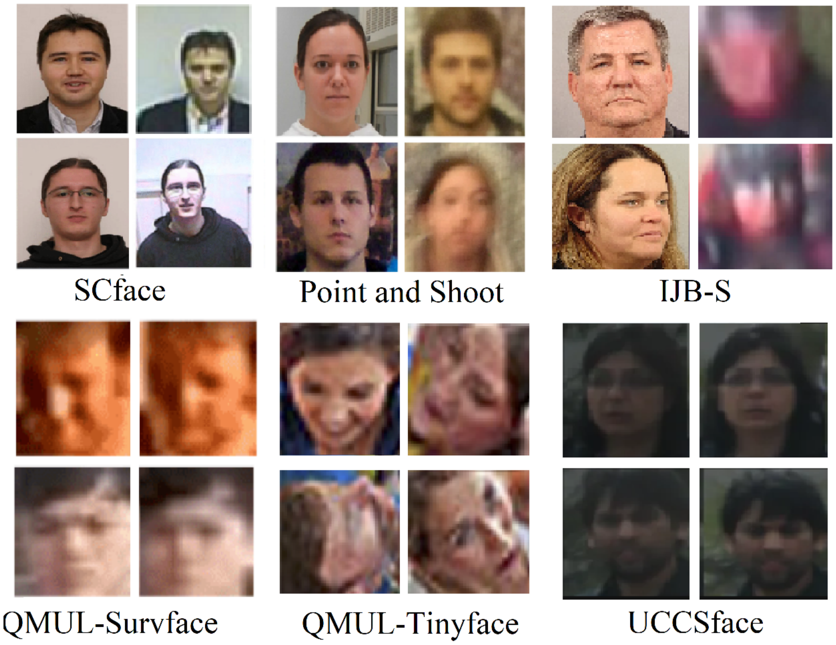


Figure 1.3: Samples from popular VLR FR datasets, taken from [75]. The domain difference from the reference image to the VLR surveillance footage, unconstrained conditions, and lack of features make this problem challenging for classic face recognition methods.

.

the quality column indicates the type of available images where ”HR” corresponds to High Resolution imagery, ”VLR” to Very Low Resolution imagery, ”LR” to Low Resolution imagery between 32×32 and 100×100, and ”blur” to blurred Low Resolution images, the static image/video column indicates whether the dataset has static images and also contains video data or not, and the next columns indicate the total number of identities and number of images (including images from videos). The datasets containing native HR and VLR data are suitable for heterogeneous face recognition approaches, where a reference gallery image is available to compare the VLR probes against. The QMUL-Survface and QMUL-TinyFace only include native Very Low Resolution face imagery, making them most suitable for homogeneous face recognition approaches, where there is no distinction when processing data from different sources. It is also possible to simulate the heterogeneous scenario and the homogeneous scenario using image interpolation methods.

1.2.2 VLR FR approaches overview

Classic approaches for attempting to solve the VLR FR problem include those based on PCA for feature extraction and Linear Discriminant Analysis or similar approaches for projecting High Resolution and Very Low Resolution images in a common subspace and posterior classification [65, 11, 104]. Newer Deep Learning-based alternatives have been proposed as well. Some of these CNNs are focused on modeling the relationship using real or synthetic HR and VLR pairs such as the, Transferable Coupled Network (TCN) [140], the Feature Aggregation

Network (FAN) [137], and the Complement Super Resolution and Identity (C-SRI) [22]. Homogeneous face recognition using Regular Lightweight CNNs for face recognition has also been explored [80], highlighting the importance of simulating VLR conditions using different image interpolation methods. We can classify the VLR FR approaches as follows for the heterogeneous VLR FR variant of the problem: solutions commonly classified as Coupled Mappings (also called Projection methods or Domain Adaptation methods) and Super Resolution (also called Synthesis Methods) techniques. Coupled Mappings aim to project images to a single latent subspace and Super Resolution methods intend to produce a better-quality image before performing classification. Methods for addressing the homogeneous VLR FR variant, without domain distinction, mostly include traditional face recognition methods, using different interpolation methods, as previously discussed. An in-depth explanation of related methods and theoretical framework for our solution is provided in Chapter 2.

1.3 Relevance

Public and private security and safeguarding, to this day, still have many areas of improvement, both in developed countries and on underdeveloped ones. Talking about public security, in the particular case of Mexico, the government is actively investing in video-surveillance equipment. However, existing security systems, still require manual human operation to monitor, call up, and attend to any emergency. In Mexico City's metropolitan area alone, more than 35,000 cameras have been installed since 2015, with a special focus on crowded places such as banks, stores, schools, hospitals, and public transportation [3]. Additionally and perhaps more importantly, some smaller commercial establishments wouldn't have the budget to invest in the best-in-class cameras for optimal operation and automated video-surveillance performance, hindered by heavy distortion and low resolution artifacts. We consider that solving the efficiency problem associated with Very Low Resolution Face Recognition would greatly improve the state of the art on automated video-surveillance systems, in terms of costs and operational efficiency. Approaching the VLR FR problem from an efficiency perspective directly tackles the social and operational implications of the problem at hand [83]. As of this moment, there is no automated face recognition and automated video surveillance within the reach of this smaller type of commercial establishments and the general public.

Some caveats with the use of state-of-the-art surveillance systems in Mexico, as detailed on [83], are:

- Being able only to monitor only a limited number of cameras at the same time. If a person can monitor around 10 cameras, it is unfeasible to have the required amount of people at a surveillance center in order to monitor the whole set of the surveillance system with 8-hour-a-day shifts for uninterrupted monitoring [95].

- Human errors and high cost of acquiring the surveillance equipment.

- For much of the crime that happens in real-time and is being recorded on camera, it still is very hard for a human operator to identify it and current systems still rely on manual emergency calls in order for the police to act.

- Some of the footage is not even being monitored in real time, it is only used for later procurement.
- Economic inequality is expressed in the sophistication of security in certain areas only, as well as in the levels of criminality and public insecurity. In this case, poor neighborhoods homogeneously segregated live in a climate of major violence and insecurity.
- The use of video surveillance systems in certain areas only contributes to the building of stereotypes and social prejudices, which negatively affect the acceptance of diversity and citizenship building.

1.4 Problem description

As Computer Vision researchers at Tecnológico de Monterrey, we are highly committed to the economic, political, social, and cultural development of our community. To fulfill our mission, this doctoral thesis work contributes to efficient facial recognition technologies for the low resolution unconstrained surveillance context. We address the scientific problem of **efficiently extracting robust facial features, accurately projecting them, and matching them in a unified space from Very Low Resolution surveillance footage and High Resolution gallery face images indistinctively**. Considering Very Low Resolution as a face image of 32×32 pixels and less, a High Resolution face image of 100×100 pixels, and real time of 5 to 10 Frames Per Second (FPS). We take into account as affordable hardware the Nvidia Jetson platforms and mobile ARM processors. We consider that solving such problem would greatly improve the state of the art on automated video-surveillance systems in terms of costs and operational efficiency, as well as solving the security and social problems detailed in the previous section.

Classic hand-crafted methods struggle with capturing the non-linearity of the unconstrained surveillance conditions and the projections from both resolution domains in a unified space. Leveraging recent developments on efficient Deep Learning-based methods can lead to more robust face descriptors [32]. Most successful methods for face recognition in VLR unconstrained scenarios rely on heavily taxing algorithms based multiple-branch CNNs or dense block designs [74, 140], which are not feasible to implement for real-time applications sometimes even with a high amount of computational resources [107, 74, 22]. Growing interest towards Lightweight CNNs, Quantized and Binarized Neural Networks (BNNs) for VLFR are still niche research areas [75].

We have identified that the factor that mostly hinders unconstrained VLR FR performance is the extraction of features robust to noise, scale, and orientation at such low resolutions. These problems are accentuated by occlusion, blur and face pose variations. As such, we are convinced that the way to create a more efficient and accurate method for this scenario is to employ the guidelines from the most recent developments in the design of CNNs for face recognition [33, 31, 79], Lightweight CNNs [99, 76, 54], robust feature extraction at VLR using residual networks [74], and quantized and binarized neural networks [4, 8].

Another problem that we have identified in the state of the art is the data augmentation strategies used to both create datasets and add robustness at the training step. Some of the methods use naïve bilinear interpolation to create synthetic VLR datasets or upscaling images for feature extraction and matching. The generated images do not represent the native real-world VLR images captured by a surveillance camera. There is an area of opportunity for leveraging data augmentation and interpolation methods. This can lead to improving robustness in the feature representation.

1.5 Objectives and contributions

The main objective of this thesis work is:

To create a Very Low Resolution Face Recognition method that efficiently extracts facial features, projects them, and matches them in a unified space from native surveillance footage and High Resolution face images indistinctively, in real-time (5-10 FPS) on an Nvidia Jetson Nano platform; while achieving comparable state-of-the-art accuracy and True Acceptance Rate-False Acceptance Rate (TAR@FAR) trade-off with faster inference time performance of a magnitude of at least 2× from successful Quantized Deep Learning-based methods from the state of the art on VLR benchmark datasets.

The designed architecture will need to achieve an adequate accuracy-efficiency trade-off to not use all the hardware resources on the platform and leave a reasonable headroom for the rest of the face recognition pipeline.

As such, the specific objectives are the following:

1. To reate a binarized approach for matching the probe face images and the enrolled gallery face images across different resolutions and conditions, with a real-time efficiency performance. We will assess the efficiency-accuracy trade-off with the amount of compromised precision percentage against the reduced inference latency compared to the state-of-the-art BNNs on VLR benchmark datasets.

2. To use an effective image interpolation strategy, to improve robustness by better simulating VLR conditions. The influence of more adequate data interpolation techniques such as bicubic and inter-area interpolation will be tested with the performance metrics identification accuracy and TAR@FAR verification rates when including such techniques and removing them to correctly measure their impact on the overall method's performance.

3. To achieve a runtime inference performance of 5 to 10 FPS in a single core on affordable ARM hardware (defined in Section 1.7). The inference runtime tests on target hardware are performed for the face recognition step of the pipeline and reported in seconds-per-image and FPS to evaluate this objective.

4. To propose a BNN architecture with at least 50% fewer operations and 50% less inference time latency compared to state-of-the-art BNNs, with more than 95% binarized

operations. This objective will be assessed with the total Multiply Accumulate operations, inference time performance in seconds per image, and the binarized operation ratio of the total network.

Efficient VLR FR for embedded systems is a very niche research area. Many VLR FR methods are not focused on bringing this surveillance application to a real-time scenario. This thesis project presents the following contributions to the state of the art:

- An original specialized method for VLR face recognition based on state-of-the-art efficient BNN approaches running in real-time on affordable hardware. This original approach does not solely rely on Lightweight CNN methods for face recognition previously tailored for HR FR.

- An approach for efficient and robust feature extraction that achieves a comparable identification and verification performance on the unconstrained VLR FR setting.

- The democratization of automated video-surveillance technologies with state-of-the-art real-time algorithms that run on affordable hardware, to reach the broadest possible public audience. The source of this research project is publicly available [a].

- An in-depth study of the limitations of automated Face Recognition at Very Low Resolutions, the implementation of binarization technologies, and processing constraints on power-efficient ARM platforms.

1.6 Hypothesis and research questions

If we create a flexible Quantized Neural Network architecture, with the latest developments in Lightweight Convolutional Neural Networks for face recognition and Binarized Neural Networks, we will effectively close the gap for a real-time automated video-surveillance application for Very Low Resolution Face Recognition in the wild, while maintaining a limited compromise on face recognition performance compared to the state-of-the-art BNNs.

As such, the research questions are:

- Which modifications from efficient network architectures would yield satisfactory results for any given recognition step, in accordance with our objectives?

- Which VLR FR approach would be the most appropriate to use to add robustness to the face descriptor while lowering the inference time for a real-time application?

- Which resolution would be the lowest for which we can produce usable results, such as having more than 50% accuracy on benchmark datasets while meeting our run-time objectives?

- Which aspects of an efficient network architecture (MACs, model size, number of parameters) affects the most for being able to implement a real-time constrained surveillance application?

[a] The code repository for this research project is available at: `https://github.com/lluevano/insightface_larq_keras`

1.7 Hardware setup

The focus of this proposal is for achieving real-time detection, alignment, and verification in affordable systems. The proposed hardware for verifying these requirements is:

- Nvidia Jetson Nano, equipped with an ARM quad-core ARM A57 CPU, 2GB RAM, 16GB eMMC, Nvidia Maxwell GPU with 128 CUDA cores at 921MHz.

Training will be performed using two systems with AMD Ryzen 7 processors, 64Gb RAM, NvME PCIe M.2 solid state drives, and two Nvidia GTX 1080Ti graphics cards.

The next section will outline the state of the art with respect to the variety of topics surrounding the research of Very Low Resolution and face recognition and Quantized Neural Networks; followed by our proposed solution, BinaryFaceNet, methodology, and concluding remarks.

Chapter 2

Related work, Remarks, and Theoretical Framework

In this chapter, we discuss in-depth the most relevant approaches for VLR face recognition, as well as the theoretical foundations for the components of our proposal, BinaryFaceNet.

2.1 Related work

In this section, we present approaches of the Very Low Resolution Face Recognition literature and also for the newer most efficient approaches for image recognition using Quantized Neural Networks, which are relevant to our efficient proposal for approaching our real-time scenario on embedded devices. Figure 2.1 shows the two main variants present in the VLR FR literature: Heterogeneous Face Recognition and Homogeneous Face Recognition approaches. Heterogeneous Face Recognition approaches attempt to model the domain difference between VLR and HR images, and Homogeneous Face Recognition methods do not make this distinction directly in the method definition.

2.1.1 Heterogeneous Face Recognition approaches for Very Low Resolution

Some of the biggest challenges in the context of surveillance using very low-resolution images are the domain disparity from the high resolution reference image in controlled conditions (gallery) and the Very Low Resolution native probe images, the lack of features at a Very Low Resolution space, and the runtime of the solutions, usually requiring a high amount of computing power such as graphical processing units (GPUs) with a high amount of VRAM and computing capabilities. This subsection details subspace projection-based approaches, also commonly called Coupled Mappings (CM) or Multidimensional Scaling (MDS).

2.1.1.1 Projection methods: Coupled Mappings

Coupled Mapping methods aim to find an adequate representation of data from different domains by projecting the data into a single unified space. In our scenario, they match images

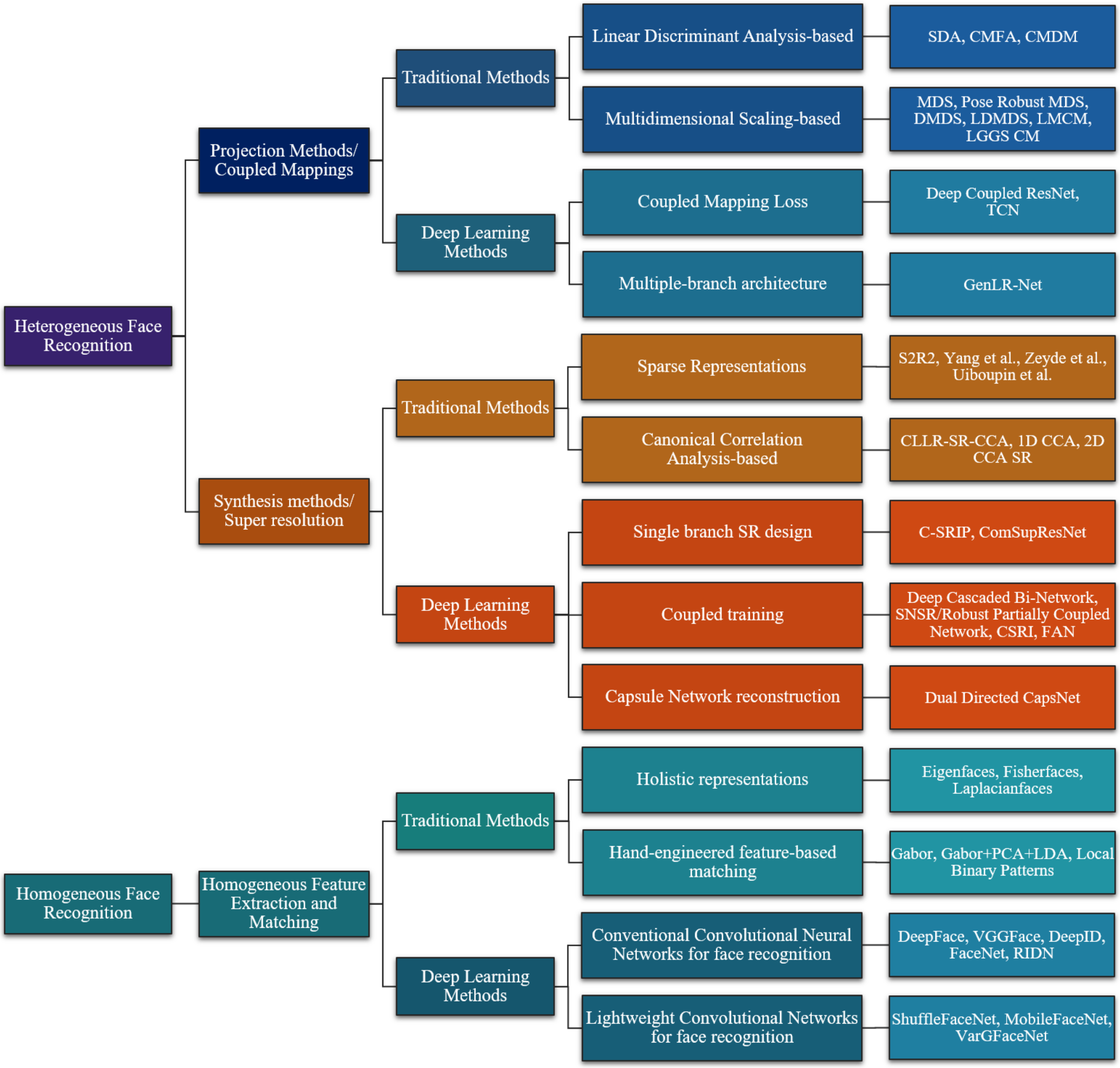

Heterogeneous Face Recognition
Projection Methods/ Coupled Mappings
Traditional Methods
Linear Discriminant Analysis-based
SDA, CMFA, CMDM
Multidimensional Scaling-based
MDS, Pose Robust MDS, DMDS, LDMDS, LMCM, LGGS CM
Deep Learning Methods
Coupled Mapping Loss
Deep Coupled ResNet, TCN
Multiple-branch architecture
GenLR-Net
Synthesis methods/ Super resolution
Traditional Methods
Sparse Representations
S2R2, Yang et al., Zeyde et al., Uiboupin et al.
Canonical Correlation Analysis-based
CLLR-SR-CCA, 1D CCA, 2D CCA SR
Deep Learning Methods
Single branch SR design
C-SRIP, ComSupResNet
Coupled training
Deep Cascaded Bi-Network, SNSR/Robust Partially Coupled Network, CSRI, FAN
Capsule Network reconstruction
Dual Directed CapsNet
Homogeneous Face Recognition
Homogeneous Feature Extraction and Matching
Traditional Methods
Holistic representations
Eigenfaces, Fisherfaces, Laplacianfaces
Hand-engineered feature-based matching
Gabor, Gabor+PCA+LDA, Local Binary Patterns
Deep Learning Methods
Conventional Convolutional Neural Networks for face recognition
DeepFace, VGGFace, DeepID, FaceNet, RIDN
Lightweight Convolutional Networks for face recognition
ShuffleFaceNet, MobileFaceNet, VarGFaceNet

Figure 2.1: Taxonomy for the VLR FR literature landscape, taken from previous work [75]. We can mainly divide the state-of-the-art approaches into Heterogeneous FR and Heterogeneous FR. These main approaches can be further classified in modern Deep Learning-based methods, contrasting with traditional approaches.

from the High Resolution domain with the ones from the Very Low Resolution space, projecting both domains to a single unified space with different considerations per source domain. Figure 2.2 graphically illustrates the idea of CM methods and the difference in the subspace projections using MDS [11], Discriminative MDS (DMDS) [133], and Large Margin Coupled Mappings (LMCM) [141], where the most successful traditional approaches enforce projected distance and margin constraints.

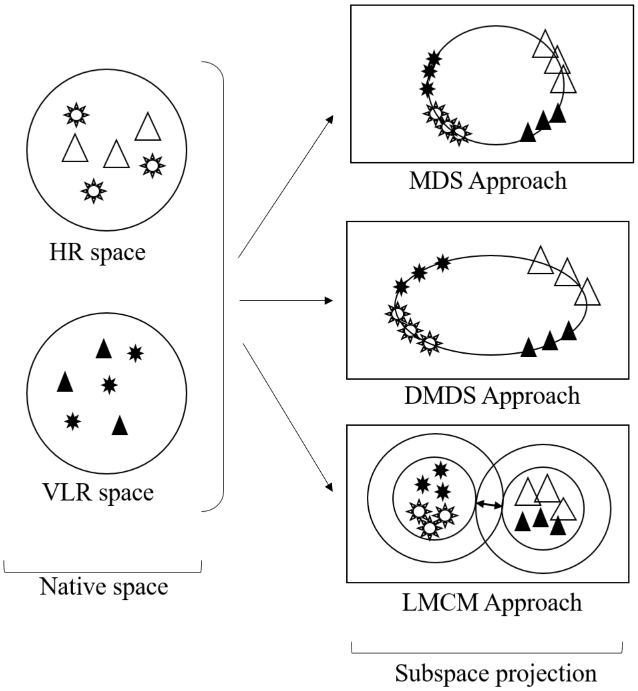


Figure 2.2: Basic idea of Coupled Mappings methods and subspace projection for Coupled Mappings methods MDS [11], DMDS [133], and LMCM [141]. Samples of the same class appear across different domains, as such, learning a projection function to have all the samples in the same space allows for a one-to-one sample normalized sample comparison. These methods include constraints to promote a closer projection of intra-class samples and a larger projection of inter-class samples and promote margin-maximization in the projected space.

2.1.1.1.1 Classic Coupled Mappings Methods

Classical methods for coupled mappings include those mainly based on classical Linear Discriminant Analysis (LDA) techniques, such as **Simulataneous Discriminant Analysis (SDA)** [18], **Coupled Marginal Fisher Analysis (CMFA)** [105], and **Coupled Marginal Discriminant Mappings (CMDM)** [145]. SDA uses individual classic LDA-based scatter matrices to project the images from the different domains. CMFA used the objective function to minimize the ratio of the sum of distances of the inter-class and intra-class projections. CMDM trains using parameters for describing the similarity of the scatter matrices, modeling the relationship as an eigen-decomposition approach.

The **Multidimensional Scaling (MDS)** [11] approach projects the images from both domains using a transformation matrix and adding constraints to optimize the distance of the projected feature vectors to the samples from the HR domain. MDS-based approaches use the iterative majorization [127] optimizer. This seminal approach inspired later approaches **Pose-Robust MDS** [10], **Discriminative MDS (DMDS)** [133] and **Local-Consistency Preserved DMDS (LDMDS)** [133], **Large Margin Coupled Mappings (LMCM)** [141], and **Local Geometry to Global Structure CM** [104].

The Pose-Robust MDS approach proposed in [10] aims to additionally model the median and mode information separately for different face orientations. DMDS [133] adds constraints to the scatter matrices for optimizing class distances in the projected subspace. In its extension, LDMDS [133], they added constraints pertaining to the source domain of each face image. The Local Geometry to Global Structure method [104] generates uses a k-neighbor voting system, much like classic clustering techniques, as a constraint in the class projected samples. Inter-class neighboring samples are heavily penalized, enforcing inter-class margins. Finally, a global projection matrix is built by concatenating the intra-class mappings from the different source domains.

2.1.1.1.2 Deep Learning-based Coupled Mappings Methods

Some Deep Learning-based methods for VLR FR have focused on the homogeneous variant of the problem, that is, aiming to build robust face descriptors from VLR and HR images indistinctively [67, 50]. This means that they do not attempt to model the relationships of the different domains in the model optimization step or the network architecture. Methods that model this relationship are **Deep Coupled ResNet** [74], **GenLR-Net** [84], and **Transferable Coupled Network (TCN)** [140].

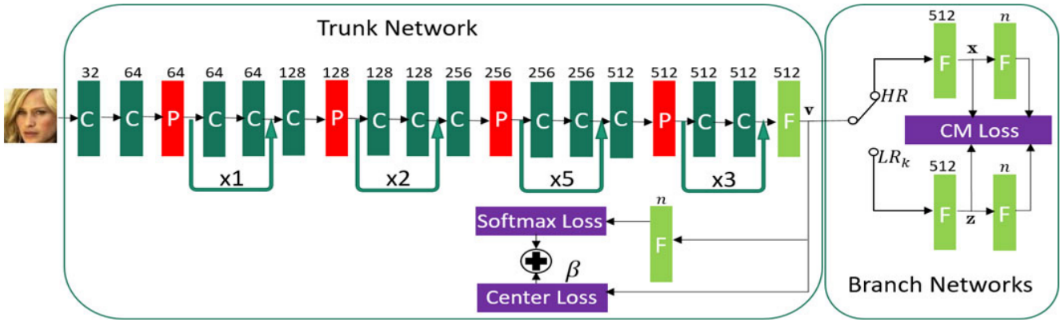


Figure 2.3: Deep Coupled ResNet architecture, taken from [74]. This trunk-branch design models the VLR to HR relationship on the branch networks and in the Couple Mapping loss at the optimization step.

Figure 2.3 shows the Deep Coupled ResNet architecture. This method uses a trunk-branch architecture, based on ResNet [48] blocks. The network features branch networks with Fully Connected (FC) layers for VLR and HR source domains separately. The network is optimized using their proposed Coupled Mapping loss, Softmax loss, and Center Loss [128]. The ResNet blocks provide a robust feature extraction while using PReLU activations consistent with state-of-the-art HR FR network design principles [81].

Then GenLR-Net method, illustrated in Figure 2.4, uses two trunk networks to separately process VLR and HR images. The VLR images are upscaled with a single convolutional layer and its separate loss function. The VLR trunk network is optimized using multiple Contrastive losses between FC layers, leaving the HR section pre-trained.

The TCN architecture (Figure 2.5 also involves two subnetworks for processing VLR and HR input images separately, with the HR subnetwork already pre-trained. They tested the ResNet and VGG backbones, with the ResNet architecture yielding better results. In the optimization step, they proposed using the triplet loss with separate anchors per source

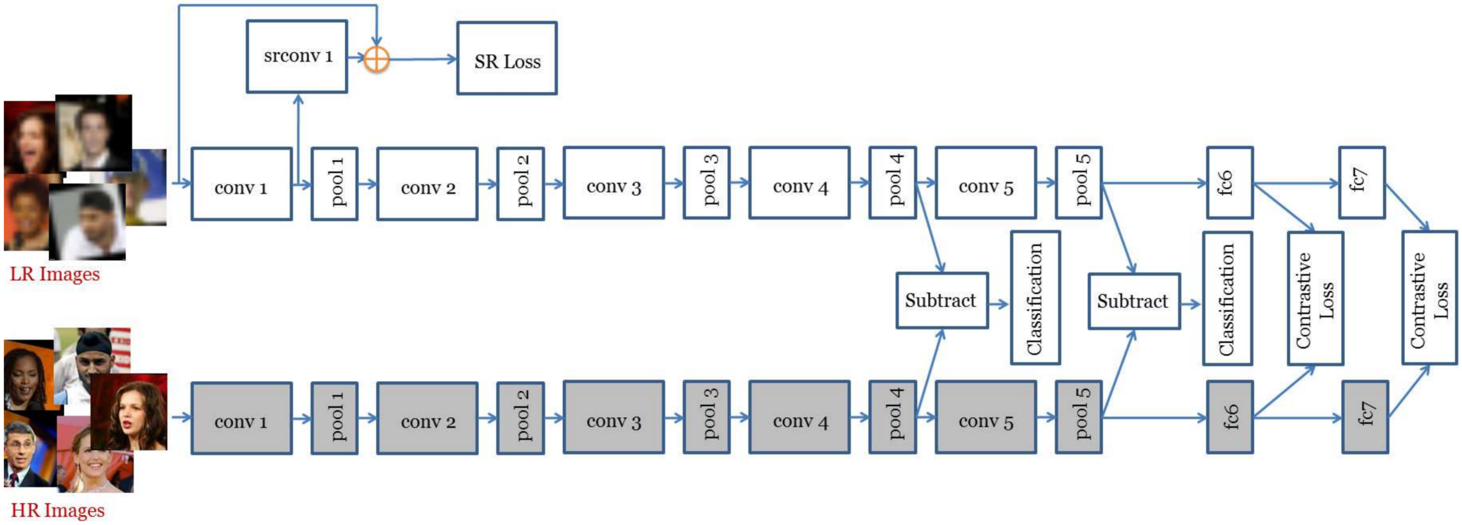


Figure 2.4: GenLR-Net architecture, taken from [84]. This method employs the VGG Face CNN as its base for feature extraction. The HR trunk network is pre-trained and the VLR trunk network is fully trained with multiple Contrastive losses.

domain, and updating them with samples from the opposite domain as well as enforcing inter-class margins for improving discriminability on the projected subspace.

