Low-Power Face Recognition using Joint Optical and Electronic Deep Neural Network

Xuening Dong

Department of Electrical & Computer Engineering

McGill University, Montreal

April 2024

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of

Master of Science - Electrical Engineering

Abstract

Face detection and recognition are key technologies for human-computer interaction, surveillance, and biometric authentication. However, the trade-off between power consumption and accuracy is crucial for such algorithms running locally on edge devices. In addition, inference latency is an important constraint for real-time systems. In this work, we propose an electro-optic hybrid multi-stage machine learning system for decoupled detection and recognition of faces. The system uses an always-on Mach-Zehnder Interferometer (MZI)-based optical neural network (ONN) to monitor the presence of faces in the environment. The computationally intensive digital deep neural network (DNN) for facial recognition is only turned on once a face is detected. This system enables lower power consumption and faster operation compared to a traditional, pure-electronic, always-on system in two ways. First, the always-on face detector takes advantage of the inherent parallelism of ONN to perform vector-matrix multiplications in linear time complexity with less power consumed. Second, the normally powered-down DNN for facial recognition significantly lowers energy consumption.

To verify the correctness and efficiency of the proposed design, we applied it to two different use cases - one with centered face alignment and the other without. We train and test the system using Neuroptica and PyTorch platforms with both the WIDER Face and Labeled Face in the Wild (LFW) datasets. Under the assumption of 1% face appearance probability, the most accurate model arises from the unaligned face scenario with a subsampling process for finding faces. It achieves 97.2% accuracy on the LFW dataset, coupled with a 10.9% reduction in power consumption compared to the same neural net-

work on digital processors. In the aligned face scenario, the best accuracy drops to 95.8% but is accompanied by a remarkable twofold reduction in power and energy usage.

We further verify the performance of the proposed system with non-ideal operating conditions by taking into account the phase drifts during the programming of phase shifters and the propagation loss of light. In general, the face recognition accuracy degrades as the non-ideal conditions deteriorate. The ONNs are comparatively more susceptible to phase drifts than propagation loss. By assuming 0.6 dB/MZI propagation loss and varying the magnitude of phase drifts, the most accurate model from the perfect operating condition experienced a maximum of an absolute 6.5% drop in system accuracy. The overall worst model can only achieve an accuracy of 70%. Finally, the power and energy consumption of the majority of the selected best-performing models drops as the non-idealities induce more false negative cases and wake up the power-demanding DNNs less often.

Abrégé

La détection et la reconnaissance des visages sont des technologies clés pour l'interaction homme-machine, la surveillance et l'authentification biométrique. Cependant, le compromis entre la consommation d'énergie et la précision est crucial pour de tels algorithmes fonctionnant localement sur des appareils périphériques. En outre, la latence de l'inférence est une contrainte importante pour les systèmes en temps réel. Dans ce travail, nous proposons un système hybride électro-optique d'apprentissage automatique en plusieurs étapes pour la détection et la reconnaissance découplées des visages. Le système utilise un réseau neuronal optique (ONN en anglais) basé sur un interféromètre Mach-Zehnder toujours actif pour surveiller la présence de visages dans l'environnement. Le réseau neuronal numérique profond (DNN en anglais) à forte intensité de calcul pour la reconnaissance faciale n'est activé que lorsqu'un visage est détecté. Ce système permet de réduire la consommation d'énergie et d'accélérer le fonctionnement par rapport à un système traditionnel, purement électronique et toujours actif, et ce de deux manières. Tout d'abord, le détecteur de visages toujours actif tire parti du parallélisme inhérent à l'ONN pour effectuer des multiplications vectorielles et matricielles en temps linéaire, tout en consommant moins d'énergie. Deuxièmement, le DNN normalement éteint pour la reconnaissance faciale réduit considérablement la consommation d'énergie.

Pour vérifier l'exactitude et l'efficacité de la conception proposée, nous l'avons appliquée à deux cas d'utilisation différents - l'un avec alignement centré des visages et l'autre sans. Nous avons formé et testé le système en utilisant les plateformes Neuroptica et PyTorch avec les ensembles de données WIDER FACE et Labeled Face in the Wild

(LFW). Dans l'hypothèse d'une probabilité d'apparition des visages de 1%, le modèle le plus précis provient du scénario des visages non alignés avec un processus de sous-échantillonnage pour trouver les visages. Il atteint une précision de 97.2% sur l'ensemble de données LFW, associée à une réduction de 10.9% de la consommation d'énergie par rapport au même réseau neuronal sur des processeurs numériques. Dans le scénario des visages alignés, la meilleure précision tombe à 95.8%, mais s'accompagne d'une remarquable réduction par deux de la consommation d'énergie.