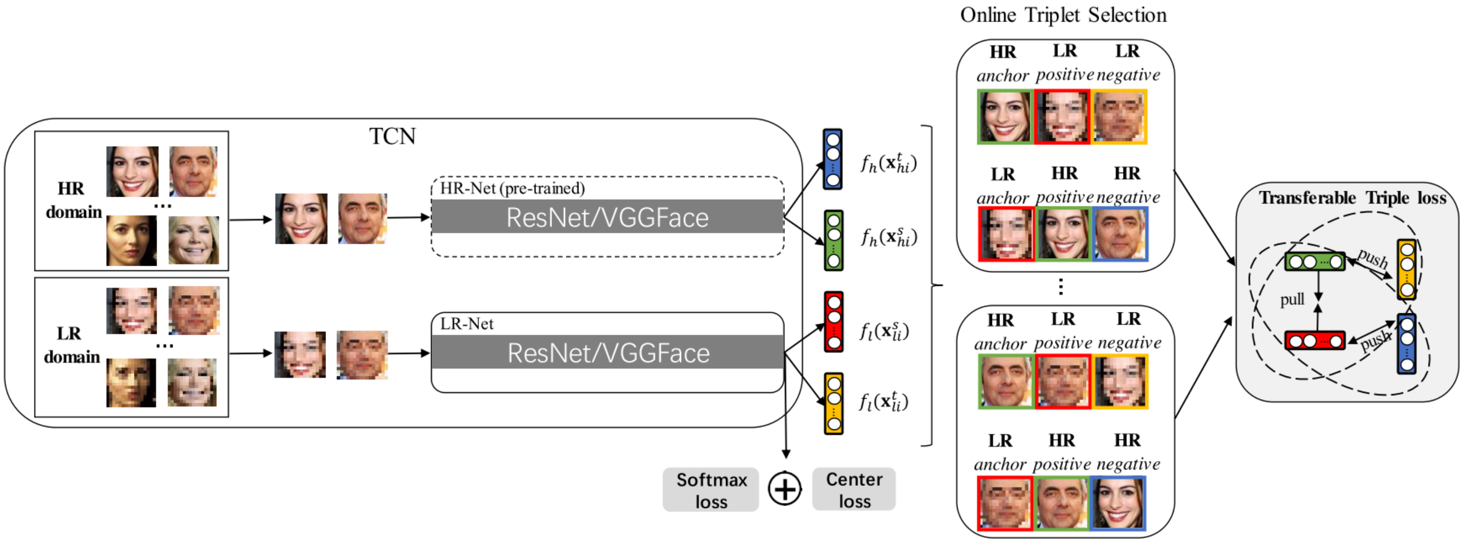


Figure 2.5: The Transferable Coupled Network (TCN), taken from [140], uses two trunk networks to process HR and VLR imagery separately, then optimizes the model using the triplet, softmax, and center losses.

Most CM methods provide benchmarks for accuracy performance on the Multi-PIE, FERET, and SCface datasets; but rarely for inference time performance. Only LMCM reports an inference time of 8.5 microseconds per image on an i5-4200U CPU, an x86 platform. MDS methods report performance on the SCface dataset, with LDMDS showing an 81.54% mean accuracy on this dataset. For Deep Learning-based methods, TCN reports a mean accuracy performance of 89.37%. We particularly note that enforcing inter-class and intra-class constraints with samples from the same domain in the projected subspace favors discriminability, showed by DMDS and LDMDS. The classical CM methods do not leverage the HR richer source domain to favor the subspace projection. The Deep Learning-based methods are not optimized for real-time performance, featuring heavy ResNet and VGGFace architectures, and have been outperformed by Homogeneous FR approaches by Lightweight CNNs [80]. The ResNet block design has proven to be a limiting factor when proposing novel methods or training methodologies.

As such, the main advantages and limitations of the Coupled Mapping approaches are:

Advantages

- Hand-crafted coupled mapping methods are more efficient at inference time, without the need for a powerful GPU to run in real-time inference.
- Later efforts based on Deep Learning techniques for CM are more accurate than classical CM methods, as they can extract robust and usable features. They are also more accurate than most Deep Learning-based methods for Super Resolution and traditional face recognition (not more modern lightweight CNN) approaches, specifically in the context of identifying a subject from VLR and HR image data.
- The focus of this approach type always takes into account the disparity between high resolution images and low resolution images, very intrinsic to the Heterogeneous FR problem. In contrast with other approaches, which take into consideration only certain aspects of the problem and do not always aid in recognition performance.
- Leveraging the ability of deep networks to capture non-linear relationships is very promising for both extracting robust features and accurately projecting the features in subspaces, further aiding in VLR FR accuracy performance.

Limitations

- A robust unified subspace is very hard to find, with classical methods struggling to add non-linearity to boost recognition accuracy performance.
- Current Deep Learning-based CM methods for feature extraction, projection, and matching are not optimized for running in real-time on heavy computation platforms, showing further limitations on embedded ARM platforms.
- The learning strategy, variable-resolution loss functions, handling of variable-resolution regions of interest, and network architecture have to be carefully designed as they have can have a major impact on recognition performance.

2.1.1.2 Synthesis methods: Super Resolution

Super Resolution (SR) approaches match the target domain by upsampling a VLR image into HR. Many classic SR approaches include benchmarks for 2×, 4× or 8× the source resolution.

Figure 2.6 illustrates the main idea behind Super Resolution. The process of super-resolving a face is also sometimes called Face Hallucination. In VLFR, some methods upsample the VLR face image to perform posterior recognition (or training) while others train the SR method including an FR constraint in the method optimization. The FR constraint promotes discriminability usable for FR for the super-resolved image. In some datasets, training SR methods separate from the FR model can negatively affect accuracy performance [22]. This is also shown in the Super-Identity Convolutional Neural Network for Face Hallucination (SICNN) [143]. This approach includes an FR loss for the SR network optimization process. This process can outperform other Deep Learning-based approaches, such as Laplacian SR [63].

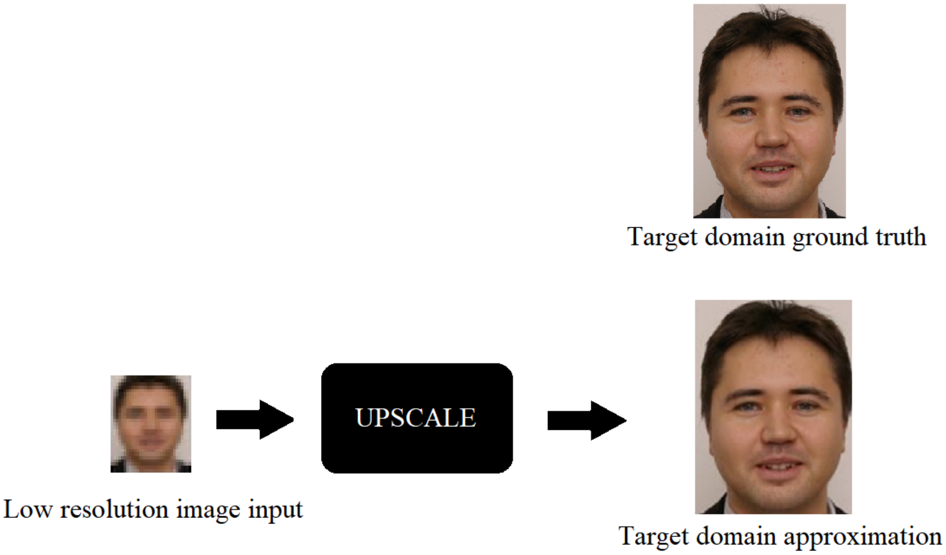

Target domain ground truth
UPSCALE
Low resolution image input
Target domain approximation

Figure 2.6: Main idea of Super Resolution approaches, taken from previous work [75]. Super Resolution methods aim to introduce a Very Low Resolution image into a High Resolution subspace, for posterior face recognition matching.

However, many methods focus on measuring the Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR). These metrics are relative to the dataset used for training and do not measure discriminability for face recognition tasks. Moreover, the many popular benchmark datasets for this task are not restrained to face images. Most contain miscellaneous imagery, such as buildings, animals, objects, etc.

Using naïve interpolation strategies only as an SR approach, such as only using bilinear interpolation for training and testing the models, presents the domain disparity problem. This disparity between native VLR datasets (from a surveillance feed) and the synthetically downscaled images drastically hinders performance on classic FR methods. Some modern approaches combine native and synthetic imagery to improve performance on native large-scale VLR benchmarks [22].

2.1.1.2.1 Classic Super Resolution approaches

This approach type mainly includes methods based on Sparse Representations and Canonical Correlation Analysis (CCA). Sparse Representation approaches include the **S2R2 matching** [49], work by **Yang *et al.*** [135], **Zeyde *et al.*** [139], and **Uiboupin *et al*** [117]. This approach type aims to find a linear combination of kernels to upscale the image. Work on CCA-based approaches include the **Coherent Local Linear Reconstruction (CLLR-SR)** [1] and the **2D CCA Face Image SR** [38] by An *et al.* The CCA-based approaches aim to model particular facial details and global information into canonical variables, perform the correlation analysis and obtain a linear combination for super-resolving the VLR image.

The S2R2 matching [49] details training for Super Resolution and face recognition. This method firstly models two matrices: one for super resolving to HR space and another one to downscale to the VLR space. This allows comparing native VLR images with the synthetic HR counterpart, and vice-versa. They also include a component to map the smoothness of the obtained HR image. The work of Uiboupin *et al.* [117] is inspired by the previous work of Yang *et al.* [135] and Zeyde *et al.* [139]. This method proposes to use one dictionary for face images and other natural images and another dictionary solely comprised of face images. It also includes a deblurring kernel for VLR images and a down-sampling operator for improving reconstruction performance. This approach uses a Hidden Markov Model with 7 states, with each state modeling a facial component, and performs feature extraction and face recognition using SVD coefficients.

The CLLR-SR CCA [1] method uses synthesizes faces using CCA to transform the features extracted from VLR images to the HR feature subspace, closer to the ones with a stronger correlation in this target subspace. The 2D CCA approach proposed by Ann *et al.* [38] features an additional dimension, contrasting with regular single-dimension CCA. They model 2D CCA using 2D PCA projection vectors for directions X and Y. They model two of these 2D projection matrices for left-side and right-side projections. The authors loosely reported efficiency metrics, of 1.38 seconds per super-resolved image in a 2.4GHz x86 processor.

2.1.1.2.2 Deep Learning-based Super Resolution

Super resolution approaches using CNNs feature designs with a single branch, coupled training with a face recognition constraint, and Capsule Network Reconstructions. Single branch designs include the **Cascaded Super-Resolution with Identity Prior (C-SRIP)** [43] and **Compact Super Resolution Network (CompSupResNet)** [91]. More complex approaches with coupled FR training are the **Deep Cascaded Bi-Network** [150], **Single Network Super Resolution (SNSR)** [124], **Robust Partially Coupled Network** [124], **Complement Super Resolution and Identity Network (CSRI)** [22], and the **Feature Adaptation Network (FAN)** [137]. Capsule Network approaches for face recognition include the **Dual Directed Capsule Network (Dual Directed CapsNet)** [107].

The Complement Super-Resolution and Identity (CSRI)[22] method features two branches: one for handling the synthetic VLR images with their native HR image pair and another one for the native VLR images without a HR ground truth pair Figure 2.7 illustrates this approach. The two branches share parameters across from each other, for the Super Resolution subnetwork and the Face Recognition subnetwork. Using this shared strategy amounts to an improvement of 8% in the Rank-1 identification rate for the QMUL-TinyFace dataset. The authors also reported a 10.1% increase in the same identification metric when training the SR network with the propagated gradients from the classification output.

The Cascaded SRIP, shown in Figure 2.8, uses multiple SR modules based on ResNet blocks. The output of these blocks is measured against the ground truth for different scale

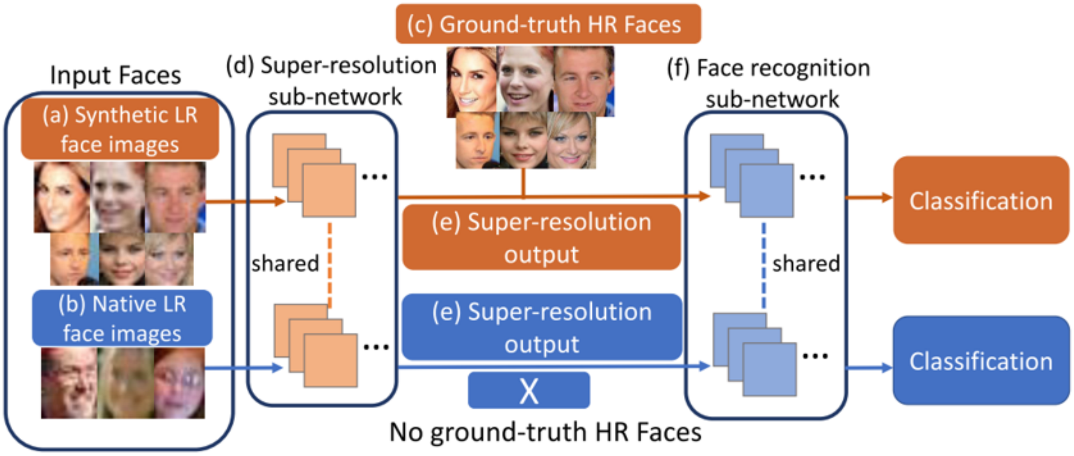

(c) Ground-truth HR Faces
Input Faces
(a) Synthetic LR face images
(b) Native LR face images
(d) Super-resolution sub-network
shared
(e) Super-resolution output
(e) Super-resolution output
X
No ground-truth HR Faces
(f) Face recognition sub-network
shared
Classification
Classification

Figure 2.7: Complement Super Resolution and Identity (CSRI) network architecture, taken from [22]. We can fully appreciate the nature of the training strategy by leveraging the information from the native dataset and face recognition gradient propagation.

factors (2×, 4×, 8×) at the end of each SR module. These outputs are used as input for multiple independent SqueezeNets, for each scaling factor, and propagate the gradient from the cross-entropy loss [16].

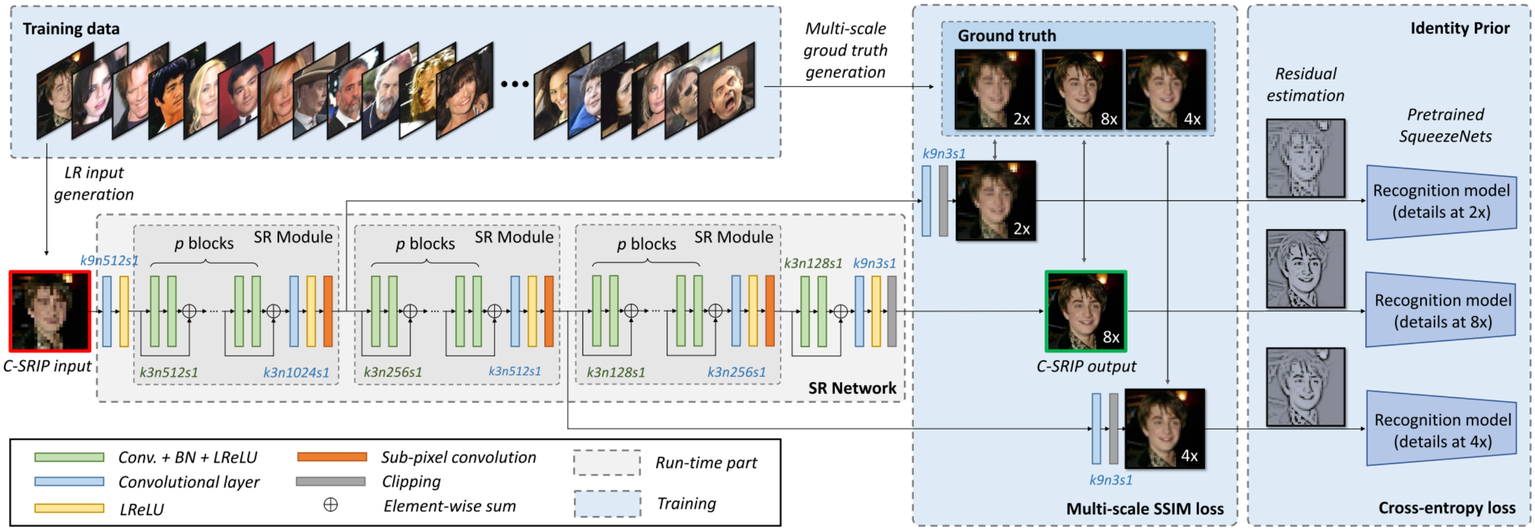


Figure 2.8: The Cascaded Super-Resolution with Identity Prior (C-SRIP) design, taken from [43]. This method sequentially upscales the input image by 2× from consecutive SR modules.

The Dual Directed Capsule Network for Very Low Resolution Image Recognition [107] utilizes native VLR images and upscales them using bilinear interpolation for training. They downscale the native HR images to the VLR domain using this same interpolation method. Figure 2.9 illustrates the components of this architecture. At the optimization level, the authors propose to utilize the features extracted from the native HR images as anchors in the feature space. With their proposed HR anchor loss function, they promote a closer distance of the VLR features and HR features before introducing them to the Capsule modules. The back-propagation step has the constraints of the classification capsule output, reconstruction module, and HR anchors. In this design, VLR information is not used in the learning process of the anchor, only HR information. This approach achieved a Rank-1 face recognition rate of 95.81% in the UCCS dataset.

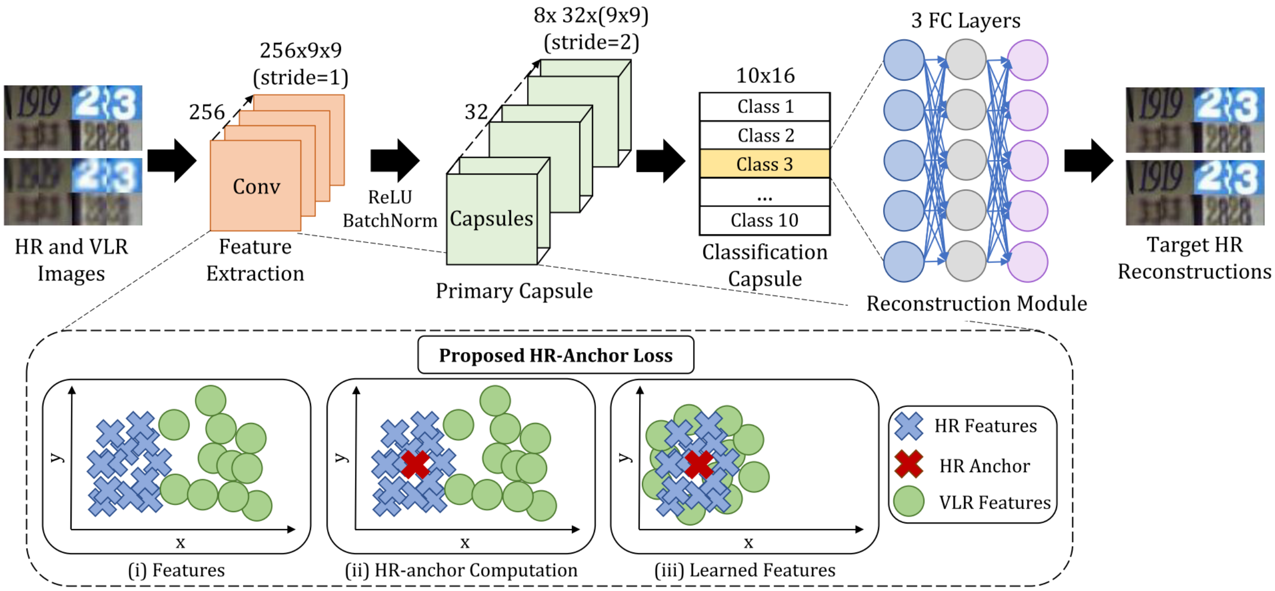


Figure 2.9: Dual Directed Capsule Network Architecture[107]. This method uses the capsule dynamic routing procedure for classification, FC layers for SR reconstruction, and uses the HR anchor loss for projecting HR and VLR features for classification.

The Feature Adaptation Network (FAN) [137] approach, depicted in Figure 2.10, proposes to learn facial encodings for synthetic VLR and native HR images separately, in a disentangled manner. In this method, the facial encodings from different subnetworks are concatenated and used as input for the decoder module performing Super Resolution. This GAN-based approach proposes to also encode non-identity features. The separation of non-identity features allows to disregard them at the loss-function level, leaving the identity features to be used for SR and FR purposes. They reported an inference time of 16ms per image on an NVIDIA Titan X GPU and a 90.27% of mean accuracy in the SCface dataset.

The Compact Super Resolution Network (ComSupResNet) is a more efficient proposal for SR. This single branch architecture is depicted in Figure 2.11. The main VLR feature extraction is performed at the first two convolutional layers, using kernels of size 9×9 and 1×1, respectively. In this design, they maintain a 64-depth channel feature map dimension throughout. Afterwards, they perform consecutive upsampling steps, each time for a factor of 2×. Their main processing blocks are ResNet-styled [48]. The authors reported a close SSIM and PSNR performance on for the CelebA and LFW datasets compared to C-SRIP and LapSRN, using only 1M parameters, against 30M for C-SRIP.

A couple of SR methods based on Lightweight CNNs and Knowledge Distillation approaches are present in [142, 37]. Although they are not proposed for uncontrolled face recognition scenarios. We discuss Lightweight CNNs at length in Paragraph 2.1.2.2.2 from the

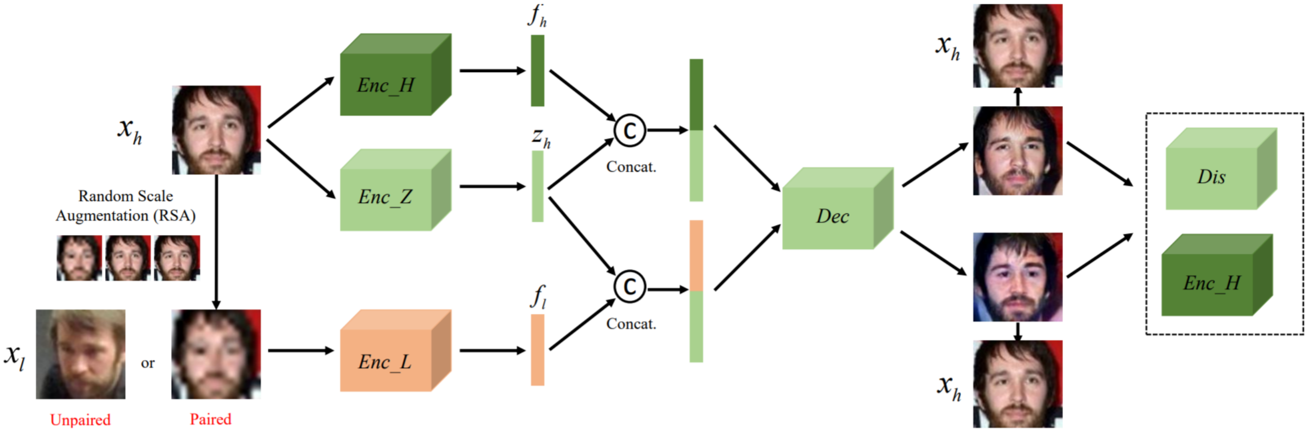


Figure 2.10: The Feature Adaptation Network (FAN) architecture, taken from [137]. In this design, the it uses an encoder for native images and another for augmented images. They are both trained using a decoder for identity features and distractors.

following subsection. We also discuss quantization approaches for SR in Section 2.1.4.3.

Super Resolution approaches are a viable approach when inference runtime is not a major concern. Their approaches are often more expensive to run than CM methods and Lightweight CNN designs. The upsampling process prior to classification using CNNs is especially heavy on computational requirements, with many designs relying on multiple-branch architectures. GAN-based approaches are also unfeasible to run in real-time, as evidenced in FAN [137], needing a not-easily-affordable NVIDIA Titan X GPU for real-time performance. The Capsule Networks [52, 98] are novel approaches sharing many design ideas from Lightweight CNNs for FR, such as avoiding max pooling but instead focusing on model activity vectors. For any given class, it yields a vector of values that later can be used as a representation for classification. However, the dynamic routing procedure is also prohibitively expensive for an embedding platform.

The advantages and limitations of the Super Resolution approaches are:

Advantages

- The possibility of leveraging both synthetic and native datasets at the same time without their corresponding pair at the training stage shows potential to improve cross-resolution robustness.
- Adding the face recognition constraint ensures that the new super-resolved face images are beneficial for learning features for more accurate face recognition performance.
- Super resolution methods allow for a qualitative evaluation of the faces, which could help on explaining the results of a face recognition model, something that is not possible with coupled mapping approaches.

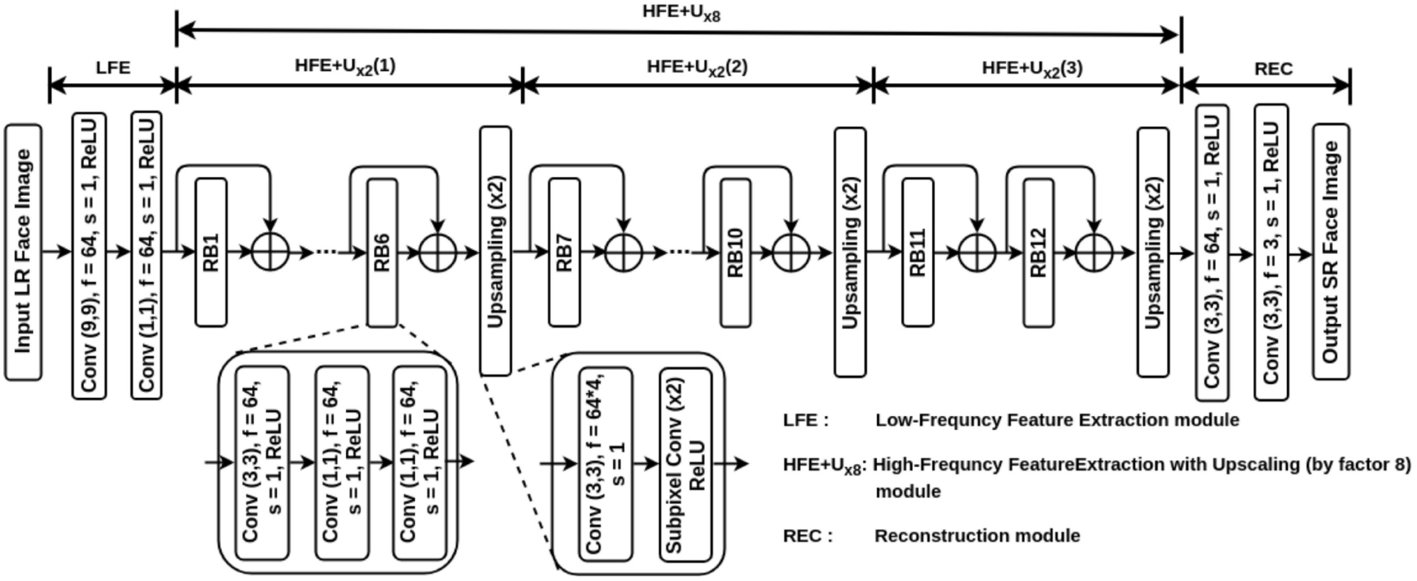

HFE+U$_{x8}$
LFE
HFE+U$_{x2}$(1)
HFE+U$_{x2}$(2)
HFE+U$_{x2}$(3)
REC
Input LR Face Image
Conv (9,9), f = 64, s = 1, ReLU
Conv (1,1), f = 64, s = 1, ReLU
RB1
RB6
Upsampling (x2)
RB7
RB10
Upsampling (x2)
RB11
RB12
Upsampling (x2)
Conv (3,3), f = 64, s = 1, ReLU
Conv (3,3), f = 3, s = 1, ReLU
Output SR Face Image
Conv (3,3), f = 64, s = 1, ReLU
Conv (1,1), f = 64, s = 1, ReLU
Conv (1,1), f = 64, s = 1, ReLU
Conv (3,3), f = 64*4, s = 1
Subpixel Conv (x2) ReLU
LFE : Low-Frequncy Feature Extraction module
HFE+U$_{x8}$: High-Frequncy FeatureExtraction with Upscaling (by factor 8) module
REC : Reconstruction module

Figure 2.11: The Compact Super Resolution (ComSupResNet), taken from [91]. This method uses two convolutional blocks for VLR feature extraction and residual blocks for posterior filtering, with an upsampling step at the end. This upsampling step is repeated for the next 2× scaling factors.

Limitations

- Although Super Resolution is a very active research area, the deep learning-based approaches are not efficient for real-time inference performance. In general, these approaches are very expensive to train and perform inference on, even when using very powerful GPUs.
- Supervised training of Super Resolution networks requires a synthetic pair to be able to train, thus adding elements that are not representative of the real-world problem.
- Performing Super Resolution and recognition afterwards, without a recognition constraint at training, most of the time will hinder performance [22]. This is due to the fact that HR FR networks are designed to extract features from a richer resolution space. Deeper networks usually degrade the quality of the already scarce feature space VLR face images.

2.1.2 Homogeneous Face Recognition

In this subsection, we synthesize and discuss the methods pertaining Homogeneous approaches for Face Recognition. As previously mentioned, this approach type processes all the input data in the same manner, regardless of the source domain (VLR or HR, in this case).

This approach type includes Face Recognition methods primarily designed for high resolution, in both constrained and unconstrained settings, for images of a region of interest of 32×32 or more. Consistent with previous work [75], we also divide these methods into Traditional methods Face Recognition methods, and Deep Learning-based methods.

2.1.2.1 Traditional methods for Face Recognition

Classic or traditional methods for Face Recognition include those based on Holistic representations and Hand-engineered feature-based matching. Some proposals for Holistic representation are **Eigenfaces** [116], **Fisherfaces** [7], and **Laplacianfaces** [87]. Methods for Hand-engineered feature-based matching include using **Gabor features**[24, 118, 102] and **Local Binary Pattern Histograms** [34].

Holistic methods encode the whole face in a mathematical representation for face representation and matching. Eigenfaces performs the projection solving the eigenvectors to reconstruct the dataset information. Fisherfaces uses LDA-based techniques for modeling the scatter matrices and factors them into the eigenvector problem to solve. LaplacianFaces relies on the Locality Preserving Projection (LPP) to model the eigenvector problem as a Laplacian Eigenmap. This involves constructing a graph with weighted edges to optimize the model and match the projected face data.

Hand-engineered feature-based FR involves extracting features from face images using human-designed filters and histograms, such as the case of SIFT [73] features and Gabor filters [24]. These approaches are based on extracting the orientated gradients of an image. Then, these image descriptors are matched using PCA [118] and PCA+LDA [102]. In the case of Local Binary Pattern Histograms, the image descriptor is extracted using the pixel value as the threshold and encoding whether the surrounding pixel values are above the threshold or not. This binarized response is encoded in a single decimal number and the histogram is built from this representation.

2.1.2.2 Deep Learning-based Face Recognition

This approach type includes modern Deep Learning-based methods for face recognition, such as **DeepFace** [111], **VGGFace** [88], **FaceNet** [101], **DeepID** [108], and the **Trunk-branch Ensemble Network** [33]. Using Convolutional Neural Networks for Face Recognition greatly improved recognition performance over traditional approaches, allowing for extracting and building more robust descriptors. However, we make the distinction between heavier methods not optimized for runtime inference efficiency and Lightweight CNN approaches for optimizing efficiency performance. Lightweight CNNs for Face Recognition include **MobileFaceNet** [20], **ShuffleFaceNet** [79], **VarGFaceNet** [132], and **MixFaceNets** [13]. Deep-Learning-based methods also involve research for newer loss function methods, such as **Cross-Entropy loss** [16], **CosFace** [122], **CenterLoss** [128], **Triplet loss** [19], and **ArcFace** [31].

2.1.2.2.1 Regular CNNs for FR

DeepFace [111] by Facebook, primarily uses a 3D template to perform face frontalization, aligning the template with the detected face fiducial points. Posteriorly, the authors used a 9-layer CNN to project the extracted 2D image face features. This design is illustrated in Figure 2.12. They trained this method using their own large-scale dataset, not available elsewhere, achieving a 97.53% accuracy on the LFW dataset. This method used the Cross-Entropy loss function for optimization with a stochastic gradient descent (SGD) optimizer.

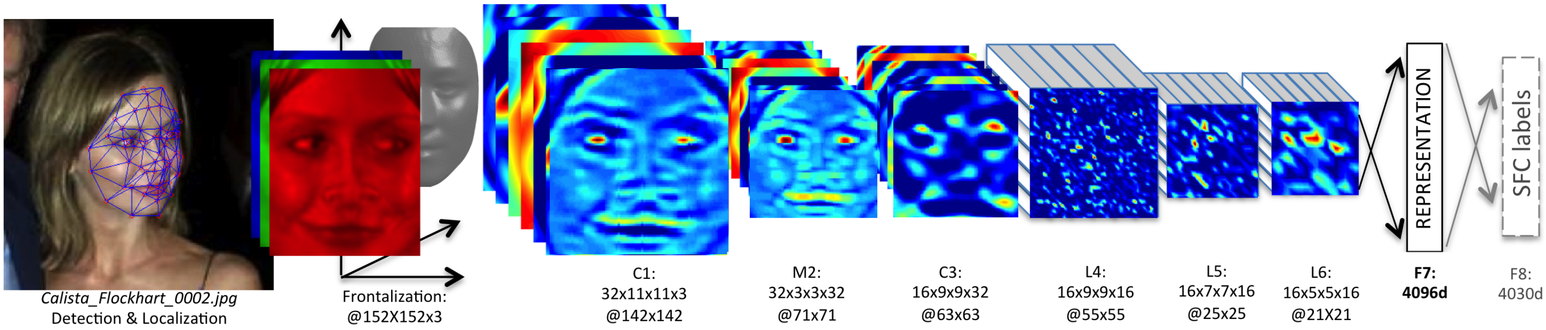


Figure 2.12: DeepFace architecture, taken from [111]. This approach uses 3D face template matching for normalization, and a CNN with a 4096-D embedding for classification.

The VGGFace architecture [88], depicted in Figure 2.13, is designed as a 20-layer architecture for feature extraction. They rely on regular convolutional layers, max pooling, and fully connected layers. In this design, the authors propose to progressively increase the depth channels of the intermediate representations, up to 512 channels. Their face embedding is a 4096-D face descriptor. They use the Softmax loss functions and they use the triplet loss for learning task-specific embeddings.

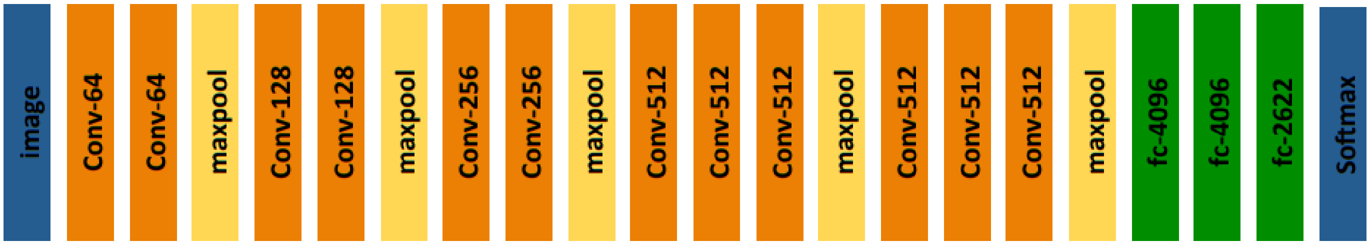


Figure 2.13: VGGFace architecture, taken from [88]. This network expands the representation every two convolutional layers, after the max pooling operation, then flattens the representation to a 4096-D face descriptor.

In FaceNet [101], the authors discuss different layer types for face recognition. Their best performing model is based on Inception [110]. They propose to expand the channels of the intermediate representation to 1024 dimensions, with an L2 normalization layer as the final output. They also evaluated various face embedding dimension sizes and concluded that a 128-D descriptor performs most optimally in accuracy performance. Their total model complexity is 7.5M parameters and 1.6 billion FLOPS.

The DeepID [108] approach uses multiple small ConvNets for different patches in the face image. The number of patches is variable to any k number. The base design for these smaller ConvNets is illustrated in Figure 2.14. The output of these multiple branches is combined in a 160-D descriptor, posteriorly used for identification or verification. They especially note that the output for the last max pooling layer and their last convolutional layer must be directly present on the DeepID descriptor, to avoid a bottleneck in the information propagation. The authors reported an accuracy of 97.25% in the LFW dataset.

The Trunk-Branch Ensemble Network for Face Recognition [33] shares low and middle-level detail layers of two different sub-networks and has one large trunk network that learns

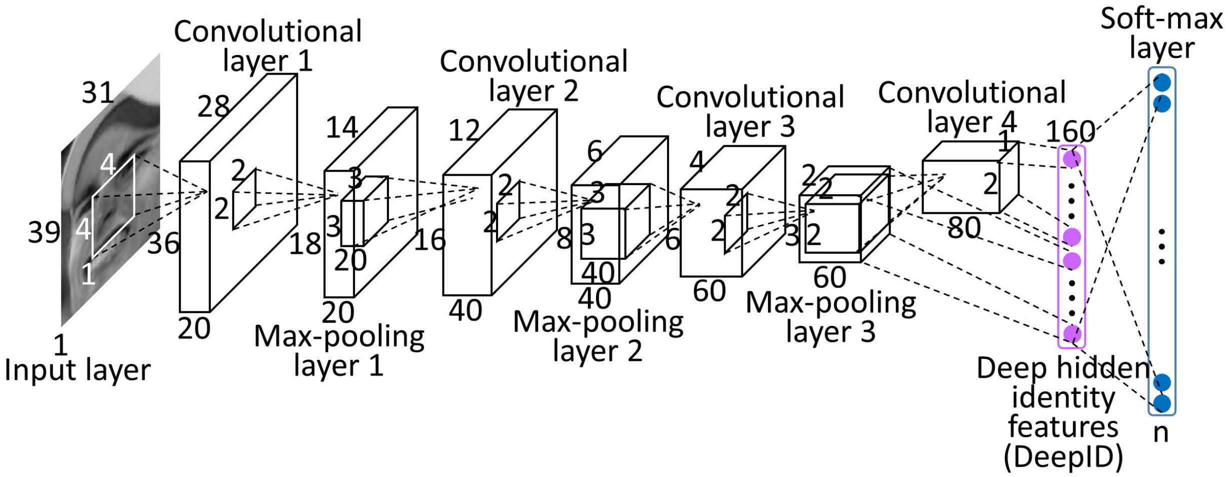


Figure 2.14: The Deep-ID architecture, taken from [108], aims to for a 160-D face descriptor with every component extracted from an image patch processed by an independent CNN.

representations for holistic images. The two branch networks are used to extract features from facial image patches, where the resulting feature vectors are concatenated at the end with the one from the larger trunk network. This architecture is shown in Figure 2.15. They also use blurring as a data augmentation technique for training. Adding the simplified, lighter branch networks do not constitute in a heavy run-time overhead, where most of the benefit comes from having the shared weights and focusing on localized features.

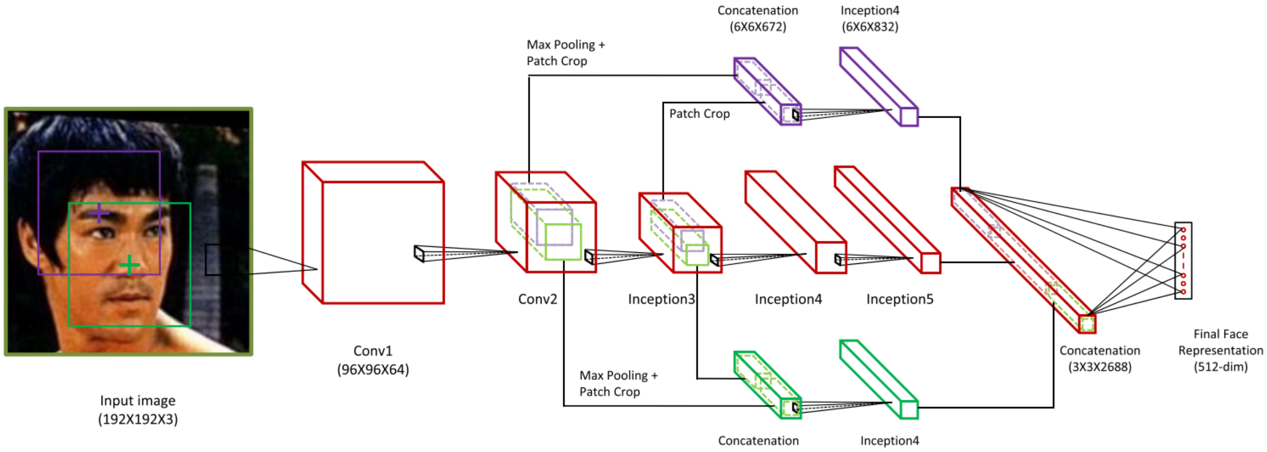


Figure 2.15: Architecture of the Trunk-Branch Ensemble network, taken from [33]. We note that the smaller branch networks learn different local features from patches in distinct locations.

2.1.2.2.2 Lightweight CNNs for FR

Recently, the need of using CNNs for mobile and embedded-oriented real-time scenarios [2] has emerged. Seminal lightweight methods such as SqueezeNet [57], ShuffleNets [147, 76], MobileNets [55, 99, 54], VarGNet [146], EfficientNet [113], and MixConvs [112] have been proposed for efficiently solving general Computer Vision tasks such as image recognition, object detection, and others. Figure 2.16 shows the accuracy-to-efficiency trade-off for general-purpose Lightweight CNN approaches. Face recognition variants of these methods have been proposed as well, such as **MobileFaceNet** [20], **ShuffleFaceNet** [79], **VarGFaceNet** [132], and **MixFaceNets** [13]. These lightweight approaches have a memory footprint of less than 20Mb and around 1GFLOPS of computational complexity [32].

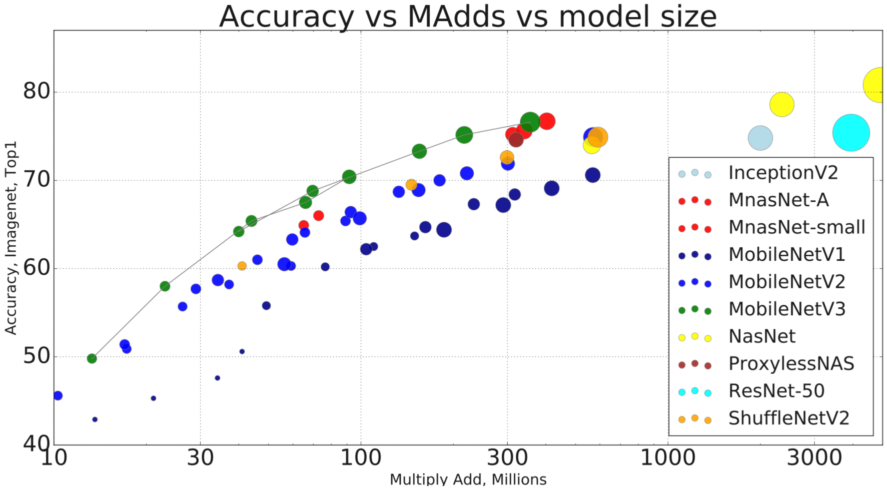


Figure 2.16: Accuracy and efficiency trade-off of various general purpose state-of-the-art lightweight convolutional neural networks, taken from [54].

These techniques follow a series of guidelines to improve efficiency without sacrificing accuracy: such as variable grouped convolutions [132], random channel shuffling [147], bottleneck structures [20], pointwise 1x1 convolutions, small embedding size [79], and favor strided convolutions instead of pooling layers. Strided convolutions reduce complexity by reducing the feature map size and retaining more information directly from the data, while the inverted bottleneck structures first reduce the number of parameters and then compact the network channels again to match the input channels. The PReLU [115] activation is also favored instead of the classic activation function ReLU. All of these improvements show accuracy, and sometimes efficiency, improvements in face recognition performance when compared against the original general-purpose approaches [81].

MobileFaceNet [20], based on MobileNetv2 [99], features a Squeeze-And-Excitation block in its head setting and bottleneck structures, replaces the Global Average Pooling (GAP) layer with a Global DepthWise Convolution (GDC) in the embedding block, a 128-D embedding output, and using PReLU. When fine-tuned to the SCface dataset, this method achieves a remarkable 95.3% of recognition rate at distance d1 (4.2m), the hardest scenario for this dataset, with an inference time performance of 62.45ms per image using an Intel i7 7700HQ laptop CPU [75].

The ShuffleFaceNet [79] approach, based on ShuffleNetV2 [76], features a 2-stride fast downsampling strategy at the first convolutional layer, also replaces the GAP layer with a GDC layer, adopts PReLU *in lieu* of ReLU, and reduces the 1024 feature representation to a compact 128 embedding face embedding after the GDC layer. The intermediate representations are smaller after every stage and gradually increment the channel size, with a 7×7×1024 before the GDC layer. This network achieves a runtime performance of 29.08ms on an Intel i7 7700HQ laptop CPU and an 86% recognition rate on the hardest scenario (distance d1) of the SCface dataset [75].

The VarGFaceNet [132] method, inspired by VarGNet [146], mainly sets a fixed channel number and has a variable number of groups in the variable group convolutional layer. Furthermore, it adopts PReLU, adds a Squeeze-And-Excitation block in their normal block adds a pointwise convolutional layer before the Fully Connected layers, and defines a 512-D embedding as output. In Figure 2.17 their training methodology for the best results at the LFR@ICCV 2019 challenge [32] is described, with a Teacher-Student design. Their teacher network is a ResNet100 (24GFLOP) design. This training methodology is also commonly known as Knowledge Distillation. Even though it outperforms MobileFaceNet in terms of accuracy performance in the LFR challenge verification dataset, this approach (student network only) is more expensive at 1.02GFLOPS, which corresponds to an inference time of 70.40ms per image in an Intel i7 7700HQ laptop processor.

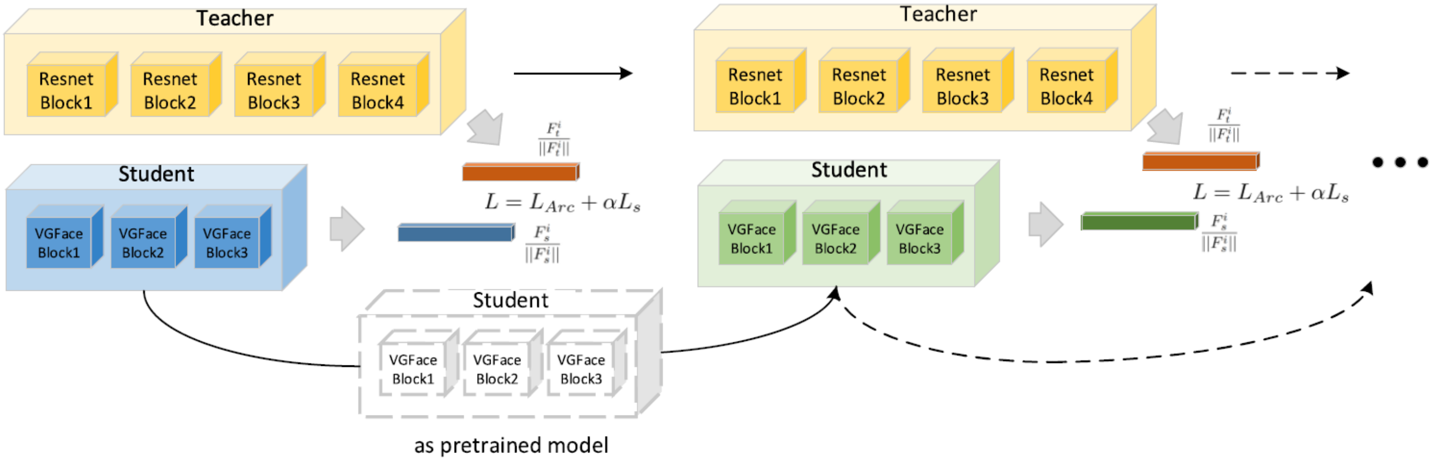


Figure 2.17: VarGFaceNet architecture, taken from [132]. The authors employ a more complex training methodology to achieve the best results, with a teacher-student design. The student network, derived from VarGNet, features improvements specific to FR performance.

In MixFaceNets [13], the authors draw inspiration from MixConvs [112]. This architecture is shown in Figure 2.18. They apply the 2-stride fast downsampling strategy at the first convolutional layer, batch normalization, and adopt the PReLU activation. They did not downsample the representation after their head block to preserve information, instead, they adopt a downsampling MixConv block with kernels of 3×3, 5×5, and 7×7, with a channel shuffle operation at the end. They adopt a 512-D embedding size for their descriptor. The authors report competitive results on AgeDB30, MegaFace, and IJB-C, while reducing complexity with their lightest proposed configuration.

Knowledge Distillation training techniques [51] can leverage the faster inference runtime of Lightweight CNNs with the distilled knowledge from the heavier teacher networks,

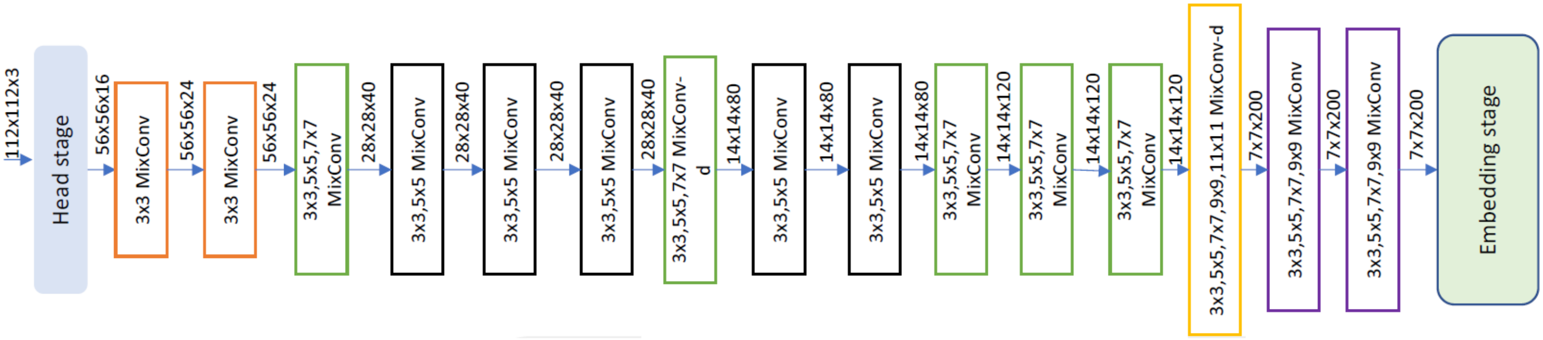


Figure 2.18: MixFaceNet Architecture, taken from [13]. This design features MixConvs with different kernel sizes, channel shuffle operations, and low-dimensional intermediate feature representations. Their descriptor is a 512-D embedding.

as illustrated by VarGFaceNet. However, this is only most convenient in a final deployment application, because it does not test the discriminative ability of the student network-per-se.

Other approaches to reduce model complexity are Quantized and Binarized Neural Networks (BNNs) [134, 94, 58]. These approaches mostly focus on reducing parameter number representation and overcoming the challenges inherent to limited representation learning. We discuss these approaches at length in Subsection 2.1.4.

The main advantages and limitations of Homogeneous FR approaches are:

Advantages

- They are focused on generating more robust and discriminative face descriptors, which can aid in cross-domain performance and have been more accurate in verification performance.
- Deep Learning Homogeneous FR approaches are highly optimized for GPU-enabled environments.
- The block-based network design approach is very flexible for balancing efficiency and accuracy performance.

Limitations

- There is no explicit way to bridge specific cross-domain gaps. As such, dataset preprocessing can impact performance in a larger way (e.g. interpolation method selection and input layer size).
- CNN-based Homogeneous FR approaches are prohibitively expensive to run in very limited embedded device scenarios, where even the lightest Lightweight CNNs would struggle to run in real-time.
- Most proposed approaches are designed with HR native face images in mind, even if there are domain gaps of age or unconstrained scenarios. This makes homogeneous approaches specific to VLR FR an area of opportunity.

2.1.3 Deblurring

Deblurring techniques applied to VLR images are also divided into classical and deep learning methods. Classic methods try to estimate a deblurring kernel and optimize a reconstruction loss function. The types of deblurring kernels being learned are local kernels and global kernels. Global deblurring attempts to estimate kernels for the whole image, which makes the process more challenging as opposed to local deblurring, which focuses only on parts of the image. Classical deblurring approaches include [114],[66], and [86]. However, Deep Learning approaches have been more successful lately. One of the most successful GAN-based deblurring techniques which use a different approach is Deep Semantic Face Deblurring [103]. In this work, the authors combine local and global constraints, which they call semantic priors, to achieve better results. The local constraint focus on gathering information from critical areas such as eyes, nose, and lips and they do not focus on estimating deblurring kernels, as opposed to the rest of the deblurring literature. The authors report a 20 FPS inference performance on an NVidia Titan X GPU, with a recognition accuracy increase of at least 12 % over the state of the art. However, its failure cases involve non-aligned faces, most likely because of the semantic prior mask that is aligned with the critical areas of the face in a semi-frontal fashion.

We figure that we can take a similar approach by estimating a semantic prior to taking into consideration the face orientation, not necessarily using a face normalization technique. GAN-based techniques are very popular in the state of the art for solving many Computer Vision tasks, however, they are also very expensive to train and to run inference on, as is evident in this case as well. However, it is also worth noting that decoupled deblurring methods do positively affect recognition performance as opposed to the decoupled Super Resolution and classification methods mentioned in the previous section.

Similarly, the advantages and limitations include:

Advantages

- Classic approaches often include a single kernel for deblurring at inference time, making them useful as an augmentation strategy.
- Deep Learning approaches include newer GAN-based methods which can yield very good results, using semantic priors.
- Local constraints for deblurring particular facial components are intuitive approaches for face recognition.

Limitations

- Classic deblurring techniques struggle with the local deblurring process. This is particularly a problem for large-scale native VLR scenarios where the native feature space is very small.
- In the same vein as Super Resolution, using deblurring techniques does not guarantee better recognition performance.
- GAN-based approaches are extremely expensive for a real-time application on affordable hardware, as with Super Resolution GAN-based approaches.

2.1.4 Quantized Neural Networks

Regular Convolutional Neural Networks use 32-bit numerical representations for the floating-point parameters. This makes convolutions a series of multiplications and additions, which get accumulated in millions of floating operations per second (Mega FLOPs). However, due to the increasing need for more efficient neural networks, approaches using lower-bit and binary representations for convolutional layers and parameters have been proposed. These approaches are referred to as Quantization techniques; 1-bit representations are particularly known as Binarization techniques.

2.1.4.1 Binarization in Neural Networks

Binarized Neural Networks (BNNs) use a single bit for the elements of the weight vectors and the result after applying activation functions. The gradient update for back-propagation is done with 32-bit floating point representations. In particular, binary representations and operations optimize efficiency in a theoretical factor of $32\times$. This makes convolutions a series of more efficient bit-shift and XNOR operations. However, the most important challenge is that the image signal and high-frequency details are heavily degraded. The first works on Binary Neural Networks (BNNs) are **BinaryNet** [26] and **BinaryConnect** [27]. However, most of the classic binarization literature is based around **XNORNET** [94], in terms of quantization and training strategies. More modern approaches include **GroupNet** [151], **ABCNet** [69], **BinaryDenseNet** [8], and **QuickNet** [4].

In XNORNET [94], proposed by Rastergari *et al.*, the authors expand over BinaryNet and BinaryConnect by adding two independent scaling terms to aid in the binarization process. Furthermore, they propose XNOR convolutions for fully binarizing CNNs. As such, weights and input convolutions can be computed using XNOR and bit-counting operations. Binary convolutions consist on repeating shift operations and dot product. The weight filter is shifted over the input and performs a dot product operation with the input and the weights. The binary dot product real-value weight estimation is computed by:

$$X^T W \approx \beta H^T \alpha * B \tag{2.1}$$

Where H, B are binary and α and β are real-valued scalars. These scalar terms are very important, as they linearly expand the range of the intermediate representations. This makes further quantization steps more precise and avoids incorrectly converging to zero. The optimization of these parameters can be formulated as: $X \odot W - \beta\alpha H \odot B$
Where the goal is to model the estimation using the sign function as such:

$$C* = sign(Y) = sign(X) \odot sign(W) = H * \odot B* \tag{2.2}$$

This formulation also allows for easy k-bit quantization, by replacing the sign function with:

$$q_k(x) = 2(\frac{\text{round}((2^k-1)(\frac{x+1}{2}))}{2^k-1} - \frac{1}{2}) \tag{2.3}$$

The authors recommend avoiding pooling layers on binary input, as it results in a significant loss of information, and normalizing the input before binarization. They also avoid quantizing the first and last layers of the network. Their proposed XNORNET block is comprised of a BatchNorm layer → BinaryActivation → BinaryConv → Pooling layers.

The theoretical speedup of using binarization techniques is $S = \frac{64CK}{(CK+64)}$, where C is the channel size and K is the feature size. The real-world speedup is of around 58% for one convolution [94].

2.1.4.2 Low-bit representation networks

Low-bit representation networks are able to dynamically downscale the number representation to lower than 32 bits, but not necessarily to binary representations. This provides a balance between preserving signals using real-valued representations and signal degradation. We discuss DOREFA-Net [149], ABC-Net [69], and Group-Net [151]. Other quantization approaches also include [148, 35]

In DOREFA-Net [149], the authors propose to deterministically quantize the Weights and activations, while quantizing the gradients in a stochastic fashion, using the Bernoulli distribution. This gradient stochastic quantization, named straight-through estimator (STE), was introduced by Hinton *et al.* in [51]. This approximation is computed due to the Bernoulli distribution not being mathematically differentiable However, it is worth noting that the gradient quantization only has a positive effect at training time because the gradients are not quantized at the back-propagation step.