Nous vérifions en outre les performances du système proposé dans des conditions de fonctionnement non idéales en tenant compte des dérives de phase pendant la programmation des déphaseurs et de la perte de propagation de la lumière. En général, la précision de la reconnaissance des visages se dégrade au fur et à mesure que les conditions non idéales se détériorent. Les ONN sont comparativement plus sensibles aux dérives de phase qu'aux pertes de propagation. En supposant une perte de propagation de 0.6 dB/MZI et en variant l'ampleur des dérives de phase, le modèle le plus précis dans des conditions de fonctionnement parfaites a connu une baisse maximale de 6.5% en valeur absolue de la précision du système. Le plus mauvais modèle ne peut atteindre qu'une précision de 70%. Enfin, la consommation d'énergie de la majorité des modèles les plus performants sélectionnés diminue à mesure que les non-idéalités induisent davantage de cas de faux négatifs et réveillent moins souvent les DNN gourmands en énergie.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Brett Meyer. Our meetings and conversations were vital in inspiring me to configure and refine the ideas of this thesis. Thank you for your invaluable guidance and support. Additionally, I would like to thank Professor Odile Liboiron-Ladouceur who introduced me to the world of photonics and helped me understand the underlying principles of optical processors.

I am also grateful to my colleagues at McGill, especially Bokun Zhao, Leonid Pascar, and Dr. Kaveh Rahbardar Mojaver, for their support and thoughtful feedback on my work.

Last but most importantly, I would be remiss in not mentioning my family and friends who have supported me in getting through the hard times in my study and research. Thank you for your love and encouragement.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ASIC Application-Specific Integrated Circuit.

AUROC Area Under the Receiver Operating Characteristics curve.

CNN Convolutional Neural Network.

DAE Deep Autoencoder.

DNN Deep Neural Network.

FFT Fast Fourier Transform.

FN False Negative.

FoM figure of merit.

FP False Positive.

FPR False Positive Rate.

GPU Graphics Processing Unit.

ID In-distribution.

IoU Intersection over Union.

kNN k-Nearest Neighbors.

LFW Labeled Faces in the Wild.

MAC multiply-and-accumulation.

MLP Multilayer Perceptron.

MTCNN Multi-Task Cascaded Convolutional Neural Networks.

MZI Mach-Zehnder Interferometer.

NAS Neural Architecture Search.

NN Neural Network.

ONN Optical Neural Network.

OOD Out-of-distribution.

PCA principle components analysis.

ROC Receiver Operating Characteristics curve.

SOTA state-of-the-art.

SVD singular value decomposition.

TN True Negative.

TP True Positive.

TPR True Positive Rate.

VMM Vector-Matrix Multiplication.

Chapter 1

Introduction

Face detection and recognition have been trending topics in the realm of computer vision. State-of-the-art face detection and recognition algorithms deploy Deep Neural Networks (DNNs) with stacked convolutional and fully connected layers to locate and verify facial biometrics [2, 3]. While these algorithms can be implemented for real-time execution on resource-constrained edge devices, this comes at high computational cost and energy consumption. Moreover, for time-sensitive applications where inference latency is crucial, such as authentication [4] and security [5], responsiveness is key.

Given the "always-on" nature of embedded face detection and recognition, complex models run continuously, resulting in high inference energy and power. Recent research efforts balancing face recognition accuracy and computational complexity can be divided into two approaches: devising resource-efficient algorithms for edge devices and applying emerging technology for more resource-efficient execution of existing algorithms. The former either optimizes an existing architecture [6] or proposes novel architectures [7] to reduce the number of DNN parameters. However, they still assume the highly accurate DNNs are always on, resulting in substantial power and energy consumption; in addition, the algorithms rely on Graphics Processing Units (GPUs) to minimize latency. Some more recent approaches develop the idea of multi-stage execution of face detection and recognition [8,9]. The face recognition pipeline is decomposed into three stages: detecting

the presence of a face, locating the position of the face, and facial feature extraction. Only the first stage always runs, while the rest of the system is woken up only when a face is present. This face detection stage, running a low-complexity model to save resources, is expected to have a higher error rate, but ideally few false negatives.

Researchers have also attempted to leverage emerging technology for energy-efficient neural network acceleration. Analog computing with neuromorphic electronics can reduce energy and latency overhead caused by data sampling and digitalization [10]. However, electronic devices still suffer from limited bandwidth in metal wires and slower performance improvement as Moore's Law slows [10]. Therefore, neuromorphic photonics serves as a promising alternative method. Data propagates at the speed of light in the waveguides, and Multiply-and-accumulation (MAC) operations can be performed in devices such as Mach-Zehnder Interferometers (MZIs) [11]. The concept of ONNs has been studied previously in [1], and under perfect operating conditions, the optical processor can achieve a maximum speed of 100 GHz, with N mW of power spent on executing an N-dimensional vector-matrix multiplication. Despite the compelling speed and energy savings, ONNs suffer from scalability limitations and precision loss due to imperfect fabrication and noise. These factors restrict the size of ONN to around 10×10 neurons [12] and reduce overall accuracy.