The Accurate Binary Convolutional Neural Network (ABC-Net) [69] method defines ApproxConv Blocks. These blocks, shown in Figure 2.19, have a fixed number of binary convolutions expanded channel-wise. The output of each one of these convolutions is scaled with its own α parameter and then aggregated for output. These blocks are applied in the same channel-wise fashion. Every input activation is binarized with its own activation function and every output is scaled by its own β parameter.

This approach is a flexible and scalable design for building binary networks, more akin to Lightweight CNNs. When using 5 ApproxConv blocks, a full-precision ResNet50 network is approximated with a 6% of difference in Top-1 accuracy. By allowing to have different binarization activation functions, the limitation of using a single criterion, such as the sign function alone, is circumvented. The convolved output is then scaled more appropriately to fit the data.

The Structured Approach for binarization, named Group-Net [151], aims to better approximate residual blocks by using a conjunction of binary blocks. However, this process is not as trivial as concatenating the output of two binary convolutions. Sequentially, the block gets broken down into parallel binary blocks and they are branched out in the last step. They also propose fusion gates to aggregate residual serialized binary convolutions. The goal of these fusion gates is to have multiple branches for features to be extracted, effectively improving the convergence and robustness of the network.

The model also leverages multi-scale information by decomposing the convolutional operators into branches with different filter sizes. These convolutions are called Binary Parallel

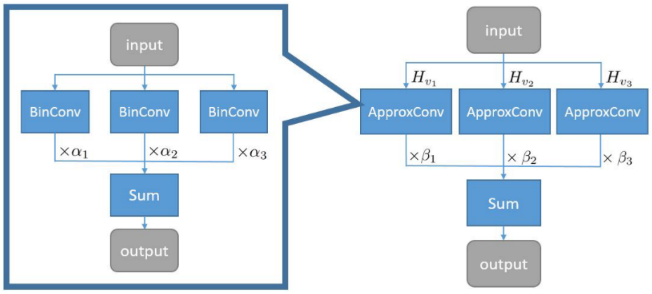


Figure 2.19: ABC-Net block structure, taken from [69]. This includes ApproxConv blocks, each one with its own learnable scaling factors. At the same time, the output of the ApproxConv is re-scaled with its own parameter.

Atrous Convolution (BPAC). They are an approximation of floating-point dilated convolutions, this binarized version applies dilation sequentially and aggregated the generated maps for its output. This structural approach for achieves better performance for approximating ResNet50 networks and units, but at the expense of K times more storage and complexity, due to the sequential binary convolution approach.

This structured binary approach becomes less computationally expensive to approximate real-value tensors. When other binary models, such as ABC-Net [69], require K weight bases and K activation bases, this approach only requires K binary convolutions which are approximated by the scalable approach of the grouped convolutions with dynamic bases, BPAC and fusion gates.

In the Knowledge Distillation for binarization [78] approach, the authors propose to match the spatial attention maps at the loss function level. The real-value network attention maps are matched with the ones computed from a binary network. They introduce a real-to-binary convolution approach for computing the scalar factors α as a data-driven channel re-scaling approach. This approach completely avoids analytic scale factors as in XNORNET and enforces this group's sequential constraints in the loss function level. Furthermore, the combination of linear function activations gives the scaling term more non-linearity at the expense of very little extra computations.

The authors also define a multiple-stage training for training using the binarized Knowledge Distillation method. At the first stage, the student network is real-valued and the $\tanh$ function is added to the activations in order to softly start binarizing the network. In the second stage, the activations are binarized and training is done using the network resulting from the previous stage as the teacher network. At the last training stage, the weights are also binarized and the network from the previous stage becomes the teacher network.

The Binary DenseNet approach [8] avoids Knowledge Distillation and other complicated training methodologies. Instead, it mainly focuses on proposing and applying design principles relevant to BNNs such as: retaining a rich flow of information down the network, eliminating bottleneck designs, and avoiding directly quantizing the same architectures as full-precision designs, since the challenges are inherently different. They propose to modify the DenseNet bottleneck block by replacing a convolutional with two layers of half of the original filter number in each one. They also increased the reduction rate in the DenseNet transition block to have a lighter depth-channel representation down the network. They employ the ReLU activation in full precision.

In QuickNet [4], the authors propose to use Quantized DepthWise Convolutions, ReLU activations, Global Average Polling, a 512-D face descriptor, and novel block designs. Firstly, they downscale the representation to half of the spatial shape and expand the channel dimension to 16 channels in their head setting. After this, they expand the representation to 64 channels and progressively upscale the depth channel up to 512 channels by the end of the network. Their transition block is used to downscale the representation by half, spatially. It features a max pooling block with a Quantized DepthWise Convolution to downsample the representation, before finalizing with a regular Quantized Convolutional layer. They reported an improved efficiency performance in latency (ms) measurements over BinaryDenseNet and Real-to-binary [78].

2.1.4.3 Quantized Super Resolution

Binarized Neural Networks is still an emerging area. Some proposals for binarizing Super Resolution networks have been proposed, such as the Efficient SR Using BNNs by **Yinglan *et al.*** [77] and **Binarized Single Image SR** [131]. These approaches rely on other architectures

The Efficient SR Using BNNs by Yinglan *et al.* [77] approach aims to binarize certain layers in popular SR approaches: SRGAN [64], LapSRN [63], and SRResNet [64]. They propose to use the sign function and gradient clipping at range [-5,5]. However, the scaling alpha parameter is learned and optimized for approximating W as a parameter in the network. This type of approach also needs greater learning rates (3-4×) due that it requiring a larger momentum to effectively change the sign function output. They binarize all convolutional layers except for the SR modules and discriminators for SRGAN. Empirically, 16 residual blocks perform the best in terms of Super Resolution metrics. After 16 blocks, it starts to get diminishing returns. However, it still compromises performance in high-frequency details. Figure 2.20 shows the binarization approach for SR ResNet. They choose to binarize only the convolutional layers of the feature extraction module and leave the first convolutional layer and the reconstruction module intact.

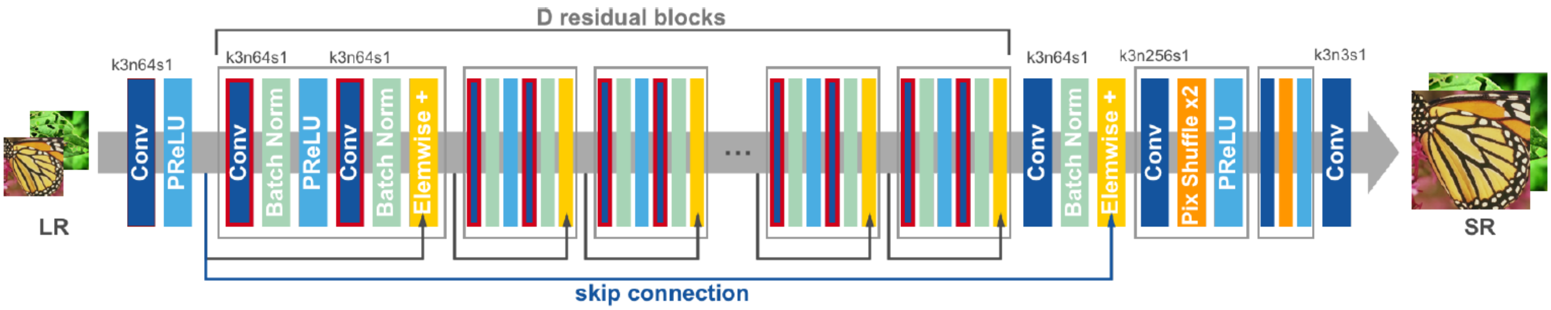


Figure 2.20: Binarization on SRResNet, taken from [77]. The convolutional layers, marked in red, are binarized using learnable alpha parameters.

In the Binarized Single Image Super Resolution approach [131], the authors propose to binarize the weights using a vector of scale factors α and update the binary filter with the mean absolute value of each weight output channel. This is described as $W_n^B = sign(BN(\alpha_1 W_1 + \alpha_2 W_2 + ... + \alpha_n W_n)) * E(|W_n|)$. In the same fashion, the authors propose to update the β scaling activation vector using an activation statistic $E(A_n)$ along the channel dimension. For the Super Resolution network, they propose the Bit Accumulation Mechanism (BAM) for the nonlinear mapping of the feature maps, featuring skip alternating residuals. Another remark is that this implementation is an isometric network, meaning that the width and height of the filters remain unchanged until the image reconstruction module, where the final output is upscaled. The usage of this binarization method in an SRResNet better approximates a Full-precision SRResNet for the Super Resolution task, leaving less than 1 point gap for PSNR in the DIV2K dataset. The authors report results outperforming BinaryNet [26], DOREFA-Net [149], and ABC-Net [69]. Furthermore, this binarization method reduces the parameters of SRResNet by 21.3% and the multiply-add count by 33.5%.

2.1.4.4 Quantized Face Recognition

In the same fashion, as Super Resolution approaches, the state-of-the-art on Quantized FR is based on binarizing some layers for face recognition networks and using Knowledge Distillation as training methodology. In **QuantFace** [14], the authors mainly studied 6-bit and 8-bit quantization for ResNet [48], and MobileFaceNet [20]. This approach is illustrated in Figure 2.21. They proposed to fine-tune the quantized version of the model using GAN-generated synthetic data from sampled noise from a Gaussian distribution $Z \sim N(0, 1)$. For their quantization scheme, they follow the scheme in [58]. The goal of the synthetic data is to expand the embeds produced by the Full-precision model for the Quantized model to learn from, using the Knowledge Distillation methodology. The quantization is defined as $Q(x, s, z) = \text{round}(\frac{x}{s} - z)$, where the scaling factor s is defined as $s = \frac{\alpha - \beta}{2^k - 1}$ and the zero point z as $z = round(\beta . \frac{2^k - 1}{\alpha - \beta} + 2^{k-1})$. The α and β values, in this case, are the highest and highest possible values for x in the Floating Point 32-bit representation space, $x \in [\beta, \alpha]$. This quantized value is then clipped to $[-2^{k-1}, 2^{k-1} - 1]$. The authors reported the most competitive verification accuracy results with the weights and activations at 8-bit precision representation against the 32-bit Full-precision representation, however, they did not report efficiency benchmarks.

The main advantages and limitations of Quantized and Binarized approaches are:

Advantages

- Very efficient convolutional computations, with the most direct efficiency improvements at binary representations.
- Theoretical speedup is up to 32×, with empirical performance of a 58% speedup in some cases [94].
- The back-propagation step is still a floating-point process, with the core speedup focused on inference runtime performance.
- Can take advantage of Knowledge Distillation training techniques when deploying these extremely light networks for specific real-world applications.

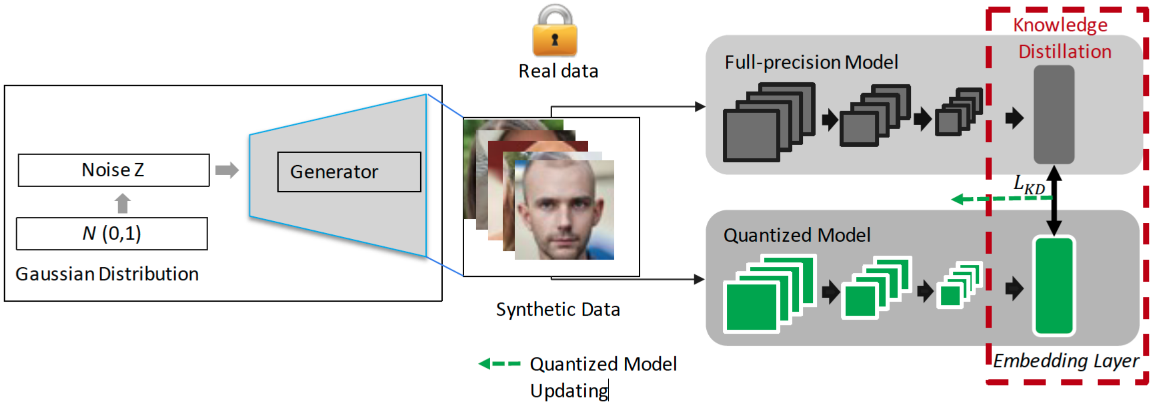


Figure 2.21: QuantFace training approach, taken from [14], trains a quantized version of a CNN, with as low as 8-bit weights and 6-bit activations, from its floating-point version as the teacher network. It also features data augmentation using Gaussian noise.

Limitations

- Real-value approximation error is very high. The loss of information from quantization and degradation from network depth directly affects accuracy performance.
- The sign function for binarization does not take into account the data distribution.
- Most of the focus of the literature is based on quantizing a determined network architecture or layers only. There is an area of opportunity for novel problem-specific architectures with the implications of lower-bit representations in mind.
- Current binarized designs do not achieve VLR FR on real-time performance on extremely hardware-restricted scenarios, such as a single core in an ARM platform like the NVIDIA Jetson Nano.

2.2 Remarks over the state of the art

In this section, we present our remarks on the state of the art. To that end, firstly we present the accuracy results of the relevant related work discussed in the previous section for VLR FR benchmarks SCface, UCCS, QMUL-TinyFace, and QMUL-SurvFace. For the Quantized approaches, we show the accuracy benchmarks on ImageNet. Secondly, we show efficiency results on different hardware configurations for Lightweight CNNs and present the complexity of different quantized networks. Thirdly, we discuss the viability of these approaches for our real-time efficient VLR FR scenario. Finally, we present our remarks from analyzing these results and the research direction of our work.

2.2.1 Accuracy results on VLR datasets and Imagenet

In this section, we present the accuracy results for Heterogeneous and Homogeneous VLR approaches discussed in the previous subsection.

For the SCface dataset benchmarking, we present the results reported in Table 2.1, as presented in our work [75, 80].

Method	d1 (4.2m)	d2 (2.6m)	d3 (1.0m)	Mean accuracy
SCface [42]	1.82	6.18	6.18	4.73
CLPM [65]	3.46	4.32	3.08	3.62
CSCDN [125]	6.99	13.58	18.97	13.18
SSR [135]	7.04	13.2	18.09	12.78
L2softmax [93]	9.20	18.80	16.80	14.93
CCA [126]	9.79	14.85	20.69	15.11
DCA [47]	12.19	18.44	25.53	18.72
LM Softmax [142]	14.00	16.00	18.00	16.00
AM Softmax [121]	14.80	20.8	18.4	18.00
C-RSDA[25]	15.77	18.08	18.46	17.44
RIDN [28]	23.0	66.0	74.0	24.96
LDMDS [133]	62.7	70.7	65.5	66.30
VGG-Face* [80]	41.3	75.5	88.8	68.53
LightCNN* [80]	35.8	79.0	93.8	69.53
Centreface* [80]	36.3	81.8	94.3	70.87
VGG-Face-FT [80]	46.3	78.5	91.5	72.10
ResNet50-ArcFace* [80]	48.0	92.0	99.3	79.77
LightCNN-FT [80]	49.0	83.8	93.5	75.43
Centreface-FT [80]	54.8	86.3	95.8	78.97
FAN* [137]	62.0	90.0	94.8	82.27
ShuffleFaceNet* [80]	55.5	95.3	99.3	83.37
MobileFaceNetV1* [55]	57.0	95.3	99.8	84.03
ResNet50-ArcFace-FT [80]	67.3	93.5	98.0	86.27
MobileFaceNetV2* [80]	68.3	97.0	99.8	88.37
DCR-FT [74]	73.3	93.5	98.0	88.27
TCN-ResNet-FT [140]	74.6	94.9	98.6	89.37
FAN-FT [137]	77.5	95.0	98.3	90.27
ShuffleFaceNet-FT [80]	86.0	99.5	99.8	95.10
MobileFaceNetV2-FT [80]	**95.3**	**100.0**	**100.0**	**98.43**

Table 2.1: Recognition rate results (%) for the SCface dataset, taken from our work on [75, 80]. Methods marked with an asterisk (*) were not fine-tuned on this dataset, while those marked with FT were trained on large-scale HR datasets and fine-tuned on SCface. Lightweight CNNs show a compelling case for efficient and effective VLR FR on surveillance scenarios, per these results.

.

We specifically note the limitations of ResNet and VGG Face-based designs, for modern Homogeneous approaches. More modern methods such as ShuffleFaceNet and MobileFaceNet are capable of producing effective embeddings suitable for face recognition in VLR surveillance scenarios when synthesizing VLR face images combining interpolation methods. Heterogeneous approaches based on ResNet and VGG such as FAN and TCN see their performance hindered by these designs. In the case of classic CM approaches, the LDMDS approach shows the best recognition rate result.

Another aspect to consider is the training methodology, which is not consistent throughout the state of the art. Some methods randomly select 50 identities for training and 80 for testing, others need pre-training on large-scale databases (a caveat from CNN-based methods), and often the synthesis methods for VLR data differ across the state of the art.

For the UCCS dataset, we present Table 2.2 the verification results as reported in [23] and Table 2.3 shows identification results, from [67, 107]. These results show CentreFace implemented in [23], as a ResNet-24 model, with compelling performance. Suggesting that homogeneous FR approaches are a viable approach for VLR FR, as seen in different homogeneous approaches in Table 2.1, with the subspace projection for verification being an extremely important component. Dual Directed CapsNet shows the best result in men identification accuracy, however, they are prohibitively expensive to run in real-time.

Method	TAR@FAR				AUC
	30%	10%	1%	0.1%	(%)
DeepID2 [23]	93.1	83.4	61.7	37.9	93.8
CentreFace [23]	**99.6**	**97.0**	**87.8**	**75.5**	**99.0**
FaceNet [23]	98.2	93.8	79.4	63.4	97.8
VGGFace [23]	97.1	90.6	72.4	55.1	96.7
SphereFace [23]	94.0	84.9	60.2	24.7	94.1

Table 2.2: UCCS face verification rates (%), taken from [23]. CentreFace performs the best at 75.5% with TAR@FAR=0.1%.

Method	Mean Acc(%)
Robust Partially Coupled Nets [124]	59.03
Selective KD [40]	67.25
LMSoftmax for VLR [67]	64.90
L2Softmax for VLR [67]	85.00
Centreface [67]	93.40
DualDirectedCapsNet [107]	**95.81**

Table 2.3: UCCS face identification benchmark [107, 67] 80×80 size HR image to 16×16 VLR image matching. The Dual Directed CapsNet design is very robust for this scenario.

The face identification results for QMUL-TinyFace, shown in Table 2.4 show MobileFaceNet as the best performing network, trained using the ArcFace loss function. These results also suggest that placing special attention on robust feature extractors for face recognition is paramount for improving and maintaining VLR performance, with the channel-wise filtering operations being efficient and improving performance for this scenario, over other ResNet and VGG designs.

Method	Rank-1	Rank-20	Rank-50	mAP
DeepID2 [22]	17.4	25.2	28.3	12.1
SphereFace [22]	22.3	35.5	40.5	16.2
VGG-Face [22]	30.4	40.4	42.7	23.1
CentreFace [22]	32.1	44.5	48.4	24.6
ShuffleFaceNet [80]	43.1	58.9	64.5	34.0
MobileFaceNet [80]	**48.7**	**63.9**	**68.2**	**40.3**

Table 2.4: Face identification results for the QMUL-TinyFace dataset, taken from previous work [80]. The MobileFaceNet architecture holds a 5% Rank-1 accuracy margin over ShuffleFaceNet.

For the QMUL-SurvFace dataset, we present the results in Table 2.5. These results show MobileFaceNet with the best TAR@FAR=0.1% verification rate, being a harder scenario than the 1% threshold. The ShuffleFaceNet method also shows a competitive mean accuracy and AUC result, while being the most efficient.

Method	TAR@FAR		AUC	Mean Acc
	1%	0.1%		
VGG-Face [23]	20.1	4	85.0	78
DeepID2 [23]	28.2	13.4	84.1	76.1
SphereFace [23]	34.1	15.6	85.0	77.6
FaceNet [23]	40.3	12.7	93.5	85.3
CentreFace [23]	**53.3**	26.8	**94.8**	**88.0**
ShuffleFaceNet [80]	38.5	11.9	89.9	82.3
MobileFaceNet [80]	52.9	**33.1**	89.9	83.2

Table 2.5: QMUL-SurvFace verification results, at TAR@FAR=1% and 0.10%, taken from previous work [80]. MobileFaceNet outperforms CentreFace at the harder TAR@FAR=0.1% setting.

For Binary Neural Networks, we show the results for the ImageNet benchmark, as reported in the BNN literature. Table 2.6 shows k-bit details for weights and activations and compares them against a 32-bit Floating Point ResNet-18 architecture. The BinaryDenseNet design heavily improves recognition rates over BinaryNet and XNORNET. The more complex quantization approaches for Group-Net, ABC-Net, and Real-to-binary improve accuracy performance by at most 3% for the Top-1 benchmark against DOREFA-Net, with the quantization approach in DOREFA not heavily penalizing inference time performance.

Method	Bitwidth(W/A)	Top-1	Top-5
BinaryNet [26]	1/1	42.2%	69.2%
XNOR-Net [94]	1/1	51.2%	73.2%
Bi-Real Net [72]	1/1	56.4%	79.5%
BinaryDenseNet28 [138]	1/1	60.7%	82.4%
BinaryDenseNet37 [138]	1/1	62.5%	83.9%
DOREFA-Net [149]	2/2	62.6%	84.3%
QuickNet [4]	1/1	63.3%	84.6%
Group-Net [151]	(1/1)×4	64.2%	85.6%
ABC-Net [69]	(1/1)×5	65.0%	85.9%
Real-to-Binary [78]	1/1	65.4%	86.2%
ResNet-18 [151]	32/32	69.3%	89.2%

Table 2.6: Binary Neural Network benchmark for ImageNet, complemented from [78]. The network scaling factor is included in Group-Net and ABC-Net. QuickNet provides a better efficiency with only a 3% accuracy difference to Real-to-Binary.

2.2.2 Efficiency of Lightweight and Quantized CNNs

In this subsection, we firstly present the inference time performance for Lightweight CNNs on x86 architectures (Table 2.7), and lastly, we show state-of-the-art BNN performance accuracy-efficiency trade-off for BNNs (Figure 2.22).

For Lightweight CNNs, ShuffleFaceNet has the best efficiency-accuracy trade-off for a real-time application on Laptop x86 CPUs. ShuffleFaceNet's inference performance is of 34.4 FPS compared to MobileFaceNet's 16 FPS, with competent accuracy results on VLR FR datasets as shown in the previous subsection. This is mainly due to the fact that it only processes part of the feature map in each stage, uses almost exclusively depthwise convolutions, and encourages robustness with the channel shuffle at the end of each unit.

Network	# Params. (Millions)	2× GTX 1080ti	GTX 1080Ti	GTX 1660Ti	Laptop GTX 1050Ti	Laptop Intel i7 7700HQ
Light CNN - 4 [130]	6.8 M	5.49 ms	12.67 ms	14.09 ms	55.50 ms	2,653.76 ms
Light CNN - 9 [130]	8.1 M	6.22 ms	14.36 ms	15.88 ms	56.17 ms	2,106.72 ms
VGG [106]	144.9 M	3.39 ms	7.83 ms	108.97 ms	35.29 ms	1,523.40 ms
VGGFace [88]	41.1M	3.99 ms	9.22 ms	14.24 ms	21.61 ms	433.39 ms
Resnet100 [48]	65.2M	2.76 ms	5.26 ms	12.96 ms	59.61 ms	285.64 ms
Light CNN - 29 [130]	31.0 M	2.01 ms	4.63 ms	2.38 ms	7.93 ms	126.71 ms
VarGFaceNet [132]	4.9 M	0.85 ms	1.48 ms	3.48 ms	27.09 ms	126.59 ms
MobileNetv2 [99]	**1.8 M**	0.80 ms	1.46 ms	4.50 ms	14.76 ms	103.69 ms
MobileNetv1 [55]	3.2 M	**0.69** ms	1.21 ms	1.88 ms	20.06 ms	98.99 ms
VarGNet [146]	4.2 M	0.68 ms	1.18 ms	2.16 ms	5.42 ms	70.40 ms
MobileFaceNetV2 [20]	2.0 M	0.88 ms	1.48 ms	3.27 ms	7.28 ms	62.45 ms
MobileFaceNetV1 [20]	3.3 M	0.74 ms	1.31 ms	1.61 ms	8.23 ms	53.49 ms
ShuffleNet - 2.0 [76]	5.3 M	1.10 ms	1.96 ms	18.79 ms	N/A	42.92 ms
ShuffleFaceNet - 2.0 [79]	4.5 M	1.00 ms	1.77 ms	2.41 ms	6.95 ms	37.46 ms
ShuffleNet-1.5 [147]	2.5 M	0.77 ms	1.33 ms	2.77 ms	12.25 ms	32.98 ms
ShuffleFaceNet-1.5 [79]	2.6 M	0.77 ms	**1.34 ms**	**1.86 ms**	**4.68 ms**	**29.08 ms**

Table 2.7: Inference time for Lightweight Convolutional Neural Networks, taken from previous work [75]. We note that ShuffleFaceNet shows the best inference runtime performance for an Intel Laptop Processor, of around 34.4 FPS for a single image.

In Figure 2.22, we present runtime performance for state-of-the-art Binary Neural Networks. The best trade-off is shown by QuickNet, Real-to-Binary, and BinaryDenseNet designs, for 224×224 image inputs tested on an ARM Cortex A76 CPU. The DepthWise operators in QuickNet show an effective efficiency-accuracy trade-off for BNN block design, similar to MobileFaceNet.

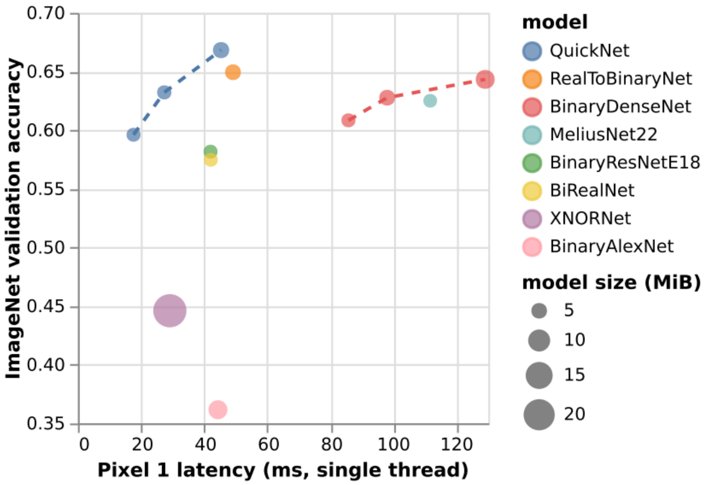


Figure 2.22: Binary Neural Network efficiency performance [4], with ImageNet used as accuracy benchmark. The QuickNet design shows a remarkable accuracy-efficiency trade-off.

2.2.3 Discussion on efficient Homogeneous approaches and research direction

In order to build a strong baseline for binarizing CNNs, the following must be considered [78]:

- Use downsampling layers with real-values due to signal degradation when skipping connections at the downsampling step [72]
- Use PReLU instead of ReLU for activation [15]. This also improves performance in Lightweight CNNs for face recognition [21, 79]
- Binarize weights prior to the activations [15].
- Learn scaling factors α and β via back-propagation, not fixed [15]. The scaling factors can be computed using stacked linear operations [78].
- Train using weight decay of $1e-5$ [78] on when binarizing the activations and later a weight decay of 0 with both weights and activations binarized [15].
- Do not back-propagate using binary representations as it degrades performance harshly poorly [94] and it does not have any benefit at inference time.
- Use a vector of scaling factors α for every element in the binary weights [149].

Furthermore, structural changes are needed to reduce the loss of signal in the binarization process as evidenced by the structures of ABC-Net [69] and the Binary Single Image SR Network [131].

Aside from the strong baseline building for CNNs, we identify the following areas of opportunity:

- The main pass for binarizing weights is to use the sign function. This does not take into account the median of the distribution of the data which might be a determining factor to better approximate real values into binary values. This is partially mitigated by ABC-Net [69] by using different activation scales.

- The scaling factors α and β are always linear in all cases, except for the training step in real-to-binary [78]. Adding further activations such as PReLU or other non-linear functions sequentially, such as $\tanh$, for these terms may further improve performance.

- Binary convolution blocks are often put in place for standard convolutions, which improves efficiency. An attempt to approximate one floating-point residual convolutional blocks using binary residual blocks was proposed in [151]. However, there is a gap for specialized binary architecture building blocks akin to lightweight convolutional neural networks. Sharing information between channels, which is what makes a strong case for homogeneous feature extraction for Very Low Resolution Face Recognition, is not present in binary convolutional units.

- The regular binarization step consists on Batchnorm → Activation → BinaryConversion → Pooling [94]. Pooling layers have proven to be ineffective for CNNs as evidenced in Lightweight CNNs [21, 79]. Strided convolutions for downsampling avoid further loss of information by having the information present in the downsampled representation instead of completely eliminating data in the pooling step.

- The effect on binarization for face recognition purposes has not been explored. Approximating face descriptors using binary representations remains as an opportunity area.

- Locality has not been fully exploited in current binarization block designs, other than [131]. This local information sharing is paramount for applications in face recognition and subspace projection mapping.

For extremely limited hardware (Section 1.7), improving on Binary Neural Network technology suggests a compelling case for a VLR FR real-time approach. These improvements need to have the VLR hardest face verification scenarios into consideration, focusing on the efficiency of generating and comparing face descriptors. These improvements are Binary Neural Network design decisions at the general architecture and convolutional block level. Considering our experimentation and analysis works in [80, 75, 81] and also presented in this Section, with our Theoretical Framework Section 2.3, we are able to propose a binarized approach for VLR FR as a Homogeneous VLR FR method.

2.3 Theoretical Framework

In this section, we present the specific components from the state of the art that take part in our hypothesis for our binarized approach to Very Low Resolution Face Recognition. We discuss quantization considerations, CNN architecture designs for FR, BNN block designs, and the state-of-the-art training environment for FR.

2.3.1 CNN architecture design for FR

In this subsection, we discuss the architecture components used on CNNs and BNNs. The typical structure of a CNN architecture involves: Head block → Feature Extraction blocks → Embedding block → Fully Connected blocks for training. Firstly, we present the operators present in compact CNN designs. After that, we present the blocks used in different efficient networks for FR.

2.3.1.1 Efficient CNN operators

1×1 convolutions, also known as pointwise convolutions, have a spatial kernel of only 1×1 with a channel dimension C. This acts as a focused channel-wise operator, saving operations from traditional convolutional squared kernels per channel.

DepthWise Convolutions have independent 2D kernels for each input channel. For each channel in the feature map, it is convolved with its corresponding 2D kernel at its depth channel, then concatenates the output. This reduces the number of channel filters present in regular convolutions.

Separable DepthWise convolutions stack DepthWise Convolutions and pointwise convolutions in a single concept, illustrated in Figure 2.23. The convolutional kernel is separated spatially to perform the pointwise convolution step. The separable convolution has a parameter count of $C_{in} \times (k^2 + C_{out})$, with a complexity of $H_{in} \times W_{in} \times C_{in}(k^2 + C_{out})$, against the complexity of a regular convolution of $H_{in} \times W_{in} \times C_{in} \times C_{out} \times k^2$.

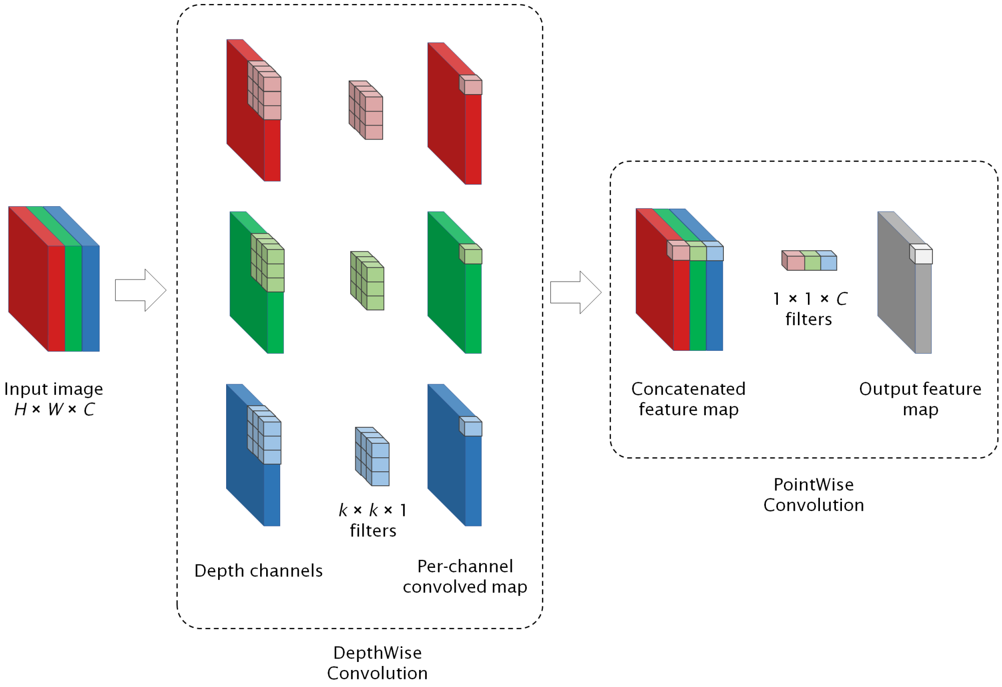


Figure 2.23: The Separable DepthWise Convolution operation. Independent kernels are split through the depth channels, then calculated spatially with a $1\times1\times C$ filter, greatly optimizing the number of operations from a regular convolution.

Grouped convolutions split the input along the channel dimension into g groups, such that each group has an independent channel filter. The output is later concatenated for the final output. This effectively lowers the complexity of a regular convolutional layer proportionally to the number of groups. The complexity is expressed as: $(H_{in} \times W_{in} \times C_{in} \times C_{out} \times k^2) \,/\, g$.

2.3.1.2 Units

We make a distinction for convolutional units as very basic convolutional layer blocks, which take part in other bigger and more complex blocks.

The most basic and common convolutional blocks in CNN design usually are Convolutional Block units containing Convolution → Batch Normalization → Activation. In CNNs for FR, the activation function of choice is the PReLU [115] activation. The PReLU activation is formally defined as:

$$f(y_i) = \begin{cases} y_i & \text{if } y_i > 0 \\ a_i y_i & \text{if } y_i \leq 0 \end{cases} \tag{2.4}$$

Where a_i is a trainable parameter that re-scales the feature map convolved output. This is beneficial for face recognition purposes as it allows for negative responses in the activation components [20], compared to ReLU.

Residual connections, also commonly called skip connections, are heavily utilized in the CNN literature. Firstly introduced in ResNet [48], the idea can be represented as $G(x) = F(x) + x$, where F is one or more convolutional layer operations applied to input x. This skip connection approach is light on resources and helps to compensate for the signal degradation after many convolutional layers deeper down the network.

Many Lightweight CNNs [57, 20, 132] employ Squeeze-And-Excitation (SE) blocks in various block designs. Figure 2.24 shows the basic functionality of this block. The block calculates a scaling factor for the feature representation, with a 1×1×C shape. Usually, this is calculated using a Global Average Pooling layer.

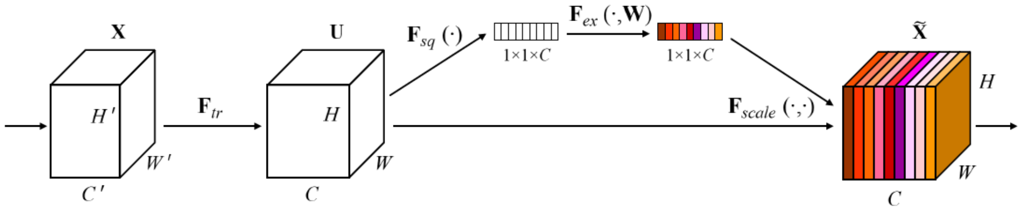


Figure 2.24: Squeeze-And-Excitation block, taken from [57]. This block calculates independent scaling factors for all channels and uses them to scale the feature map posteriorly.

2.3.1.3 Block Designs

Building blocks for CNNs are the core of modern CNN architectures. They are designed with a specific purpose in mind. In the case of Lightweight CNNs, they are designed to compensate for lower accuracy when using less costly operators, at different points of the network.

2.3.1.3.1 Head blocks

The head block section is responsible for handling the direct input signal from the input layers. Avoiding information loss at this stage is paramount for a good information flow in deeper stages from the network. In modern Lightweight CNN approaches for FR, the input layer size is 112×112.

We consider the first layers of SqueezeNet [57] as the block design. The authors directly downsample the input image with a 7×7 kernel and a 2-stride filter, expanding from 3 channels to 96, and applying a max pooling operation with a 3×3 / 2 patch.

The MobileFaceNet head block, depicted in Figure 2.25a uses a 2-strided convolutional block to downscale the input image and expand to 64 channels. Afterwards, they propose to use a Separable DepthWise Convolutional Block with a 3x3 kernel, before sending the feature map to the inverted residual block structures.

The ShuffleFaceNet architecture employs a convolutional block with a 3×3 with stride 2 to rescale the 112×112 input to 56×56, and expands to 24 channels. This employs a fast downsampling strategy for the posterior feature extractors.

The VarGFaceNet head block design consists of a more complex approach. This block is illustrated in Figure 2.25b. The input signal is not downscaled in the first convolutional layer. Instead, the full 112×112 area is used to expand to 40 channels, retaining more information. Afterwards, it features two separate branches for downscaling the feature map and performs a set of one variable group convolution before one pointwise convolution. One branch performs two sets of these operations and ends with a SE block before adding the signal from the other branch with a single variable group convolutions and pointwise output.

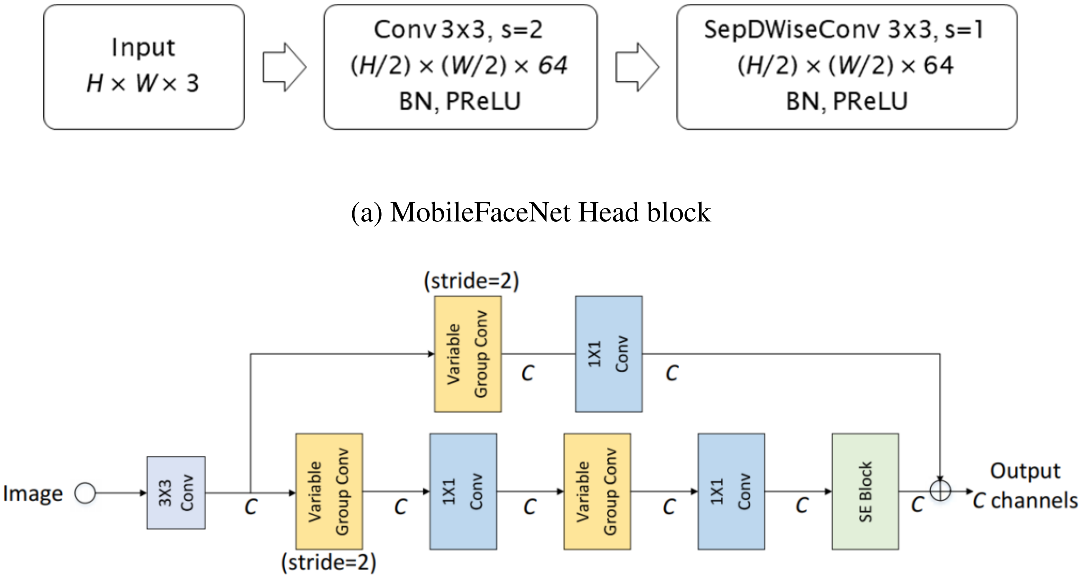
2.3.1.3.2 Feature Extractor blocks

Block designs for most state-of-the-art CNNs are based on ResNet-style [48] blocks. This involves stacking this design of three convolutional layers and a residual connection from the input of the block summed with the output of the block. However, this design has been identified to be a limit factor for VLR FR [75]. For optimizing the feature map sizes and reducing convolutional operations, inverted bottleneck designs and depthwise convolutions [20] are commonly used by Lightweight CNNs.

Lightweight CNNs make heavy use of DepthWise and PointWise convolutions in different ways. ShuffleFaceNet almost exclusively uses DepthWise operators in its block design. In its main block, it features a channel split operation, then convolves only a part of the input channels with pointwise and separable convolutions, then concatenates the feature map and shuffles the channel. This channel shuffle operation promotes other channels to be convolved at a later stage. In the downsampling block, the authors use 2-strided DepthWise convolutions, pointwise convolutions, and channel shuffling as well.

The MobileFaceNet feature extractor primarily constitutes residual bottleneck blocks and the inverted bottleneck residual blocks. They are based on the bottleneck and expansion block depicted in Figure 2.27b. The bottleneck design features a depthwise convolutional unit (strided when used for downsampling), followed by a pointwise unit without the activation. This design may also have a residual connection to the end of the block, dubbed as an inverted residual structure. Their expansion block expands the depth channels using a pointwise convolution, then uses a depthwise convolutional block, and a final pointwise convolution unit.

The VarGFaceNet feature extractor blocks heavily feature Variable Group convolutions



(a) MobileFaceNet Head block

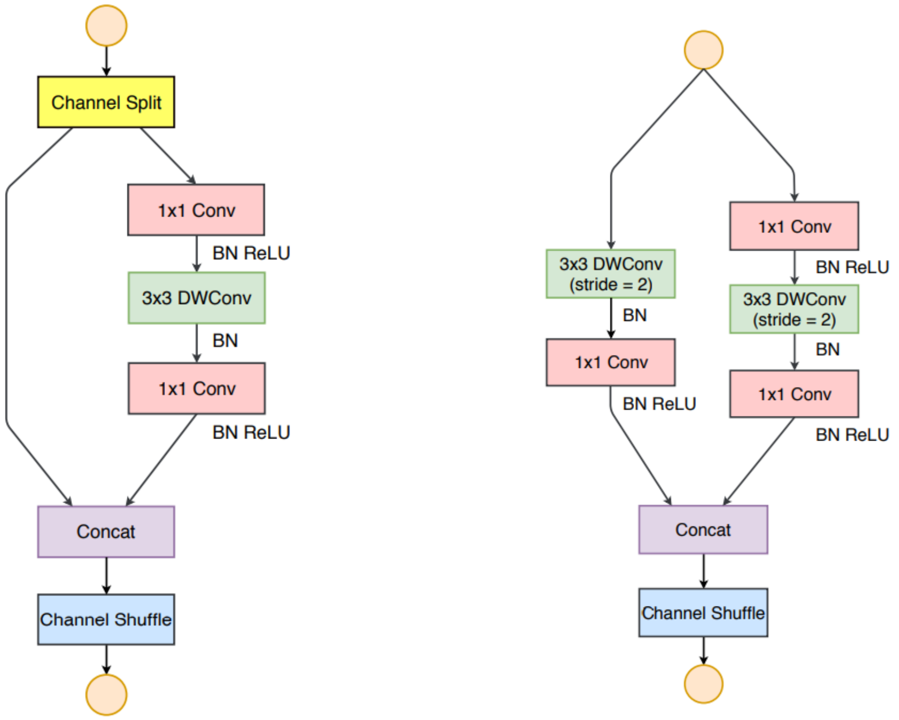
(b) VarGFaceNet head block, taken from [132]

Figure 2.25: Head blocks for Lightweight CNNs. Subfigure 2.25a shows the head block for MobileFaceNet with the channel expansion. Subfigure 2.25b shows the VarGFaceNet head block strategy with residual connections, variable group convolutions, and the same number of output channels.

and pointwise convolutions. The normal block uses variable group convolutions to duplicate the number of channels and subsequently reduce them back to the original number of channels with a pointwise convolution. This process is done two times, with a SE block and a residual connection at the end. The downsampling block reduces the feature representation spatially (2-stride) while outputting a $2 \times C$ channel representation. This design processes the same signal with three different sets of variable group convolutions followed by pointwise convolutions. Two of those outputs are joined and then processed with another variable group convolution, which duplicates again the number of channels, and another pointwise convolution reduces the channels by half. One of the outputs at the first step is joined with the latest feature map before exiting the downsampling block. These designs are illustrated in Figure 2.28.

2.3.1.3.3 Embedding blocks

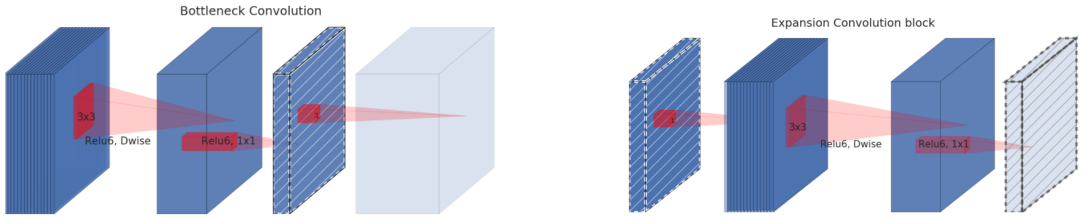
VarGFaceNet and MobileFaceNet embeddings usually increase the number of channels and perform a reduction to $1 \times 1 \times C$ using a 7×7 kernel with a variable groups convolution or a Global DepthWise convolution, respectively. This solution works well when we do not have number representation constraints and can easily expand the channel representation at the embedding layer without losing information. However, this is not the case with binarized networks. State-of-the-art BNNs do not employ this approach in their embedding settings.



(a) ShuffleFaceNet feature extractor

(b) ShuffleFaceNet downsampling

Figure 2.26: ShuffleFaceNet main blocks based on ShuffleNetv2, taken from [76]. The feature extractors in Subfigure 2.26a feature the channel split with depthwise convolutions and channel shuffle. Subfigure 2.26b shows the downsampling strategy with two branches of 2-strided depthwise convolutions, pointwise convolutions, and a channel shuffle



(a) MobileFaceNet bottleneck

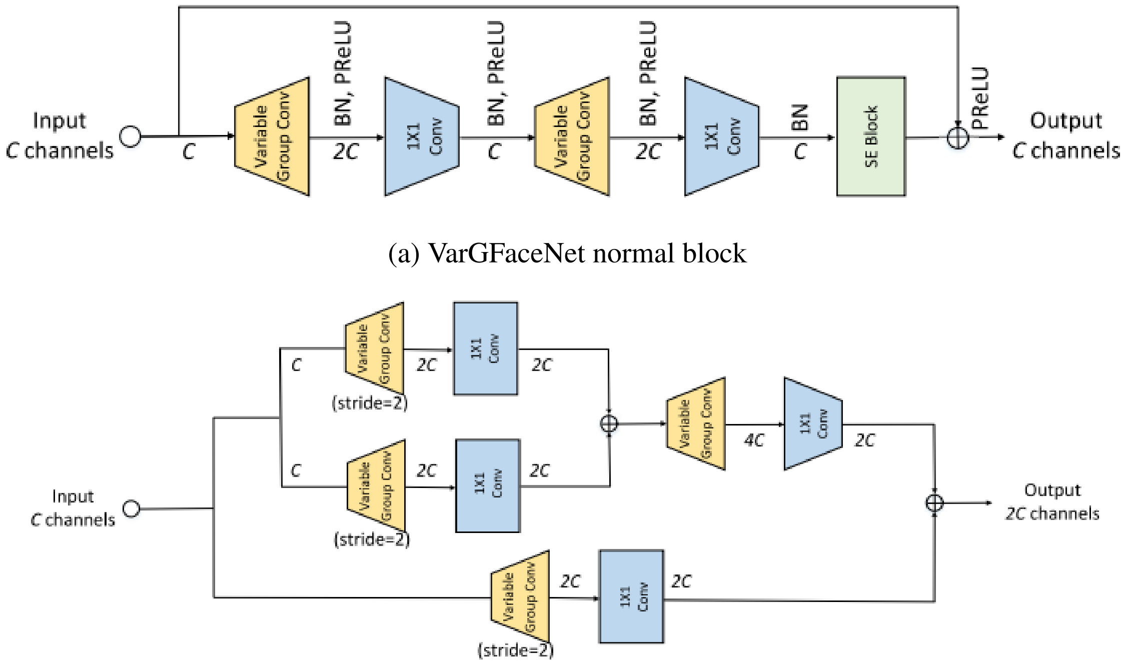
(b) MobileFaceNet expansion block

Figure 2.27: MobileNetV2 block design, also used on MobileFaceNet, taken from [99]. The block design shown is based on MobileNetV2. Subfigure 2.27a shows the bottleneck structure for feature extraction and Subfigure 2.27b shows the strategy for expanding the channels.

2.3.2 BNN block designs

In this subsection, we show BNN convolutional block designs of state-of-the-art binary approaches, as discussed in subsection 2.2.2. These blocks mainly focus on compensating for the heavy loss of information from the quantization process.

The Real-to-Binary block [78], illustrated in 2.29, scales down the input to a single



(a) VarGFaceNet normal block

(b) VarGFaceNet downsampling block

Figure 2.28: VarGFaceNet feature extractor and downsampling blocks, taken from [132]. The normal block (2.28a features a channel expansion and reduction, with variable group convolutions and a residual connection. The downsampling block (2.28b has two main branches with 2-strided variable group convolutions, the result is joined before outputting.

vector by using a Global Average Pooling Layer. After that, a series of linear transformations are applied in cascade to add to the calculation of the term.

In the BinaryDenseNet block designs, the authors proposed to increase efficiency by removing the pointwise convolution and reducing the channels by half in the residual connection. They also added extra shortcuts, compared to the original DenseNet design. Furthermore, the representation is efficiently downsampled by first applying max pooling, then a ReLU activation, and finally a pointwise convolution with reduced channels in a full-precision representation. These reductions in size are depicted in Figure 2.30. With this design, the authors report a $\sim 64\times$ speedup compared to the original DenseNet architecture.

In QuickNet [4], the authors propose to downsample the representation by half at the first full-precision Convolutional layer while expanding the channels to 16, and later at the DepthWise convolution. In contrast with Lightweight CNNs, this approach does not expand and reduce the depth channel intermediate representations, as presented in the embedding layers for MobileFaceNet and VarGFaceNet. The transition block is used in the feature extraction part of the network, with a depthwise convolution to downsample the representation after joining the residual connection from the previous layer and applying a max pooling operator. This design features the ReLU activation at a floating-point representation. These designs are depicted in Figure 2.31.

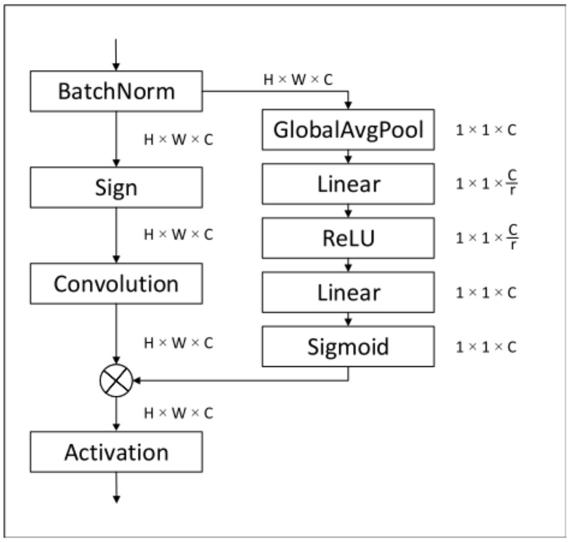
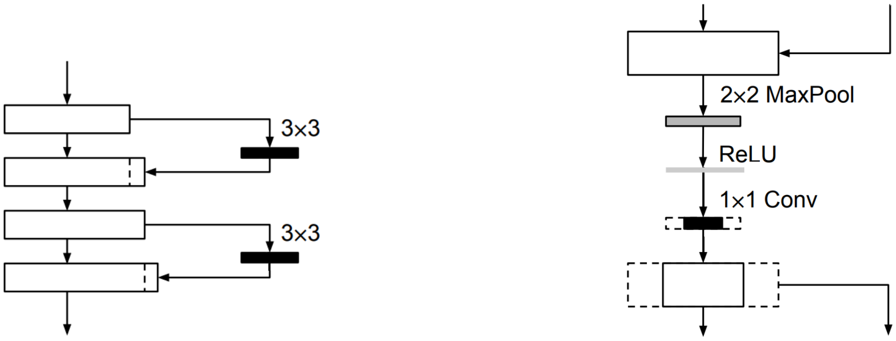


Figure 2.29: Real-to-binary convolution block, taken from [78]. The right side is the gating function which computes the scaling factor via convolutions, similar to an Inception or Squeeze-And-Excitation block.



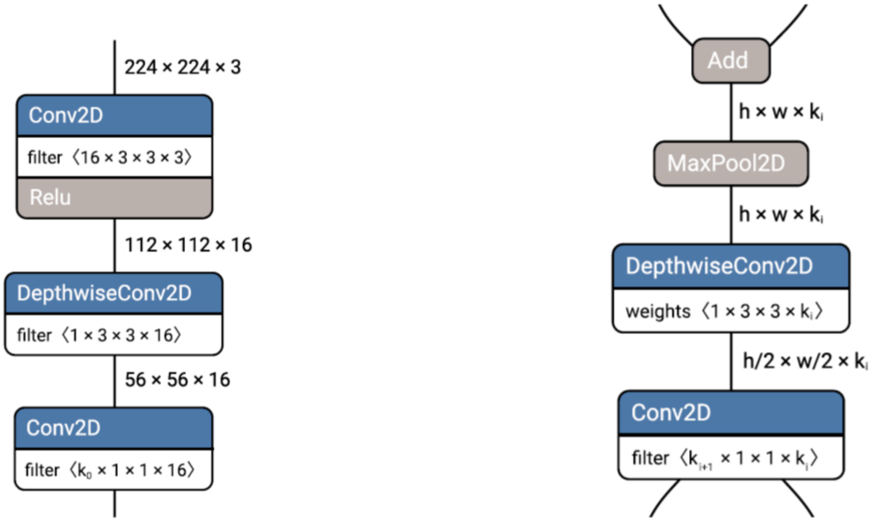
(a) BinaryDenseNet Residual convolution block

(b) BinaryDenseNet transition block.

Figure 2.30: BinaryDenseNet block designs for feature extraction, taken from [8]. Subfigure 2.30a shows the modified DenseNet feature extractor and Subfigure 2.30b shows the spatial downsampling strategy with a reduced intermediate feature map.

2.3.3 Quantization schemes

In this subsection, we discuss the quantization methods used to inform the quantization and block design choices behind our binarized approach. Firstly, we discuss the regular quantization process via scaling factors, then the scaling factor residual accumulation mechanism relevant to quantized block design, and the DOREFA quantizer applying non-linearities for efficient inference time quantization.



(a) Quicknet first convolution block

(b) QuickNet transition block.

Figure 2.31: QuickNet proposed block designs, taken from [4]. The authors heavily expand the representation to only 16 channels, but heavily spatially downsizing the representation at the first steps (Subfigure 2.31a. In the transition block (Subfigure 2.31b), a 2-strided depthwise convolution is used after a max pooling operation for reducing the feature map spatially, then this representation is filtered with a regular quantized convolution.

2.3.3.1 Binarization using scaling factors

In XNORNET [94], the binarization process is performed by estimating the real-value weight filter W using a binary filter B, where $B = -1, +1$. It uses the scaling factor α such that B and α approximate W.

$$I * W \approx (I \oplus B)\alpha \tag{2.5}$$

Estimating binary weights: Find estimation $W \approx \alpha * B$ The objective function is defined as:

$$J(B, \alpha) = ||W - \alpha B||^2$$
$$\alpha*, B* = \alpha B J(B, \alpha)$$

Where $B_i = +1$ if $W_i \geq 0$ and $B_i = -1$ if $W_i < 0$. This can be modeled using the sign function as: $B^* = sign(W)$. For finding the optimal scaling factor α^*, the derivative of the objective function J with respect to α is employed, finally modeling the optimal α as:

$$\alpha^* = \frac{(W^T B^*)}{n} \tag{2.6}$$

For updating the parameters in the back-propagation step, real-value weights are utilized. The reason is that in gradient descent the parameter changes are tiny, as such, these small changes get ignored when the gradients are binarized. Floating-point representations are used as input for the convolutional layers.

The basic Straight-Through-Estimators (STE) include trainable real-valued scalars α and β, used to linearly approximate real-value weights with their binarized counterparts. This simple binarization process leads to a heavy loss of information.

The STE quantization is based on a Bernoulli distribution, $q \sim Bernoulli(p)$. As such, generalizing the k-bit STE for weight vector r_i is defined as:

$$\begin{aligned} \textbf{Forward: } & r_o = \frac{1}{2^k - 1} \text{round}((2^k - 1)\, r_i) \\ \textbf{Backward: } & \frac{\partial J}{\partial r_i} = \frac{\partial J}{\partial r_o} \end{aligned} \tag{2.7}$$

The binarized convolution is defined as:

$$x \cdot y = \sum_{m=0}^{M-1} \sum_{k=0}^{K-1} 2^{m+k} \text{bitcount}[\text{and}(Q_m(x), Q_k(y))] \tag{2.8}$$

2.3.3.2 Residual and accumulating scaling factors

The Bit Accumulation Mechanism (BAM) on the Super Resolution method [131], illustrated in Figure 2.32, features alternating residual connections for the α and β quantization terms. In this structure, every 2 layers constitute a Local Binarization (LB) block and the conjunction of various LB blocks constitutes a High Precision Binarization (HPB) block. This scheme aggregated the weight scaling vectors α and activation scaling vectors β in every LB block. The output of the weight scaling vector is then summed independently to the output of the first Binary Convolution of the next LB block.

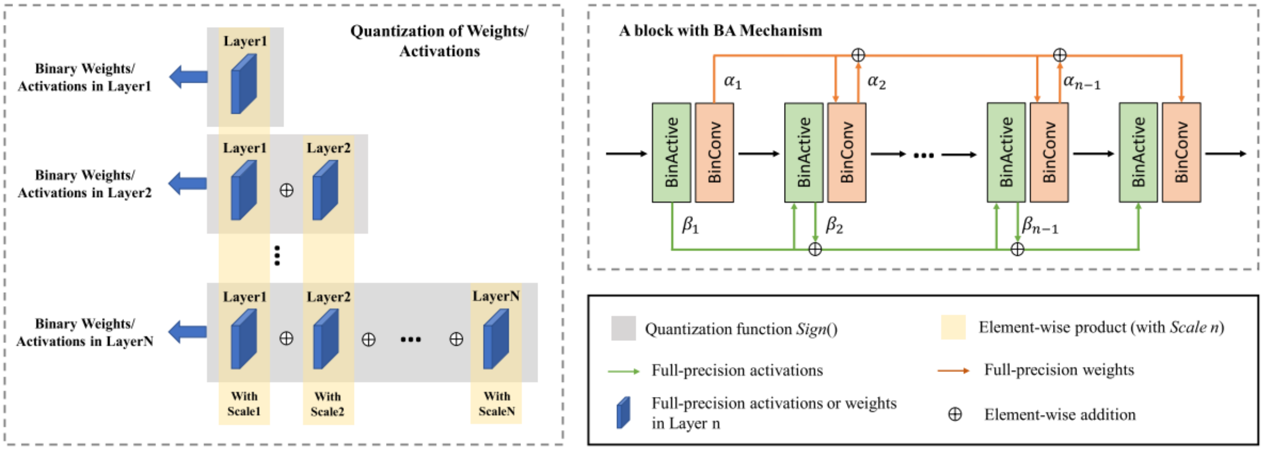

Quantization of Weights/ Activations
Binary Weights/ Activations in Layer1
Binary Weights/ Activations in Layer2
Binary Weights/ Activations in LayerN
Layer1
Layer1
Layer2
Layer1
Layer2
LayerN
With Scale1
With Scale2
With ScaleN
A block with BA Mechanism
BinActive
BinConv
α_1
α_2
α_{n-1}
β_1
β_2
β_{n-1}
Quantization function Sign()
Element-wise product (with Scale n)
Full-precision activations
Full-precision weights
Full-precision activations or weights in Layer n
Element-wise addition

Figure 2.32: Bit Accumulation Mechanism (BAM), taken from [131] for non-linear feature mapping. The individual scaling vectors of weights α and activations β are aggregated sequentially and independently from the standard convolution procedure.

2.3.3.3 Binarization using DOREFA

In DOREFA-Net [149], the authors propose to introduce the $\tanh$ non-linearity to the quantization process and scale the terms to the $[-1, 1]$ range. This also adds stability to the quantization process, and retains the k-bit representation flexibility:

$$\begin{gathered}\textbf{Forward: } r_o = 2 \text{ quantize}_k\left(\frac{\tanh(r_i)}{2\max(|\tanh(r_i)|)} + \frac{1}{2}\right) - 1 \\ \textbf{Backward: } \frac{\partial c}{\partial r_i} = \frac{\partial r_o}{\partial r_i}\frac{\partial c}{\partial r_o}\end{gathered} \tag{2.9}$$

This notion of using the $\tanh$ as a soft regularizer or clipper in binary networks is also used in the Knowledge Distillation training process presented in Real-to-binary [78].

2.3.4 State-of-the-art FR training environment

We also discuss the training environment for binarized FR. This is also relevant to the design and evaluation of our approach since it impacts learning for the generated descriptors. As a Homogeneous FR approach, we discuss the foundations of Softmax-based loss functions and gradient optimization; we do not rely on complex training methodologies, such as Knowledge Distillation, to validate the performance of our approach.

2.3.4.1 Loss functions

Loss functions used in the state of the art for FR include Softmax-based functions, center loss [128], and the triplet loss[19]. Effective loss functions for large-scale face data are Cosface[122], Sphereface[70], and arcface[31]. The margin-based Softmax functions are generalized in the following equation [109]:

$$L = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{s(cos(\theta_{y_i} m_1 + m_2) - m_3}}{e^{s(cos(\theta_{y_i} m_1 + m_2) - m_3} + \sum_{j=1, j\neq y_i}^{n} e^{s\, cos\theta_j}} \tag{2.10}$$

Where s is the scaling factor directly impacting the gradients, and the margins m1, m2, and m3 affect the angle in the projected space. The Cosface loss function can be represented with $m_1 = 1$, $m_2 = 0$, and $m_3 = 0.35$. The Sphereface loss with values $m_1 = 4$, $m_2 = 0$, and $m_3 = 0$. And the Arcface function as $m_1 = 1$, $m_2 = 0.5$, and $m_3 = 0$.

In particular, the ArcFace loss performs the best on most large-scale FR benchmarks. This additive angular margin loss function was proposed to promote inter-class discriminative properties by enforcing an angled margin. This constitutes to a more accurate geometric spherical representation in which features from similar classes get compressed while keeping them more adequately separated from the features from other classes. The geometrical representation is shown in Figure 2.33.

Aside from the accuracy gains, this loss function is very efficient with very little overhead over the original base idea in SphereFace [70]. they achieved this by directly employing the $\arccos$ function instead of the more complex double-angle formula.

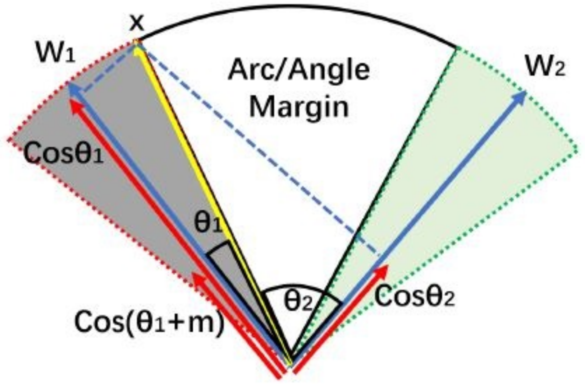


Figure 2.33: Geometrical interpretation of the ArcFace loss function, taken from [31]. At the time of training, the class center is learned and the angled margin is maximized using the cosine function.

2.3.4.2 Optimizer

The Stochastic Gradient Descent optimizer updates the objective function J in mini-batches [97], described by:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i:i+n)}; y^{(i:i+n)}) \tag{2.11}$$

Where J is the objective function, η is the learning rate, and θ are our parameters. The θ parameters are regularized using the momentum hyperparameter γ and the regularizer term υ_{t-1}. This variant is known as SGD with decoupled weight decay (SGDW). The θ estimation is updated with the following equations:

$$\begin{aligned} \upsilon_t &= \gamma \upsilon_{t-1} + \eta \nabla_\theta J(\theta) \\ \theta &= \theta - \upsilon_t \end{aligned} \tag{2.12}$$

Chapter 3

BinaryFaceNet: VLRFR using BNNs for Real-time performance

In this chapter, we present in detail our approach named BinaryFaceNet. BinaryFaceNet is based on a block architecture, scalable in depth with a depth channel hyperparameter in the network definition.

3.1 Approach

As per our analysis from the previous section, we found a gap for specialized binary architecture building blocks akin to Lightweight CNNs for face recognition. The main inspirations for our approach are the design guidelines from MobileFaceNet [20], VarGFaceNet [132], and Group-Net [151]. We emphasize sharing information between channels, which is what makes a strong case for homogeneous feature extraction for face recognition.

3.2 Block-based design

The network is organized in a Head block, a Global Feature Extraction (GFE) block, two Local Feature Extraction (LFE) blocks, and an Embedding block. We established the following design guidelines for our method:

- We avoid pooling layers in favor of using greater strided convolutions when we need to decrease the feature map size.

- Using a Squeeze-and-Excitation (SE) module in the head block, calculating the SE factor using progressive downsizing via quantized convolutions.

- Using BatchNorm and PReLU with floating-point precision as an activation function after convolutional layers.

- Adding alternating residual connections (Adds) in the block designs to compensate for the heavy signal degradation from the quantized layers.

- Grouped convolutions with 2-step strides for efficiently downsampling the feature representations.
- Using the DOREFA quantizer at every quantized layer for adding non-linearity to the quantization and optimization steps, further stability, and k-bit flexibility to the representation.
- Using a compact 128-Dimensional vector as feature output.
- Using Quantized DepthWise(QDepthWiseConv) and Quantized Separable (QSeparableConv) convolutions to share channel-wise filters.

Our basic Quantized Convolutional block (QConvBlock) is defined with a Binarized Convolution followed by a BatchNorm layer and a PReLU activation. The BatchNorm layer aids in the stability and convergence of the network. For the network dimensions, we take an approach similar to MobileFaceNet with 2-strided convolutions. Table 3.1 describes the overview of our downsampling methodology in our method.

Name	Input	Output
Input layer	3×112×112	3×112×112
Head Block	3×112×112	64×56×56
GFE Block	64×56×56	64×56×56
LFE Block 1	64×56×56	64×28×28
LFE Block 2	64×28×28	64×14×14
Embedding Block	64×14×14	128×1×1

Table 3.1: General overview of the steps and feature map sizes in BinaryFaceNet. We first expand to 64 channels in the head block and downsize the feature map, then process a high-resolution 56x56 feature map in the General Feature Extraction (GFE) block, and further downsize the representation at the start of the Local Feature Extraction (LFE) blocks. We finalize with the Embedding block downscaling and expanding the representation channels to a 128-Dimensional vector using strided convolutions.

3.2.1 Head Block

The main challenge of BNN is signal degradation. To improve the discriminative ability further down the network, we maintain the input floating-point representation for retaining the details in the input signal as much as possible. Figure 3.1 details the operations in the head block. Our Head block expands the representation to 64 channels, then calculates the SE weight factor for a weighted floating-point feature map subsequently. However, we avoid using pooling operations due to their reduced discrimination ability and use a Reduction block to favor quantized grouped strided convolutions to retain more information and extract more accurate scaling factors to the data while downsizing. Finally, we scale the data with the output from the SE block and downscale the representation using a 2-strided convolution to be passed down along the network. We choose to use grouped convolutions heavily in this step due to their discrimination ability while increasing the efficiency when downsizing the representation.

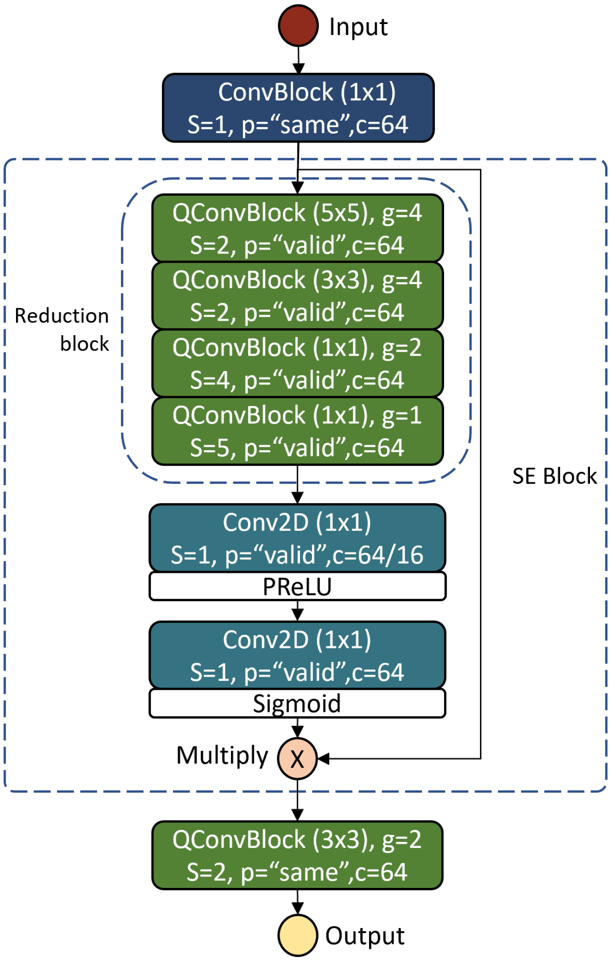
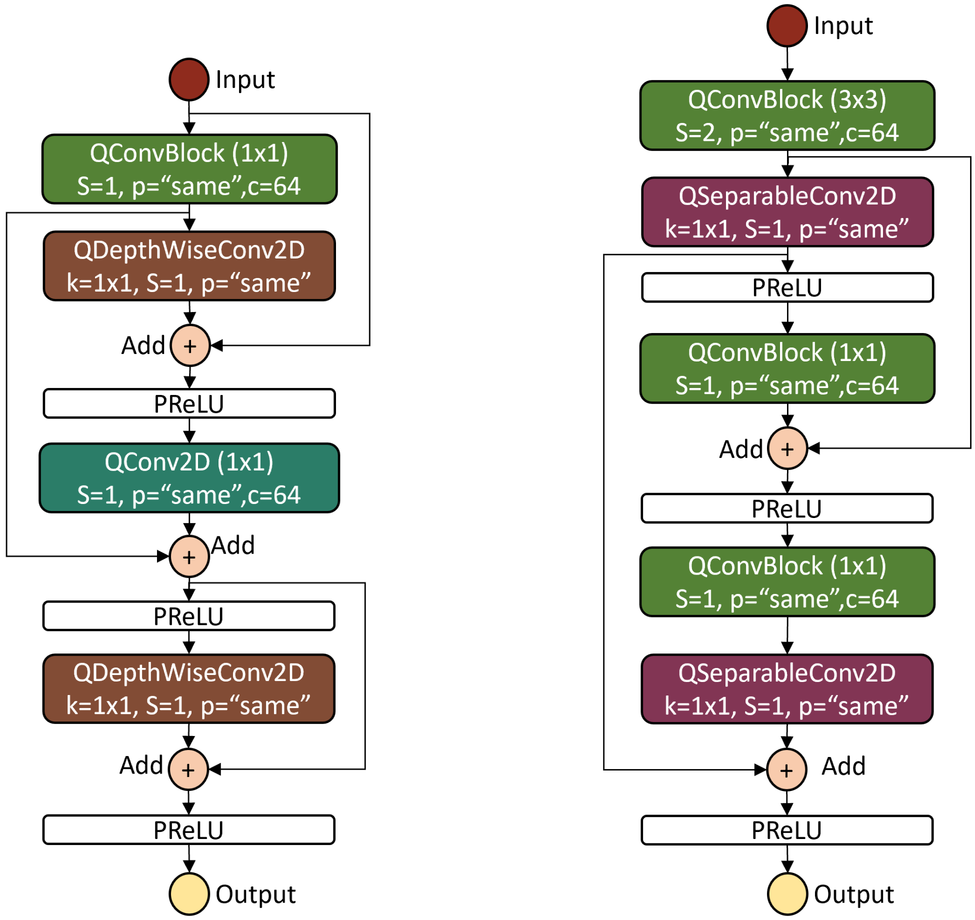


Figure 3.1: Head block design of BinaryFaceNet. We propose to use a Reduction block with QConv operations inside the Squeeze-and-Excitation (SE) block for calculating the weights, instead of a pooling layer. We use strided (2,2,4,5 sequentially) convolutions with different kernel sizes (5,3,1,1 sequentially) and grouped convolutions to alleviate the operations overhead.

3.2.2 Global Feature Extraction Block

Our Global Feature Extraction block, shown in Figure 3.2a, works with the same representation size throughout. This allows us to combine the input signal with the residual of two posterior convolutional operations. This block processes HR feature maps with a resolution above 32×32. Similar to QuickNet [4], we use Quantized DepthWise Convolutions (QDepthWiseConv) for efficient channel filtering through the depth dimension of the representation. We choose the QDepthWiseConv operation to reserve more complex operations for the other building blocks in our architecture while still sharing filters through the channel dimension.

We use a PReLU layer to re-scale the sum of the feature maps. The input signal is introduced back after two convolutional operations and the signal from the first convolution gets reintroduced after a third single convolutional layer. This helps to compensate for the signal degradation from the quantization process at different processing points. The compensated signal before activation is later reintroduced after a QDepthWiseConv layer and re-scaled. This last signal is the output of the GFE block for posterior processing at the LFE blocks.



(a) Global Feature Extraction Block

(b) Local Feature Extraction Block

Figure 3.2: BinaryFaceNet feature extractors. Figure 3.2a shows the Global Feature Extraction block in our architecture, for higher resolution feature maps. We use residual connections from the signals from the input, the first QConvBlock, and after the second addition operation. This mitigates the heavy signal degradation from quantization. We choose to use QDepthWiseConv operations for depth-wise filtering to add robustness without the overhead of pointwise convolutions. Figure 3.2b shows the Local Feature Extraction block design of BinaryFaceNet. This module processes VLR feature maps and features more operations than the GFE block. It reduces the representation size by half with a 2-strided convolution at the start of the block and propagates that signal to add it after the second QConvBlock. The signal from the two QSeparableConv layers is added at the end of the block.

3.2.3 Local Feature Extraction Block

The Local Feature Extraction block, illustrated in Figure 3.2b, firstly downsizes the representation with a non-grouped 2-strided convolution. We choose to use Quantized Separable Depthwise Convolution layers (QSeparableConv) instead of the simpler QDepthWiseConv layers from the previous GFE block. The QSeparableConv performs additional pointwise convolutions separating spatially the depth kernel after a QDepthWiseConv, but is less expensive than a regular convolutional layer. After a PReLU activation layer, we use another QConvBlock and we reintroduce the signal from the initial downsampling back to the feature map flow. After re-scaling with PReLU, we further process the feature map by using another QConvBlock followed by a QSeparableConv. Finally, we add the signal from the first QSeparableConv layer back to the network and re-scale for the final block output.

3.2.4 Embedding Block

In our embedding block, detailed in Figure 3.3, we define our face descriptor as a 128-Dimensional vector. A small face descriptor, such as this one, is efficient and adequate, shown to perform adequately in face recognition performance benchmarks [79, 80]. We downscale the representation using two serialized full-precision convolutional blocks, favoring strided convolutions and further avoiding pooling operations, with a kernel size of (3,3) and (4,4), and strides of 3 steps and 1 step, respectively. When using a low-dimension output vector, we are optimizing the computations in the posterior Fully Connected (FC) layers or any other posterior verification and matching calculations.

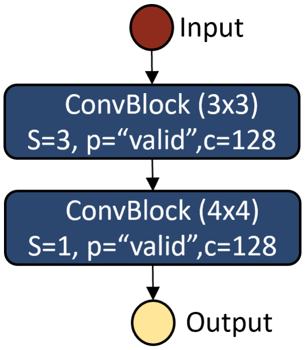


Figure 3.3: The Embedding block consists of two straightforward full-precision Convolution blocks with ”valid” padding, filtering the full area of the feature map with no additional padding. The channel dimension is expanded to 128 at the start of the block and is consistent throughout.

Chapter 4

Evaluations on VLRFR datasets and discussion

In this chapter we present our experiments for validating our hypothesis through our experimentation methodology, we present our results and discuss our findings in terms of limitations on hardware implementations, the trade-off between complexity and accuracy, the particularities of training ultra-compact binarized FR methods, and future work regarding bringing this approach to an application setting.

4.1 Methodology

We performed experiments of our proposed BinaryFaceNet against other state-of-the-art BNNs: QuickNet [4] and BinaryDenseNet [8], with different complexity configurations for BinaryDenseNet (depths 28, 37, and 45). For training, we used the LARQ framework [41] on Keras over Tensorflow. We use the SGDW optimizer with batch sizes of 128 per our GPU memory limitations for any given model. For our hardware settings, we use one NVIDIA GeForce GTX 1080Ti 11GB VRAM GPU for training. The learning rate is set to 0.01 for BinaryFaceNet for stability; the rest of the networks use a 0.1 value. The learning rate is adaptative, multiplied by 0.1 at iterations 9, 15, and 18. The optimizer parameters are set to 0.9 for the momentum and 5e-4 for the weight decay. We use the ArcFace loss function [31] for train and fine-tuning, which is highly optimized for face recognition performance. We selected an ArcFace scaling value of 16 for all BNNs, and tested empirically for stability. We use an input resolution of $112\times112\times3$ and 128-Dimensional embeddings for all BNNs.

4.1.1 Pre-processing and employed dataset overview

As per studies in the VLR FR area [80], mentioned in previous sections, we employ inter-area interpolation to simulate the conditions of VLR images as a single method for interpolation, then up-scaling to the input-layer size of 112×112.

We perform pre-training on the MS1-Celeb-1M dataset [46] and fine-tuning on the CASIA WebFace [136], SCface [42], QMUL-TinyFace [22], and QMUL-SurvFace [23] datasets. All of these datasets, except the CASIA WebFace and SCface, contain VLR images of 32×32

pixels or less natively. We upscale all the input images to 112×112 using inter-area interpolation for all datasets except the CASIA WebFace, for which we use bicubic interpolation as well. The input images are normalized to a [-1, 1] range by subtracting the mean pixel value of 127.5 and dividing by 128.

The SCface dataset contains 130 subjects and 4,160 images from different surveillance camera distances: d1(4.2m), d2(2.6m), and d3(1.0m) with native VLR images and their corresponding subject's HR gallery counterpart. The images at d1 contains face image regions marginally above to 32×32 pixels, making them the most useful in our study.

The CASIA WebFace dataset is an unconstrained face recognition dataset synthetically downscaled to LR images of 7×7, 14×14, 28×28 and 56×56 using bicubic and inter-area subsampling strategies. It contains 10,575 identities and 494,414 images in total.

The QMUL-SurvFace contains native surveillance images from 15,573 subjects and 463,507 images. The input resolutions vary but are always below 32×32 pixels. The QMUL-TinyFace dataset contains unconstrained face images from the web with 5,139 identities and 169,403 images.

For fine-tuning on the QMUL-Tinyface and QMUL-SurvFace datasets, we combined them into a single one by matching the identities. We test the two individual datasets with their own test set under this trained model on the combined dataset.

The MS-Celeb-1M dataset contains a web compilation of over 10 million HR face images from over 100K identities gathered from the web. We used the cleaned version of this dataset (also known as MS1M-V3 or MS1M-RetinaFace), also used as the baseline of the LFR@ICCV19 lightweight face recognition challenge [32].

4.1.2 Performance metrics

In this subsection, we present the metrics we employ for assessing the efficiency performance in our embedded platform, as well as accuracy performance metrics for face identification and face verification for VLR datasets.

4.1.2.1 Efficiency metrics

The presented efficiency metrics are the inference time of our method running the Nvidia Jetson Nano, per our Hardware settings Section 1.7, and the model statistics of our design. We detail these measurements in the following paragraphs.

4.1.2.1.1 Runtime efficiency

We measure the runtime performance on our target hardware using the standard 10W power mode over the Larq Compute Engine benchmarking tool [4] in a single-thread setting. We report the mean inference time results provided by the benchmarking tool, in seconds per image. This is formally defined as:

$$T = \frac{\sum_{i}^{n} t_i}{n} \tag{4.1}$$

Where t_i is the measured inference runtime for one single image and n is the total number of images measured for runtime. The default value for n is 50.

The Frames Per Second (FPS) metric is extrapolated as the number of processed images in one second: $1/T$; where t is the mean runtime in seconds per image of each method, as defined above.

4.1.2.1.2 Model statistics

A k-bit Multiply-Accumulate Count operation (MAC) is the result of multiplying and adding three scalars on the same representation level: $a+(b \times c)$. At an implementation level, the 32-bit MACs and 1-bit MACs are not directly comparable, hence we report the total sum of k-bit MAC operations in our design separately.

The total number of parameters refers to the sum of trainable and non-trainable weights used by the convolutional layers present in the architecture design to compute the next neurons. A regular convolutional layer L_i for a feature map of dimensions $H \times W \times C$ needs $(W_{L_i} \times H_{L_i} \times C_{L_{i-1}} + 1) \times C_{L_i}$ parameters to compute the next neuron. More efficient Depth-Wise layers use less number of parameters, as defined in 2.3. The final calculation is provided by the LARQ framework [41].

The model size reflects the memory needed for storing the weights and model definition of our network using the Keras framework. This memory footprint is reported in MegaBytes(MB), where 8×1024 binary weights = 1KB and 1MB = 1024KB.

4.1.2.2 Accuracy metrics

For accuracy metrics, we define the Face Identification and Face Verification measurements used for the datasets used in our experiments. For matching, we extract the embeddings generated by each method for each image, resulting in a $1 \times 1 \times 128$ descriptor. We score them using the squared difference $S(x_i, y_i) = (F(x_i) - F(y_j))^2$, where F is the Neural Network's feed-forward function.

4.1.2.2.1 Face Identification

The Rank-k identification metric scores the embeddings from a probe p_i from a probe set P and a gallery image set $G = g_1, ..., g_N$, as a 1-to-N matching. We expect the true identity to be present in the top-k closest matches, per the score measurement. This metric is being applied to closed-set and open-set scenarios in the SCface and the QMUL-TinyFace datasets, respectively.

The mean Average Precision (mAP) metric reports the average precision per probe for the Precision-Recall (PR) curve. This curve quantifies The mAP for probe set $P = p_1, ..., p_N$ and Gallery set $G = g_1, ..., g_n$ is defined as [120]:

$$mAP(P, G) = \frac{1}{|P|} \sum_{i}^{|P|} \sum_{k}^{|Q|} \text{Precision}(p_i, g_k) \times [\text{Recall}(p_i, g_k) - \text{Recall}(p_i, g_{k-1})] \tag{4.2}$$

The mAP metric is used on the QMUL-TinyFace for the open-set identification scenario, as defined in its evaluation protocol [22]. This metric is selected due to the evaluation of a precision and recall measurement per probe, as opposed to a proportion of true matches present on a Cumulative Matching Characteristics Curve.

4.1.2.2.2 Face Verification

For face verification, we perform 1-to-1 comparisons to determine if two embeddings belong to the same identity. We employ the False Acceptance Rate (FAR), False Rejection Rate (FRR), and True Acceptance Rate (TAR) as follows:

$$\text{FAR}(t) = \frac{|\{s \geq t, \text{where } s \in U\}|}{|U|} \tag{4.3}$$

$$\text{FRR}(t) = \frac{|\{s \geq t, \text{where } s \in M\}|}{|M|} \tag{4.4}$$

$$\text{TAR}(t) = 1 - \text{FRR}(t) \tag{4.5}$$

Where U is the set of probes with unmatched gallery enrollments (impostors), M is the set of probes with enrolled matching gallery pairs (enrolled identities), s is the matching score, and t is the identity match threshold. These measurements are dependent on the established t threshold. The opposing metrics TAR@FAR at a certain threshold are used to evaluate the face verification performance. This is used to generate a Receiver Operator Characteristics (ROC) curve, where the Area Under the Curve (AUC) measurement is used in this work for the QMUL-SurvFace dataset in the open-set verification setting.

For the LFW simulated HR-to-VLR matching settings, we report the verification accuracy of two compared probes. This verification accuracy rate is represented as the proportion of correctly verified pairs (True Positives and True Negatives) at threshold t with respect of the total number of comparisons made (Positive and Negative total pairs).

4.2 Experiments

In this section, firstly we present the computational complexity of our face recognition model against state-of-the-art BNNs for general-purpose Computer Vision tasks: Quicknet [4] and BinaryDenseNet [8] with depth configurations 28, 37, and 45. Secondly, we show our face recognition results on traditional face recognition benchmarks and VLR-specific datasets using those same BNNs, except for BinaryDenseNet45.

4.2.1 Efficiency performance

In Table 4.1 we report the number of binary Multiply-Accumulate (MAC) operations for 1-bit and Full Precision 32-bit representations, the model size in Megabytes(MB), the total number of parameters and the runtime on the Larq Compute Engine (LCE) [4] benchmark tool in single-threaded performance, running on an NVIDIA Jetson Nano 2Gb with an ARM A57 processor. The inference time benchmark from LCE represents the time it takes for each

method to perform a forward pass under the aforementioned setting.
With our BinaryFaceNet architecture, we perform 57% less binary MACs and 44% less floating-point 32-bit MACs, compared to QuickNet. The 1-bit MAC operations gap is wider against BinaryDenseNet designs: of 79%, 83%, and 88% on network depths 28, 37, and 45, respectively. In terms of model size, our method has a memory requirement of only 3.8MegaBytes, with the rest needing 44% or more space to run. Our design also runs with only 506K parameters, representing only 3.9% of the total parameters for QuickNet (12.7M) and 11% of the total parameters of BinaryDenseNet 28 (4.5M). Most importantly, for inference time per image in a practical setting using LCE, BinaryFaceNet shows a 91% runtime speedup against QuickNet

Method	1-bit MACs	32-bit MACs	Mem. (MB)	Total Params	Time img/s	FPS
BinaryDenseNet45[8]	1.59G	59.7M	4.2	13.1M	6.2s	0.16
BinaryDenseNet37[8]	1.12G	48.9M	2.6	8.0M	4.4s	0.22
BinaryDenseNet28[8]	893.2M	49.99M	1.8	4.5M	3.7s	0.27
QuickNet[4]	431.7M	6.8M	2.21	12.7M	1.8s	0.55
BinaryFaceNet	**184.9M**	**3.8M**	**1.3**	**506K**	**0.16s**	**6.25**

Table 4.1: Efficiency for the state-of-the-art Binary Neural Networks, for a 112×112 input and 128-Dimensional feature output. Our ultra-compact model BinaryFaceNet is the most efficient in design, inference runtime, and Frames Per Second (FPS) in the Larq Compute Engine [4] single-threaded benchmark on the NVIDIA Jetson Nano ARM A57 processor.

The presented efficiency results point to the feasibility of implementing a real-time face recognition application on an affordable embedded device like the NVIDIA Jetson Nano. Our method is the only one capable of achieving 6 to 7 Frames Per Second (FPS) using only one thread of the ARM A57 CPU. In this scenario, we can offload the rest of the application and FR pipeline processing burden to the rest of the ARM A57 CPU cores and the included GPU.

4.2.2 Accuracy performance

In this subsection, we present our VLR FR experiments with the training settings described in our Methodology section 4.1. We show accuracy results for BinaryFaceNet, QuickNet, and BinaryDenseNet in depths 28 and 37. We choose to omit BinaryDenseNet at depth 45 due to the diminishing returns in terms of accuracy with respect to the shallower configurations against the extended training time.

4.2.2.1 SCface

We present the identification results for the SCface dataset [42] according to the protocol used in the state of the art [80, 137, 140], using 50 randomly chosen subject identities for training and 80 for testing. Table 4.2 shows the recognition rates for the three distances of the subjects to the camera, as defined in the dataset (4.2m, 2.6m, and 1.0m). The presented verification results compare the VLR native surveillance image with the HR reference image.

Method	d1 (4.2m)	d2(2.6m)	d3 (1.0m)
BinaryDenseNet-37 [8]	38.0%	79.2%	86.7%
BinaryDenseNet-28 [8]	35.0%	79.2%	89.0%
QuickNet [4]	43.2%	**83.0%**	**91.2%**
BinaryFaceNet (ours)	**48.7%**	70.5%	60.5%

Table 4.2: SCface identification performance using the 50-80 subject testing protocol [80]. Our method outperforms the rest of the BNN methods on the hardest cases (d1 at 4.2m).

For this identification test, BinaryFaceNet performs with a higher accuracy performance on the hardest setting (4.2m) where the camera is the farthest from the subject, compared to the next most efficient state-of-the-art BNNs. The method has higher performance drops on the 2.6m and 1.0 settings, where we start seeing the limitations of our approach. This shows the effectiveness of our block design for processing the very low resolution feature maps and the compromise on capturing higher-density details when balancing the accuracy-efficiency trade-off.

4.2.2.2 QMUL-TinyFace and QMUL-SurvFace

For the combined QMUL-TinyFace and QMUL-SurvFace dataset, we present the identification results for QMUL-TinyFace in Table 4.3 and the verification results for QMUL-SurvFace in Table 4.4.

Method	Rank-1	Rank-20	Rank-50	mAP
BinaryDenseNet-37 [8]	39.1%	56.2%	61.6%	31.3%
BinaryDenseNet-28 [8]	37.5%	56.2%	61.3%	30.0%
QuickNet [4]	**40.9%**	**59.0%**	**64.5%**	**32.7%**
BinaryFaceNet (ours)	22.9%	38.0%	43.5%	16.9%

Table 4.3: QMUL-TinyFace identification experiments using our combined QMUL-TinyFace and QMUL-SurvFace dataset. Our method achieves a consistent performance compromise on real-world datasets against other BNNs while taking less than 10% of the inference time required for those methods.

In the QMUL-TinyFace identification setting (Table 4.3), our method scores an accuracy drop across the Rank-1, Rank-20, and Rank-50 metrics of 18% in the hardest setting and 20% in the other two settings respectively, against QuickNet. This gap is reduced against both BinaryDenseNet approaches in all settings, which are considerably more expensive methods.

Method	TAR@FAR		AUC	Mean
	1%	0.1%		Acc.
BinaryDenseNet-37 [8]	**49.1%**	21.0%	**91.7%**	**84.2%**
BinaryDenseNet-28 [8]	44.7%	**22.1%**	91.3%	83.7%
QuickNet [4]	47.3%	18.1%	91.6%	83.9%
BinaryFaceNet	30.0%	8.3%	87.1%	78.6%

Table 4.4: QMUL-SurvFace verification performance when fine-tuning with the combined QMUL-TinyFace and QMUL-SurvFace dataset. In this scenario, our method has less of a performance compromise comparatively, with only a 9.8% drop for TAR@FAR=0.1% and competitive results on AUC and overall mean accuracy on this real-world surveillance benchmark.

In the QMUL-SurvFace verification results (Table 4.4), our method achieves a similar drop against Quicknet, showing a drop of 17.3% on the TAR@FAR=1% setting and 9.8% on the TAR@FAR=0.1% setting, with this latter one being the harder verification setting. In a Mean Accuracy metric, our method achieves only a 5% drop against the next most efficient approach as well.

4.2.2.3 LFW simulated HR-VLR matching

Running our synthetic dataset benchmarks for simulated VLR verification scenarios, as also presented in [80], we present Table 4.5, showing the verification rates for our simulated HR matching with VLR probe scenario with the CASIA WebFace and SCface datasets, respectively. The VLR probes are images from the LFW dataset downscaled using inter-area interpolation. This experiment aims to test cross-dataset verification performance with the simulated VLR conditions provided by the interpolation methods.

Method	CASIA FT - LFW			SCface FT - LFW	
	7×7	14×14	28×28	7×7	14×14
BinaryDenseNet-37 [8]	**90.2%**	95.3%	**97.7%**	56.6%	**79.3%**
BinaryDenseNet-28 [8]	**90.2%**	95.0%	97.5%	55.3%	78.1%
QuickNet [4]	89.1%	**94.5%**	97.1%	53.9%	75.2%
BinaryFaceNet (ours)	81.6%	89.2%	92.8%	**58.7%**	78.1%

Table 4.5: Verification rates for the simulated HR matching with VLR probes from the LFW dataset using inter-area interpolation, when fine-tuning using the CASIA WebFace and SCface datasets separately. The BinaryFaceNet descriptor achieves a very competitive cross-dataset verification performance, outperforming other BNNs with fine-tuning using SCface, a dataset with native VLR imagery.

Our proposal BinaryFaceNet achieves a 7.5% accuracy compromise on the hardest case against QuickNet when fine-tuning in the CASIA WebFace and a superior verification performance in the same case when fine-tuning using the SCface dataset. This shows a compelling result for the descriptor generated by BinaryFaceNet, especially for verification performance on affordable hardware.

4.3 Discussion and future work

In this section, we discuss the outcomes of our research work in terms of training ultra-compact binarized models, limitations on hardware implementations, and balancing efficiency and accuracy; while providing insights for future research directions. We primarily discuss these topics from a real-time application perspective using affordable hardware.

4.3.1 Limitations on hardware implementations

Computer Vision technology in real-world applications is usually dependent on the target hardware constraints. Our target hardware for this work is the NVIDIA Jetson Nano, an affordable inexpensive device in terms of power efficiency. In theory, BNNs should be extremely efficient [94], however, we are currently constrained by the software frameworks supported for our hardware configuration. In this project, we used the LARQ Compute Engine [4] for its support for state-of-the-art BNN operations such as grouped binary convolutions, DOREFA quantizers, and its highly-efficient custom binary ops. On the other hand, this platform only supports ARM and xCORE architectures. Even though mainstream x86 architectures are affordable and available, binarized convolutional operations are not optimized for these platforms. This also means that embedded platforms with onboard GPU modules cannot be utilized due to the binary framework only working on ARM CPUs. Additional research efforts are needed to broaden the availability of robust BNN technology on more mainstream platforms.

In order to close the gap for a complete VLR FR application to run in such power-efficient devices, we must consider the computational effort of running the full face recognition pipeline: camera image processing, face detection, alignment, feature extraction, and matching. As previously mentioned, BinaryFaceNet is the only one capable of running in a VLR FR application close to a real-time setting (6 to 7 FPS, without overclocking) in a single ARM core with comparable accuracy performance to the state-of-the-art BNNs while reserving the rest of the CPU cores and GPU for the rest of the operations in the FR pipeline. We attribute the inference time speedup of our method to the combination of drastically reducing the number of network parameters and total MAC operation count. Reducing the total parameter count by at least 89%, maintaining most of our network binarized (over 97% of total MAC operations), and reducing the 1-bit MAC count by 57% and the 32-bit MAC count by 44% against other BNNs has allowed us to achieve inference performance closer to real-time scenarios.

4.3.2 Trade-off between complexity and accuracy

The BinaryFaceNet proposal in this paper is particularly optimized to our hardware constraints and VLRFR scenario. The face descriptor generated by BinaryFaceNet is a step towards a more efficient representation for a cross-dataset verification scenario. The BinaryFaceNet architecture can also benefit from more hardware resources available by tuning network hyperparameters; particularly the internal depth size and the quantization bits. The quantization bit selection from the reduction block inside the head block (Figure 3.1) is one of the primary points for balancing for efficiency and accuracy performance. Our quantizer choice, DOREFA [149], also allows for an easy k-bit adjustment. Balancing this quantized k-bit representation can lead to a more robust representation since it directly affects the calculation of the SE weight that scales the input image after the first full-precision convolution. This opens the door for efficient face recognition research using this BNN design in other scenarios such as heterogeneous FR (sketches, NIR-VIS, etc.), aging subjects, and racial bias. In our own design experiments, we achieved around a 10% accuracy improvement in the aging subjects AgeDB30 verification benchmark when using 32-bit FP layers in the reduction block.

The practical scenarios enabled by the specific configuration present in this method proposal are VLR FR applications on extremely-limited ARM-based processors. We can expect real-time performance for recognizing one face image per ARM-CPU core available. The accuracy of our method does not have a significant negative in face verification scenarios with VLR probes and HR images, as shown in our previous section. To mitigate the accuracy drops in face recognition scenarios, other training methodologies, such as Knowledge Distillation, can be used, where the efficient design of the configuration proposed in this work can be leveraged for learning to output embeddings closer to other more complex VLR FR methods without sacrificing efficiency. Combining the finetuning of network hyperparameters with more complex methodologies shall heavily reduce the accuracy shortcomings of this method wihle taking advantage of our highly-efficient design for real-time harware-constrained applications.

4.3.3 Training ultra-compact face recognition binarized stable models

In our experiments, we found several hyperparameters and training settings to appropriately work on the face recognition scenario. The recommended state-of-the-art optimizer for training BNNs using general-purpose datasets is the Adam optimizer [78], however, in our experiments with face recognition datasets we found this strategy to be ineffective when optimizing the state-of-the-art FR loss function ArcFace [31]. As such, we employed the SGDW and achieved the reported performance in the previous section. Furthermore, the ArcFace loss scale hyperparameter s was adjusted to 16 from its usual empirical value set to 64. The scale hyperparameter affects the vector scaling in the projected subspace, as such, a larger value leads to higher gradients. This, in turn, also leads to more instability when training BNNs. In the same fashion, the training stability is sensitive to the learning rate hyperparameter. With reduced network designs, such as the case in BinaryFaceNet, the learning rate must be reduced to avoid exploding gradients. The quantizer choice is also important in the design of ultra-compact networks since it influences training stability and accuracy performance. The DOREFA quantizer leverages non-linear functions ($\tanh$) for scaling and gradient clipping, making it an appropriate choice for our ultra-compact model. Further research in hyperparameter tuning for FR scenarios using ultra-compact BNNs is encouraged to further maximize the efficiency benefits of BNN technology.

4.3.4 Future work

In future work with this ultra-compact binarized design, we intend to increase the accuracy of this method without sacrificing much efficiency. We could do this by leveraging other k-bit quantization schemes, such as 8-bit weights and 6-bit activations for the reduction block inside our head block design. Recent work on quantization rates and Knowledge Distillation [14] suggest a viable approach for a specific application scenario. For dataset-specific scenarios, there is an opportunity to propose more effective pre-processing techniques for VLR FR applications. Research on jointly trained shallow binarized Super Resolution network for this scenario could be a viable approach for scenarios with more permissive hardware. Allocating the face detection and normalization resources into an ensemble shallow network for face normalization may improve performance on VLR FR applications. Another aspect to further expand the applicability of this work is biometric face sample encryption for privacy. Our previous work [53] shows that it is feasible for a network to train to recognize encrypted face samples. We intend to explore the feasibility of improving this privacy aspect on our real-time embedded hardware-constrained settings.

Chapter 5

Conclusions and Research Outcomes

In this section, we provide the final remarks of our research work and the outcomes achieved within the duration of this thesis project.

5.1 Conclusions

For the duration of this thesis work, we mainly reviewed the Lightweight CNN for FR and Very Low Resolution Face Recognition literature, proposed a taxonomy for organizing the state-of-the-art Heterogeneous FR and Homogenous FR approaches, analyzed the shortcomings of the most efficient CNN and BNN approaches for real-time VLR FR scenarios using embedded platforms, proposed a novel BNN architecture for dramatically improving efficiency this scenario, and explored possibilities for expanding the applicability of our work regarding biometric face sample encryption.

Firstly, we tested the Lightweight CNNs for FR in the Lightweight Face Recognition challenge at ICCV 2019, proposing ShuffleFaceNet. This work allowed us to focus on improving efficiency with competent accuracy performance and the particularities of Neural Network architectural components for achieving competent FR performance. For our VLR FR state-of-the-art taxonomy, we divided the existing works into Homogeneous and Heterogeneous approaches, if any method bridges the domain gap in its design. For our analysis, we tested the viability of Lightweight CNN as Homogeneous FR approaches on VLR benchmark datasets with fine-tuning using different interpolation methods to better simulate VLR conditions as a pre-processing step. We also tested runtime performance in various x86 architectures using, CPU and GPU for inference.

After achieving encouraging results with our proposed work ShuffleFaceNet running in real-time using a Laptop CPU, we proceeded to review the Binary Neural Network approaches, to bring VLR FR to an embedding application scenario. With our Binary Neural Network background work analysis, we could find the areas of opportunity in the area, mainly stemming from information degradation and not using the improvements present in Lightweight CNNs for FR approaches.

We proceeded to design our binarized architecture for Very Low Resolution Face Recognition, using other BNNs in the state of the art as baselines. We set our training settings standardized with the FR literature and only modify the optimizer hyperparameters when facing limitations of learning stability. After fine-tuning our architecture design for extreme efficiency and comparable accuracy for VLRFR performance, we achieved a dramatically low parameter count and Multiply Accumulate Count operations. This finally resulted in the only binarized method running in real-time on a single thread of the NVIDIA Jetson Nano ARM A57 processor. Considering more aspects for bringing this application to a real-world setting as future work, we explored the face recognition for biometric face encryption avenue, with encouraging results. Likewise, we could expand this proposal to various degrees of efficiency-accuracy tradeoffs, depending on the hardware resources available, but still prioritizing efficiency. Further work also includes more complex training methodologies relying on knowledge distillation, where efficient architectures can approximate embeddings generated by heavier methods.

5.2 Final research outcomes

During this thesis work and per our objectives, we made the following contributions to the state-of-the-art:

- One thesis work on efficient Very Low Resolution Face Recognition for affordable embedded hardware: the NVIDIA Jetson Nano.
- One GitHub public repository: adapted training and deployment framework for Binary-FaceNet and other BNNs on Keras/Tensorflow: `https://github.com/lluevano/insightface_larq_keras`
- Three JCR journal papers:
  - One survey paper proposing a taxonomy for the Very Low Resolution Face Recognition literature and discussing these approaches from an application standpoint: LUEVANO, L. S., CHANG, L., MÉNDEZ-VÁZQUEZ, H., MARTÍNEZ-DÍAZ, Y., AND GONZÁLEZ-MENDOZA, M. A study on the performance of unconstrained Very Low Resolution Face Recognition: Analyzing current trends and new research directions. *IEEE Access* 9 (2021), 75470–75493
  - One Lightweight CNNs for FR benchmark study, evaluating different High Resolution Face Recognition scenarios:
    MARTÍNEZ-DÍAZ, Y., NICOLÁS-DÍAZ, MÉNDEZ-VÁZQUEZ, H., LUEVANO, L. S., CHANG, L., GONZÁLEZ-MENDOZA, M., AND SUCAR, L. Benchmarking lightweight face architectures on specific face recognition scenarios. *Artificial Intelligence Review* (02 2021).
  - One Lightweight CNNs for FR benchmark on Masked FR scenarios for evaluating robustness on the latest methods:
    Y. MARTÍNEZ-DÍAZ, H. MÉNDEZ-VÁZQUEZ, L. S. LUEVANO, M. NICOLÁS-DÍAZ, L. CHANG, AND M. GONZÁLEZ-MENDOZA. Towards Accurate and Lightweight Masked Face Recognition: An Experimental Evaluation. In *IEEE Access* 10 (2022) 7341-7353

- Three conference papers:
  - One Lightweight CNN proposal, improving on inference runtime performance:
    MARTÍNEZ-DÍAZ, Y., LUEVANO, L. S., MÉNDEZ-VÁZQUEZ, H., NICOLÁS-DÍAZ, M., CHANG, L., AND GONZÁLEZ-MENDOZA, M. Shufflefacenet: A lightweight face architecture for efficient and highly-accurate face recognition. In *The IEEE International Conference on Computer Vision (ICCV) Workshops* (Oct 2019).
  - One interpolation method evaluation for VLRFR using Lightweight CNNs for FR:
    MARTÍNEZ-DÍAZ, Y., MÉNDEZ-VÁZQUEZ, H., LUEVANO, L. S., CHANG, L., AND GONZÁLEZ-MENDOZA, M. Lightweight low-resolution face recognition for surveillance applications. In *2020 25th International Conference on Pattern Recognition (ICPR)* (2021), pp. 5421–5428
  - One Biometric Face Encryption using CNNs proposal, for expanding our work for a real-world application:
    HOFBAUER, H., MARTÍNEZ-DÍAZ, Y., LUEVANO, L. S., MÉNDEZ-VÁZQUEZ, H., UHL, A. Utilizing CNNs for Cryptanalysis of Selective Biometric Face Sample Encryption. In *Proceedings of the 26th International Conference on Pattern Recognition (ICPR)* (2022), p. 8.
- One manuscript in the process of submission to a JCR Q1 journal: BinaryFaceNet: A Binarized Approach for Real-Time Very Low Resolution Face Recognition in Video Surveillance Scenarios
- One manuscript draft in progress, from the experimentation work done at the research stay at the Idiap Research Institute: Synthesis of Very Low Resolution face images usable on Very Low Resolution Face Recognition scenarios.

Appendix A

BinaryFaceNet detailed MAC output per layer

Block	Layer	Output size	Memory	1-bit	32-bit
			(KBs)	MACs	MACs
	input-1	(-1, 112, 112, 3)	0	-	-
Head block	conv2d	(-1, 112, 112, 64)	0.75	0	2408448
	batch-normalization	(-1, 112, 112, 64)	0.50	0	0
	p-re-lu	(-1, 112, 112, 64)	0.25	-	-
	quant-conv2d	(-1, 54, 54, 64)	3.12	74649600	0
	batch-normalization-1	(-1, 54, 54, 64)	0.50	0	0
	p-re-lu-1	(-1, 54, 54, 64)	0.25	-	-
	quant-conv2d-1	(-1, 18, 18, 64)	1.12	2985984	0
	batch-normalization-2	(-1, 18, 18, 64)	0.50	0	0
	p-re-lu-2	(-1, 18, 18, 64)	0.25	-	-
	quant-conv2d-2	(-1, 5, 5, 64)	0.25	51200	0
	batch-normalization-3	(-1, 5, 5, 64)	0.50	0	0
	p-re-lu-3	(-1, 5, 5, 64)	0.25	-	-
	quant-conv2d-3	(-1, 1, 1, 64)	0.50	4096	0
	batch-normalization-4	(-1, 1, 1, 64)	0.50	0	0
	p-re-lu-4	(-1, 1, 1, 64)	0.25	-	-
	reshape	(-1, 1, 1, 64)	0	-	-
	conv2d-1	(-1, 1, 1, 4)	1.02	0	256
	p-re-lu-5	(-1, 1, 1, 4)	0.02	-	-
	conv2d-2	(-1, 1, 1, 64)	1.25	0	256
	multiply	(-1, 112, 112, 64)	0	-	-
	quant-conv2d-4	(-1, 56, 56, 64)	1.12	28901376	0
GFE block 1	batch-normalization-5	(-1, 56, 56, 64)	0.50	0	0
	p-re-lu-6	(-1, 56, 56, 64)	0.25	-	-
	quant-conv2d-5	(-1, 56, 56, 64)	0.75	12845056	0
	p-re-lu-7	(-1, 56, 56, 64)	0.25	-	-

GFE block 1	quant-depthwise-conv2d	(-1, 56, 56, 64)	0.26	200704	0
	add	(-1, 56, 56, 64)	0	-	-
	p-re-lu-8	(-1, 56, 56, 64)	0.25	-	-
	quant-conv2d-6	(-1, 56, 56, 64)	0.75	12845056	0
	add-1	(-1, 56, 56, 64)	0	-	-
	p-re-lu-9	(-1, 56, 56, 64)	0.25	-	-
	quant-depthwise-conv2d-1	(-1, 56, 56, 64)	0.26	200704	0
	add-2	(-1, 56, 56, 64)	0	-	-
	p-re-lu-10	(-1, 56, 56, 64)	0.25	-	-
LFE block 1	quant-conv2d-7	(-1, 28, 28, 64)	4.50	28901376	0
	batch-normalization-6	(-1, 28, 28, 64)	0.50	0	0
	p-re-lu-11	(-1, 28, 28, 64)	0.25	-	-
	quant-separable-conv2d	(-1, 28, 28, 64)	0.76	3261440	0
	p-re-lu-12	(-1, 28, 28, 64)	0.25	-	-
	quant-conv2d-8	(-1, 28, 28, 64)	0.50	3211264	0
	batch-normalization-7	(-1, 28, 28, 64)	0.50	0	0
	add-3	(-1, 28, 28, 64)	0	-	-
	p-re-lu-13	(-1, 28, 28, 64)	0.25	-	-
	quant-conv2d-9	(-1, 28, 28, 64)	0.50	3211264	0
	batch-normalization-8	(-1, 28, 28, 64)	0.50	0	0
	p-re-lu-14	(-1, 28, 28, 64)	0.25	-	-
	quant-separable-conv2d-1	(-1, 28, 28, 64)	0.76	3261440	0
	add-4	(-1, 28, 28, 64)	0	-	-
	p-re-lu-15	(-1, 28, 28, 64)	0.25	-	-
LFE block 2	quant-conv2d-10	(-1, 14, 14, 64)	4.50	7225344	0
	batch-normalization-9	(-1, 14, 14, 64)	0.50	0	0
	p-re-lu-16	(-1, 14, 14, 64)	0.25	-	-
	quant-separable-conv2d-2	(-1, 14, 14, 64)	0.76	815360	0
	p-re-lu-17	(-1, 14, 14, 64)	0.25	-	-
	quant-conv2d-11	(-1, 14, 14, 64)	0.50	802816	0
	batch-normalization-10	(-1, 14, 14, 64)	0.50	0	0
	add-5	(-1, 14, 14, 64)	0	-	-
	p-re-lu-18	(-1, 14, 14, 64)	0.25	-	-
	quant-conv2d-12	(-1, 14, 14, 64)	0.50	802816	0
	batch-normalization-11	(-1, 14, 14, 64)	0.50	0	0
	p-re-lu-19	(-1, 14, 14, 64)	0.25	-	-
	quant-separable-conv2d-3	(-1, 14, 14, 64)	0.76	815360	0
	add-6	(-1, 14, 14, 64)	0	-	-
	p-re-lu-20	(-1, 14, 14, 64)	0.25	-	-
Emb block	conv2d-3	(-1, 4, 4, 128)	288.00	0	1179648
	batch-normalization-12	(-1, 4, 4, 128)	1.00	0	0
	p-re-lu-21	(-1, 4, 4, 128)	0.50	-	-
	conv2d-4	(-1, 1, 1, 128)	1024.00	0	262144
	batch-normalization-13	(-1, 1, 1, 128)	1.00	0	0

	p-re-lu-22	(-1, 1, 1, 128)	0.50	-	-

Table A.1: BinaryFaceNet MAC count per convolutional layer

Total params	506 K
Trainable params	504 K
Non-trainable params	2.05 K
Model size	1.32 MB
Float-32 Equivalent	1.93 MB
Compression Ratio of Memory	0.68
Number of MACs	189 M
Ratio of MACs that are binarized	0.9796

Table A.2: BinaryFaceNet architecture summary per the summary tool of the LARQ BNN framework.

Appendix B

Synthetic VLRFR and HRFR verification rates

Method	SCFace-FT				TinyFace+SurvFace-FT				CASIA LR-interArea-FT		
	7×7	14×14	hr2lr-7	hr2lr-14	7×7	14×14	21×21	28×28	7×7	14×14	28×28
BinaryDenseNet-37 [8]	63.48%	84.83%	56.67%	**79.30%**	**68.50%**	74.10%	78.23%	81.95%	**90.25%**	**95.37%**	**97.73%**
BinaryDenseNet-28 [8]	61.63%	**84.97%**	55.30%	78.13%	68.37%	72.98%	73.15%	75.80%	90.20%	95.03%	97.53%
QuickNet [4]	61.85%	85.55%	53.02%	75.27%	67.70%	74.45%	77.38%	79.32%	89.15%	94.57%	97.17%
BinaryFaceNet (ours)	**63.63%**	80.35%	**58.75%**	78.18%	67.53%	**75.55%**	**82.48%**	**84.93%**	81.65%	89.27%	92.80%

Table B.1: Complete Synthetic VLR FR verification benchmarks for Binary Networks downsampled probes for LFW with VLR-to-VLR matching and HR-to-VLR matching.

Method	Type	ms-celeb-1M train		
		LFW	CFP-FP	AGEDB-30
VarGFaceNet [132]	L.CNN	99.73%	97.67%	97.5%
MobileFaceNet [20]	L.CNN	97.33%	99.71%	97.56%
ShuffleFaceNet [79]	L.CNN	97.38%	97.25%	97.31%
BinaryDenseNet-45 [8]	BNN	99.28%	92.88%	91.03%
BinaryDenseNet-37 [8]	BNN	99.17%	92.59%	90.72%
BinaryDenseNet-28 [8]	BNN	99.17%	92.11%	90.72%
QuickNet [4]	BNN	98.97%	92.00%	89.00%
BinaryFaceNet (ours)	BNN	95.07%	77.93%	75.12%

Table B.2: HRFR verification rates for Lightweight CNNs and BNNs using the insightface verification verification benchmarks [32]

Bibliography

[1] AN, L., AND BHANU, B. Face image super-resolution using 2d cca. *Signal Processing 103* (2014), 184 – 194. Image Restoration and Enhancement: Recent Advances and Applications.

[2] ANGULO, A., VEGA-FERNÁNDEZ, J. A., AGUILAR-LOBO, L. M., NATRAJ, S., AND OCHOA-RUIZ, G. Road damage detection acquisition system based on deep neural networks for physical asset management. In *Advances in Soft Computing* (Cham, 2019), L. Martínez-Villaseñor, I. Batyrshin, and A. Marín-Hernández, Eds., Springer International Publishing, pp. 3–14.

[3] ARTEAGA BOTELLO, N. Regulación de la videovigilancia en México. Gestión de la ciudadanía y acceso a la ciudad. *Espiral (Guadalajara) 23* (08 2016), 193 – 238.

[4] BANNINK, T., BAKHTIARI, A., HILLIER, A., GEIGER, L., DE BRUIN, T., OVERWEEL, L., NEEVEN, J., AND HELWEGEN, K. Larq compute engine: Design, benchmark, and deploy state-of-the-art binarized neural networks, 2020.

[5] BANSAL, A., NANDURI, A., CASTILLO, C. D., RANJAN, R., AND CHELLAPPA, R. Umdfaces: An annotated face dataset for training deep networks. *2017 IEEE International Joint Conference on Biometrics (IJCB)* (Oct 2017).

[6] BAY, H., ESS, A., TUYTELAARS, T., AND VAN GOOL, L. Speeded-up robust features (surf). *Comput. Vis. Image Underst. 110*, 3 (June 2008), 346–359.

[7] BELHUMEUR, P. N., HESPANHA, J. P., AND KRIEGMAN, D. J. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence 19*, 7 (1997), 711–720.

[8] BETHGE, J., YANG, H., BORNSTEIN, M., AND MEINEL, C. Back to Simplicity: How to Train Accurate BNNs from Scratch?, jun 2019.

[9] BEVERIDGE, J. R., PHILLIPS, P. J., BOLME, D. S., DRAPER, B. A., GIVENS, G. H., LUI, Y. M., TELI, M. N., ZHANG, H., SCRUGGS, W. T., BOWYER, K. W., FLYNN, P. J., AND CHENG, S. The challenge of face recognition from digital point-and-shoot cameras. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)* (Sep. 2013), pp. 1–8.

[10] Biswas, S., Aggarwal, G., Flynn, P. J., and Bowyer, K. W. Pose-robust recognition of low-resolution face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 12 (Dec 2013), 3037–3049.

[11] Biswas, S., Bowyer, K. W., and Flynn, P. J. Multidimensional scaling for matching low-resolution face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 10 (Oct 2012), 2019–2030.

[12] Boom, B. J., Beumer, G. M., Spreeuwers, L. J., and Veldhuis, R. N. J. The effect of image resolution on the performance of a face recognition system. In *2006 9th International Conference on Control, Automation, Robotics and Vision* (Dec 2006), pp. 1–6.

[13] Boutros, F., Damer, N., Fang, M., Kirchbuchner, F., and Kuijper, A. Mixfacenets: Extremely efficient face recognition networks, 2021.

[14] Boutros, F., Damer, N., and Kuijper, A. Quantface: Towards lightweight face recognition by synthetic data low-bit quantization. In *26th International Conference on Pattern Recognition, ICPR 2022, Montreal,Quebec ,August 21-25, 2021* (2022), IEEE.

[15] Bulat, A., and Tzimiropoulos, G. Xnor-net++: Improved binary neural networks, 2019.

[16] Cao, J., Su, Z., Yu, L., Chang, D., Li, X., and Ma, Z. Softmax cross entropy loss with unbiased decision boundary for image classification. In *2018 Chinese Automation Congress (CAC)* (2018), pp. 2028–2032.

[17] Cao, Z., Yin, Q., Tang, X., and Sun, J. Face recognition with learning-based descriptor. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (June 2010), pp. 2707–2714.

[18] Changtao Zhou, Zhiwei Zhang, Dong Yi, Lei, Z., and Li, S. Z. Low-resolution face recognition via simultaneous discriminant analysis. In *2011 International Joint Conference on Biometrics (IJCB)* (2011), pp. 1–6.

[19] Chechik, G., Sharma, V., Shalit, U., and Bengio, S. Large scale online learning of image similarity through ranking. *J. Mach. Learn. Res. 11* (Mar. 2010), 1109–1135.

[20] Chen, S., Liu, Y., Gao, X., and Han, Z. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. *Lecture Notes in Computer Science* (2018), 428–438.

[21] Chen, S., Liu, Y., Gao, X., and Han, Z. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices, 2018.

[22] Cheng, Z., Zhu, X., and Gong, S. Low-Resolution Face Recognition. *arXiv preprint arXiv:1811.08965* (nov 2018), 1–16.

[23] CHENG, Z., ZHU, X., AND GONG, S. Surveillance face recognition challenge. *arXiv preprint arXiv:1804.09691* (2018).

[24] CHENGJUN LIU, AND WECHSLER, H. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image Processing 11*, 4 (April 2002), 467–476.

[25] CHU, Y., AHMAD, T., BEBIS, G., AND ZHAO, L. Low-resolution face recognition with single sample per person. *Signal Processing 141* (2017), 144 – 157.

[26] COURBARIAUX, M., AND BENGIO, Y. Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1. *CoRR abs/1602.02830* (2016).

[27] COURBARIAUX, M., BENGIO, Y., AND DAVID, J. Binaryconnect: Training deep neural networks with binary weights during propagations. *CoRR abs/1511.00363* (2015).

[28] DAN ZENG, HU CHEN, AND QIJUN ZHAO. Towards resolution invariant face recognition in uncontrolled scenarios. In *2016 International Conference on Biometrics (ICB)* (2016), pp. 1–8.

[29] DENG, J., GUO, J., AN, X., ZHU, Z., AND ZAFEIRIOU, S. Masked face recognition challenge: The insightface track report. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)* (2021), pp. 1437–1444.

[30] DENG, J., GUO, J., VERVERAS, E., KOTSIA, I., AND ZAFEIRIOU, S. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020).

[31] DENG, J., GUO, J., XUE, N., AND ZAFEIRIOU, S. Arcface: Additive angular margin loss for deep face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019).

[32] DENG, J., GUO, J., ZHANG, D., DENG, Y., LU, X., AND SHI, S. Lightweight face recognition challenge. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops* (Oct 2019).

[33] DING, C., AND TAO, D. Trunk-branch ensemble convolutional neural networks for video-based face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 40*, 4 (Apr 2018), 1002–1014.

[34] DONG-CHEN HE, AND LI WANG. Texture unit, texture spectrum, and texture analysis. *IEEE Transactions on Geoscience and Remote Sensing 28*, 4 (July 1990), 509–512.

[35] FROMM, J., PATEL, S., AND PHILIPOSE, M. Heterogeneous bitwidth binarization in convolutional neural networks, 2018.

[36] FURLÁN, F., RUBIO, E., SOSSA, H., AND PONCE, V. Rock detection in a mars-like environment using a cnn. In *Pattern Recognition* (Cham, 2019), J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, J. A. Olvera-López, and J. Salas, Eds., Springer International Publishing, pp. 149–158.

[37] GANG, R., LIU, S., LI, C., AND SONG, R. Assr: Lightweight super resolution network with aggregative structure. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (2019).

[38] GAO, W., CAO, B., SHAN, S., CHEN, X., ZHOU, D., ZHANG, X., AND ZHAO, D. The cas-peal large-scale chinese face database and baseline evaluations. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 38*, 1 (2008), 149–161.

[39] GATYS, L. A., ECKER, A. S., AND BETHGE, M. Image style transfer using convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016).

[40] GE, S., ZHAO, S., LI, C., AND LI, J. Low-resolution face recognition in the wild via selective knowledge distillation. *IEEE Transactions on Image Processing 28*, 4 (Apr 2019), 2051–2062.

[41] GEIGER, L., AND TEAM, P. Larq: An open-source library for training binarized neural networks. *Journal of Open Source Software 5*, 45 (Jan. 2020), 1746.

[42] GRGIC, M., DELAC, K., AND GRGIC, S. Scface - surveillance cameras face database. *Multimedia Tools Appl. 51* (02 2011), 863–879.

[43] GRM, K., SCHEIRER, W. J., AND ŠTRUC, V. Face hallucination using cascaded super-resolution and identity priors. *IEEE Transactions on Image Processing 29* (2020), 2150–2165.

[44] GROSS, R., MATTHEWS, I., COHN, J., KANADE, T., AND BAKER, S. Multi-pie. *Image Vision Comput. 28*, 5 (May 2010), 807–813.

[45] GUO, Y., ZHANG, L., HU, Y., HE, X., AND GAO, J. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV* (2016).

[46] GUO, Y., ZHANG, L., HU, Y., HE, X., AND GAO, J. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *Computer Vision – ECCV 2016* (Cham, 2016), B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Springer International Publishing, p. 87–102.

[47] HAGHIGHAT, M., AND ABDEL-MOTTALEB, M. Low resolution face recognition in surveillance systems using discriminant correlation analysis. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)* (May 2017), pp. 912–917.

[48] HE, K., ZHANG, X., REN, S., AND SUN, J. Deep residual learning for image recognition, 2015.

[49] HENNINGS-YEOMANS, P. H., BAKER, S., AND KUMAR, B. V. K. V. Simultaneous super-resolution and feature extraction for recognition of low-resolution faces. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (2008), pp. 1–8.

[50] HERRMANN, C., WILLERSINN, D., AND BEYERER, J. Low-resolution convolutional neural networks for video face recognition. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (Aug 2016), pp. 221–227.

[51] HINTON, G., VINYALS, O., AND DEAN, J. Distilling the knowledge in a neural network, 2015.

[52] HINTON, G. E., KRIZHEVSKY, A., AND WANG, S. D. Transforming auto-encoders. In *Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I* (Berlin, Heidelberg, 2011), ICANN'11, Springer-Verlag, pp. 44–51.

[53] HOFBAUER, H., MARTÍNEZ-DÍAZ, Y., LUEVANO, L. S., MÉNDEZ-VÁZQUEZ, H., AND UHL, A. Utilizing cnns for cryptanalysis of selective biometric face sample encryption. In *Proceedings of the 26th International Conference on Pattern Recognition (ICPR)* (2022), p. 8.

[54] HOWARD, A., SANDLER, M., CHU, G., CHEN, L.-C., CHEN, B., TAN, M., WANG, W., ZHU, Y., PANG, R., VASUDEVAN, V., LE, Q. V., AND ADAM, H. Searching for mobilenetv3. In *The IEEE International Conference on Computer Vision (ICCV)* (October 2019).

[55] HOWARD, A. G., ZHU, M., CHEN, B., KALENICHENKO, D., WANG, W., WEYAND, T., ANDREETTO, M., AND ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *ArXiv abs/1704.04861* (2017).

[56] HUANG, G. B., RAMESH, M., BERG, T., AND LEARNED-MILLER, E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[57] IANDOLA, F. N., MOSKEWICZ, M. W., ASHRAF, K., HAN, S., DALLY, W. J., AND KEUTZER, K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡1mb model size. *ArXiv abs/1602.07360* (2017).

[58] JACOB, B., KLIGYS, S., CHEN, B., ZHU, M., TANG, M., HOWARD, A., ADAM, H., AND KALENICHENKO, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018).

[59] KALKA, N. D., MAZE, B., DUNCAN, J. A., ORCONNOR, K., ELLIOTT, S., HEBERT, K., BRYAN, J., AND JAIN, A. K. Ijb-s: Iarpa janus surveillance video benchmark. *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)* (2018), 1–9.

[60] KEMELMACHER-SHLIZERMAN, I., SEITZ, S. M., MILLER, D., AND BROSSARD, E. The megaface benchmark: 1 million faces for recognition at scale. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), pp. 4873–4882.

[61] KLARE, B., KLEIN, B., TABORSKY, E., BLANTON, A., CHENEY, J., ALLEN, K. E., GROTHER, P., MAH, A., BURGE, M., AND JAIN, A. K. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), 1931–1939.

[62] KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (USA, 2012), NIPS'12, Curran Associates Inc., pp. 1097–1105.

[63] LAI, W.-S., HUANG, J.-B., AHUJA, N., AND YANG, M.-H. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence 41*, 11 (Nov 2019), 2599–2613.

[64] LEDIG, C., THEIS, L., HUSZÁR, F., CABALLERO, J., CUNNINGHAM, A., ACOSTA, A., AITKEN, A., TEJANI, A., TOTZ, J., WANG, Z., AND SHI, W. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 105–114.

[65] LI, B., CHANG, H., SHAN, S., AND CHEN, X. Low-resolution face recognition via coupled locality preserving mappings. *IEEE Signal Processing Letters 17*, 1 (Jan 2010), 20–23.

[66] LI, J., ZHANG, C., HU, J., AND DENG, W. Blur-robust face recognition via transformation learning. In *Computer Vision - ACCV 2014 Workshops* (Cham, 2015), C. V. Jawahar and S. Shan, Eds., Springer International Publishing, pp. 15–29.

[67] LI, P., PRIETO, L., MERY, D., AND FLYNN, P. J. On Low-Resolution Face Recognition in the Wild: Comparisons and New Techniques. *IEEE Transactions on Information Forensics and Security PP*, c (2019), 1–1.

[68] LI, S. Z., AND JAIN, A. K. *Handbook of Face Recognition*, 2nd ed. Springer Publishing Company, Incorporated, 2011.

[69] LIN, X., ZHAO, C., AND PAN, W. Towards accurate binary convolutional neural network, 2017.

[70] LIU, W., WEN, Y., YU, Z., LI, M., RAJ, B., AND SONG, L. Sphereface: Deep hypersphere embedding for face recognition. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 6738–6746.

[71] LIU, Z., LUO, P., WANG, X., AND TANG, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)* (December 2015).

[72] LIU, Z., WU, B., LUO, W., YANG, X., LIU, W., AND CHENG, K.-T. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm, 2018.

[73] LOWE, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60*, 2 (Nov. 2004), 91–110.

[74] LU, Z., JIANG, X., AND KOT, A. Deep Coupled ResNet for Low-Resolution Face Recognition. *IEEE Signal Processing Letters 25*, 4 (2018), 526–530.

[75] LUEVANO, L. S., CHANG, L., MÉNDEZ-VÁZQUEZ, H., MARTÍNEZ-DÍAZ, Y., AND GONZÁLEZ-MENDOZA, M. A study on the performance of unconstrained very low resolution face recognition: Analyzing current trends and new research directions. *IEEE Access 9* (2021), 75470–75493.

[76] MA, N., ZHANG, X., ZHENG, H.-T., AND SUN, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Computer Vision – ECCV 2018* (Cham, 2018), V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Springer International Publishing, pp. 122–138.

[77] MA, Y., XIONG, H., HU, Z., AND MA, L. Efficient super resolution using binarized neural network. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2019), pp. 694–703.

[78] MARTINEZ, B., YANG, J., BULAT, A., AND TZIMIROPOULOS, G. Training binary neural networks with real-to-binary convolutions, 2020.

[79] MARTINEZ-DIAZ, Y., LUEVANO, L. S., MENDEZ-VAZQUEZ, H., NICOLAS-DIAZ, M., CHANG, L., AND GONZALEZ-MENDOZA, M. Shufflefacenet: A lightweight face architecture for efficient and highly-accurate face recognition. In *The IEEE International Conference on Computer Vision (ICCV) Workshops* (Oct 2019).

[80] MARTÍNEZ-DÍAZ, Y., MÉNDEZ-VÁZQUEZ, H., LUEVANO, L. S., CHANG, L., AND GONZALEZ-MENDOZA, M. Lightweight low-resolution face recognition for surveillance applications. In *2020 25th International Conference on Pattern Recognition (ICPR)* (2021), pp. 5421–5428.

[81] MARTÍNEZ-DÍAZ, Y., NICOLÁS-DÍAZ, M., VAZQUEZ, H., LUÉVANO GARCÍA, L., CHANG, L., GONZALEZ-MENDOZA, M., AND SUCAR, L. Benchmarking lightweight face architectures on specific face recognition scenarios. *Artificial Intelligence Review* (02 2021).

[82] MAZE, B., ADAMS, J., DUNCAN, J. A., KALKA, N., MILLER, T., OTTO, C., JAIN, A. K., NIGGEL, W. T., ANDERSON, J., CHENEY, J., AND GROTHER, P. Iarpa janus benchmark - c: Face dataset and protocol. In *2018 International Conference on Biometrics (ICB)* (2018), pp. 158–165.

[83] MORENO PÉREZ, S., AND HERNÁNDEZ OLASCOAGA, S. Un panorama de la videovigilancia en méxico. *En Contexto* (2018).

[84] MUDUNURI, S. P., SANYAL, S., AND BISWAS, S. Genlr-net: Deep framework for very low resolution face and object recognition with generalization to unseen categories. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2018), pp. 602–60209.

[85] NECH, A., AND KEMELMACHER-SHLIZERMAN, I. Level playing field for million scale face recognition. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 3406–3415.

[86] NISHIYAMA, M., HADID, A., TAKESHIMA, H., SHOTTON, J., KOZAKAYA, T., AND YAMAGUCHI, O. Facial deblur inference using subspace analysis for recognition of blurred faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 4 (April 2011), 838–845.

[87] NIU, B., YANG, Q., SHIU, S. C. K., AND PAL, S. K. Two-dimensional laplacianfaces method for face recognition. *Pattern Recognition 41*, 10 (2008), 3237 – 3243.

[88] PARKHI, O. M., VEDALDI, A., AND ZISSERMAN, A. Deep face recognition. In *Proceedings of the British Machine Vision Conference (BMVC)* (September 2015), X. Xie, M. W. Jones, and G. K. L. Tam, Eds., BMVA Press, pp. 41.1–41.12.

[89] PEREIRA, S., PINTO, A., ALVES, V., AND SILVA, C. A. Brain tumor segmentation using convolutional neural networks in mri images. *IEEE Transactions on Medical Imaging 35*, 5 (May 2016), 1240–1251.

[90] PHILLIPS, P. J., MOON, H., RIZVI, S. A., AND RAUSS, P. J. The feret evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, 10 (Oct. 2000), 1090–1104.

[91] RAI, A., CHUDASAMA, V., UPLA, K., RAJA, K., RAMACHANDRA, R., AND BUSCH, C. Comsupresnet: A compact super-resolution network for low-resolution face images. In *2020 8th International Workshop on Biometrics and Forensics (IWBF)* (2020), pp. 1–6.

[92] RAJAWAT, A., PANDEY, M. K., AND RAJPUT, S. S. Low resolution face recognition techniques: A survey. In *2017 3rd International Conference on Computational Intelligence Communication Technology (CICT)* (Feb 2017), pp. 1–4.

[93] RANJAN, R., CASTILLO, C. D., AND CHELLAPPA, R. L2-constrained softmax loss for discriminative face verification. *CoRR abs/1703.09507* (2017).

[94] RASTEGARI, M., ORDONEZ, V., REDMON, J., AND FARHADI, A. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Computer Vision – ECCV 2016* (Cham, 2016), B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Springer International Publishing, pp. 525–542.

[95] REYES, M., AND OLIVA, L. . Seguridad con videovigilancia. *Ciencia UNAM* (Oct 2018).

[96] ROBERTSON, D. J., NOYES, E., DOWSETT, A. J., JENKINS, R., AND BURTON, A. M. Face recognition by metropolitan police super-recognisers. *PLOS ONE 11*, 2 (02 2016), 1–8.

[97] RUDER, S. An overview of gradient descent optimization algorithms, 2016.

[98] SABOUR, S., FROSST, N., AND HINTON, G. E. Dynamic routing between capsules. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (USA, 2017), NIPS'17, Curran Associates Inc., pp. 3859–3869.

[99] SANDLER, M., HOWARD, A., ZHU, M., ZHMOGINOV, A., AND CHEN, L. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (June 2018), pp. 4510–4520.

[100] SAPKOTA, A., AND BOULT, T. E. Large scale unconstrained open set face database. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)* (Sep. 2013), pp. 1–8.

[101] SCHROFF, F., KALENICHENKO, D., AND PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. In *CVPR* (2015), IEEE Computer Society, pp. 815–823.

[102] SHAN, S., ZHANG, W., SU, Y., CHEN, X., AND GAO, W. Ensemble of piecewise fda based on spatial histograms of local (gabor) binary patterns for face recognition. In *18th International Conference on Pattern Recognition (ICPR'06)* (2006), vol. 3.

[103] SHEN, Z., LAI, W., XU, T., KAUTZ, J., AND YANG, M. Deep semantic face deblurring. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (June 2018), pp. 8260–8269.

[104] SHI, J., AND QI, C. From local geometry to global structure: Learning latent subspace for low-resolution face image recognition. *IEEE Signal Processing Letters 22*, 5 (May 2015), 554–558.

[105] SIENA, S., BODDETI, V. N., AND VIJAYA KUMAR, B. V. K. Coupled marginal fisher analysis for low-resolution face recognition. In *Computer Vision – ECCV 2012. Workshops and Demonstrations* (Berlin, Heidelberg, 2012), A. Fusiello, V. Murino, and R. Cucchiara, Eds., Springer Berlin Heidelberg, pp. 240–249.

[106] SIMONYAN, K., AND ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition, 2014.

[107] SINGH, M., NAGPAL, S., SINGH, R., AND VATSA, M. Dual directed capsule network for very low resolution image recognition. In *The IEEE International Conference on Computer Vision (ICCV)* (October 2019).

[108] SUN, Y., WANG, X., AND TANG, X. Deep learning face representation from predicting 10,000 classes. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 1891–1898.

[109] SVITOV, D., AND ALYAMKIN, S. Margindistillation: distillation for margin-based softmax, 2020.

[110] SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V., AND RABINOVICH, A. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 1–9.

[111] TAIGMAN, Y., YANG, M., RANZATO, M., AND WOLF, L. Deepface: Closing the gap to human-level performance in face verification. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2014).

[112] TAN, M., AND LE, Q. Mixconv: Mixed depthwise convolutional kernels. In *Proceedings of the British Machine Vision Conference (BMVC)* (September 2019), K. Sidorov and Y. Hicks, Eds., BMVA Press, pp. 116.1–116.13.

[113] TAN, M., AND LE, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[114] TIAN, D., AND TAO, D. Coupled learning for facial deblur. *IEEE Transactions on Image Processing 25*, 2 (Feb 2016), 961–972.

[115] TROTTIER, L., GIGUERE, P., AND CHAIB-DRAA, B. Parametric exponential linear unit for deep convolutional neural networks. *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)* (Dec 2017).

[116] TURK, M., AND PENTLAND, A. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition* (1991), pp. 586–587.

[117] UIBOUPIN, T., RASTI, P., ANBARJAFARI, G., AND DEMIREL, H. Facial image super resolution using sparse representation for improving face recognition in surveillance monitoring. In *2016 24th Signal Processing and Communication Application Conference (SIU)* (2016), pp. 437–440.

[118] VINAY, A., SHEKHAR, V., MURTHY, K. B., AND NATARAJAN, S. Face recognition using gabor wavelet features with pca and kpca - a comparative study. *Procedia Computer Science 57* (2015), 650–659. 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015).

[119] VIOLA, P., AND JONES, M. 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* (Dec 2001), vol. 1, pp. I–I.

[120] WANG, D., OTTO, C., AND JAIN, A. K. Face search at scale: 80 million gallery, 2015.

[121] WANG, F., CHENG, J., LIU, W., AND LIU, H. Additive margin softmax for face verification. *IEEE Signal Processing Letters 25*, 7 (2018), 926–930.

[122] WANG, H., WANG, Y., ZHOU, Z., JI, X., GONG, D., ZHOU, J., LI, Z., AND LIU, W. Cosface: Large margin cosine loss for deep face recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Jun 2018).

[123] WANG, M., AND DENG, W. Deep face recognition: A survey. *Neurocomputing 429* (2021), 215–244.

[124] WANG, Z., CHANG, S., YANG, Y., LIU, D., AND HUANG, T. S. Studying very low resolution recognition using deep networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 4792–4800.

[125] WANG, Z., LIU, D., YANG, J., HAN, W., AND HUANG, T. Deep networks for image super-resolution with sparse prior. In *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 370–378.

[126] WANG, Z., YANG, W., AND BEN, X. Low-resolution degradation face recognition over long distance based on cca. *Neural Computing and Applications 26* (02 2015).

[127] WEBB, A. R. Multidimensional scaling by iterative majorization using radial basis functions. *Pattern Recognition 28*, 5 (1995), 753 – 759.

[128] WEN, Y., ZHANG, K., LI, Z., AND QIAO, Y. A discriminative feature learning approach for deep face recognition. In *Computer Vision – ECCV 2016* (Cham, 2016), B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Springer International Publishing, pp. 499–515.

[129] WOLF, L., HASSNER, T., AND MAOZ, I. Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011* (June 2011), pp. 529–534.

[130] WU, X., HE, R., SUN, Z., AND TAN, T. A light cnn for deep face representation with noisy labels, 2015.

[131] XIN, J., WANG, N., JIANG, X., LI, J., HUANG, H., AND GAO, X. Binarized neural network for single image super resolution. In *Computer Vision – ECCV 2020* (Cham, 2020), A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Springer International Publishing, pp. 91–107.

[132] YAN, M., ZHAO, M., XU, Z., ZHANG, Q., WANG, G., AND SU, Z. Vargfacenet: An efficient variable group convolutional neural network for lightweight face recognition. In *The IEEE International Conference on Computer Vision (ICCV) Workshops* (Oct 2019).

[133] YANG, F., YANG, W., GAO, R., AND LIAO, Q. Discriminative Multidimensional Scaling for Low-Resolution Face Recognition. *IEEE Signal Processing Letters 25*, 3 (mar 2018), 388–392.

[134] YANG, H., FRITZSCHE, M., BARTZ, C., AND MEINEL, C. Bmxnet: An open-source binary neural network implementation based on mxnet. In *Proceedings of the 25th ACM International Conference on Multimedia* (New York, NY, USA, 2017), MM '17, ACM, pp. 1209–1212.

[135] YANG, J., WRIGHT, J., HUANG, T. S., AND MA, Y. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing 19*, 11 (2010), 2861–2873.

[136] YI, D., LEI, Z., LIAO, S., AND LI, S. Z. Learning face representation from scratch. *ArXiv abs/1411.7923* (2014).

[137] YIN, X., TAI, Y., HUANG, Y., AND LIU, X. Fan: Feature adaptation network for surveillance face recognition and normalization. In *Computer Vision – ACCV 2020* (Cham, 2021), H. Ishikawa, C.-L. Liu, T. Pajdla, and J. Shi, Eds., Springer International Publishing, pp. 301–319.

[138] YUAN, C., AND AGAIAN, S. S. A comprehensive review of binary neural network, 2021.

[139] ZEYDE, R., ELAD, M., AND PROTTER, M. On single image scale-up using sparse-representations. In *Curves and Surfaces* (Berlin, Heidelberg, 2012), J.-D. Boissonnat, P. Chenin, A. Cohen, C. Gout, T. Lyche, M.-L. Mazure, and L. Schumaker, Eds., Springer Berlin Heidelberg, pp. 711–730.

[140] ZHA, J., AND CHAO, H. Tcn: Transferable coupled network for cross-resolution face recognition*. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), pp. 3302–3306.

[141] ZHANG, J., GUO, Z., LI, X., AND CHEN, Y. Large Margin Coupled Mapping for Low Resolution Face Recognition. In *PRICAI 2016: Trends in Artificial Intelligence* (Berlin, Heidelberg, 2016), C. Zhang, H. W. Guesgen, and W.-K. Yeap, Eds., vol. 3157 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 661–672.

[142] ZHANG, K., GU, S., TIMOFTE, R., HUI, Z., WANG, X., GAO, X., XIONG, D., LIU, S., GANG, R., NAN, N., LI, C., ZOU, X., KANG, N., WANG, Z., XU, H., WANG, C., LI, Z., WANG, L., SHI, J., SUN, W., LANG, Z., NIE, J., WEI, W., ZHANG, L., NIU, Y., ZHUO, P., KONG, X., SUN, L., AND WANG, W. Aim 2019 challenge on constrained super-resolution: Methods and results. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (2019), pp. 3565–3574.

[143] ZHANG, K., ZHANG, Z., CHENG, C.-W., HSU, W. H., QIAO, Y., LIU, W., AND ZHANG, T. Super-identity convolutional neural network for face hallucination. *Lecture Notes in Computer Science* (2018), 196–211.

[144] ZHANG, K., ZHANG, Z., LI, Z., AND QIAO, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters 23*, 10 (Oct 2016), 1499–1503.

[145] ZHANG, P., BEN, X., JIANG, W., YAN, R., AND ZHANG, Y. Coupled marginal discriminant mappings for low-resolution face recognition. *Optik 126*, 23 (2015), 4352 – 4357.

[146] ZHANG, Q., LI, J., YAO, M., SONG, L., ZHOU, H., LI, Z., MENG, W., ZHANG, X., AND WANG, G. Vargnet: Variable group convolutional neural network for efficient embedded computing. *ArXiv abs/1907.05653* (2019).

[147] ZHANG, X., ZHOU, X., LIN, M., AND SUN, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Jun 2018).

[148] ZHOU, A., YAO, A., GUO, Y., XU, L., AND YURONG, C. Incremental network quantization: Towards lossless cnns with low-precision weights. In *International Conference on Learning Representations,ICLR2017* (2017).

[149] ZHOU, S., WU, Y., NI, Z., ZHOU, X., WEN, H., AND ZOU, Y. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients, 2018.

[150] ZHU, S., LIU, S., LOY, C. C., AND TANG, X. Deep cascaded bi-network for face hallucination. In *Computer Vision – ECCV 2016* (Cham, 2016), B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Springer International Publishing, pp. 614–630.

[151] ZHUANG, B., SHEN, C., TAN, M., LIU, L., AND REID, I. Structured binary neural networks for accurate image classification and semantic segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 413–422.

Curriculum Vitae

M.Sc. Luis Santiago Luévano García was born in Mexico City, Mexico. He finished his B.Sc. in Computer Science and Technology at Tecnológico de Monterrey in 2015. After a year and a half of industry experience at Dell, Inc., in 2016 Luis Santiago pursued an M.Sc. Computer Science degree at the Stevens Institute of Technology in New Jersey, USA, graduating in 2018, where he focused his studies on artificial intelligence and computer vision. In 2019 Luis Santiago enrolled in the Ph.D. Computer Science program at his *alma mater*, Tecnológico de Monterrey, as part of the Intelligent Systems Research Group at Tecnologico de Monterrey, focused on Very Low resolution Face Recognition research as his doctoral thesis topic, striving to close the gap for real-time applications. In 2022, he undertook a research stay at the Biometrics Security and Privacy research group at the Idiap Research Institute in Switzerland, working on expanding state-of-the-art Very Low Resolution Face Recognition methods with permissive hardware resources.

This document was typed in using LaTeX 2_ε[a] by Luis Santiago Luévano García.

[a] The style file `phdThesisFormat.sty` used to set up this thesis was prepared by the Center of Intelligent Systems of the Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Estado de México