In this work, we combine the idea of a multi-stage wake-up algorithm and the use of an optical processor for face detection, benefiting from the energy-efficient ONN and mitigating its low accuracy with event-driven digital DNNs. The face detection stage is executed on the optical processor while the rest of the system remains digital. Although 2D-pixel array input to the optical processor has recently been proven to be feasible [13], we focus on a more conventional way of using the optical processor, which involves lasers for light injection and an additional stage of dimensionality reduction. We consider the input/output size, topologies, and number of ONN layers as the hyperparameters of the ONN design. Moreover, we added false negative reduction methods to the training of the face detection model to overcome the challenge of undetected valid faces. We perform a

grid search over all possible designs and select the best models from the Pareto optimal solution set.

Under perfect operating conditions, the most accurate model achieves 97.2% accuracy on the LFW [14] dataset with 16 features extracted for an ONN in Clements [15] topology with a subsampling process for finding faces. This model achieves 11% savings in power and 57.5% shorter latency compared to always-on digital processors implementing the same neural networks. On the other hand, the most efficient model allows a $1.83\times$ reduction in power, latency, and energy, but with an absolute sacrifice of 7% in accuracy.

Device imperfection will alter the accuracy of face recognition. In this work, we mainly consider the phase shift drifts in the programming process of phase shifters and the propagation loss of light through waveguides. Comparatively, the phase shifter programming deviations have a more profound impact than the attenuation of light in waveguides on the ONN and the overall system, as seen by a greater magnitude of accuracy drop as deviation increases. We subject the optimal face recognition models selected with perfect operating assumptions to imperfect conditions, by fixing the propagation loss to 0.6 dB per MZI and injecting random Gaussian noise to the phase shifts. The most accurate model under perfect operating conditions experienced a worst-case 6.5% drop in end-to-end face recognition accuracy. Meanwhile, the least imperfection-resilient model has its accuracy dropped below 70%. Finally, the majority of the selected models exhibit a decline in power and energy consumption as the phase drift magnitude increases. This is attributed to the growing number of false negatives made by ONN, leading to less frequent activation of the power-demanding digital DNN.

The rest of this thesis is organized as follows: in Chapter 2, we review the concepts and algorithms of face detection and recognition, especially their implementation on edge devices. We also introduce the basic concepts of optical neural networks, including the underlying mathematics, the imperfect operation of the hardware, and the history of its deployment in deep learning. Next, in Chapter 3, we present an overview of the proposed electro-optical hybrid system, including its algorithmic components and the evaluations

applied. In Chapter 4, we describe the experiments to be conducted for the verification and validation of the proposed system. In Chapters 5 and 6, we show our results of the experiments under the lossless and lossy operating conditions of ONN. Finally, in Chapter 7, we conclude our observations and explore potential avenues for improvements of the current design.

Statement of Contribution

All contents in this thesis are my original work, including texts, tables, and figures in Chapters 1 to 7, and data and plots in Chapters 5 and 6.

Chapter 2

Background and Related Work

As stated in Chapter 1, current researchers tackle the challenge of balancing face recognition precision and its computational complexity from both software and hardware perspectives. In software, researchers devise efficient algorithms to perform face recognition at a high accuracy but with fewer arithmetic operations. In hardware, they take note of emerging technology for more power- and energy-efficient execution of arithmetic operations. This chapter provides an introduction to the current research landscape of face detection and recognition algorithms and discusses some State-of-the-art (SOTA) solutions to the problem of accuracy-efficiency tradeoffs in terms of the software algorithm and hardware accelerator design.

2.1 Face Detection and Recognition

Face detection and recognition algorithms have been widely applied to daily applications such as human-computer interaction (HCIs), surveillance, and biometric authentication. In this section, we review the key advancements in these algorithms and bring in challenges faced by today's edge systems deploying such algorithms.

2.1.1 Face Detection

The goal of face detection is to determine the presence of faces in an image and locate their positions using bounding boxes if present [16]. Such face detection algorithms can be applied to real-life use cases ranging from standalone tasks such as digital camera auto-focus [17] and organizing electronic photo albums [18] to assisting other systems such as face verification [19], age and gender estimation [20].

State-of-the-art face detection algorithms can be divided into four categories [16]: cascade-based methods, part-based methods, channel-feature-based methods, and Neural Network (NN)-based methods. Cascade-based methods contain multiple stages for extracting and learning engineered facial features. For instance, Zhang *et al.* [21] introduces a cascaded system feeding Haar-like facial features extracted by Viola-Jones algorithm [22] to AdaBoost classifiers for detection.

Part-based methods consider human faces as an aggregation of local facial features, such as hair, eyes, mouth, and nose. Subsequently, attribute-aware models are designed to identify and locate each of the local features and their responses are combined to form the candidate window of faces [23]. The candidate window is then sent to a classifier model to determine its "faceness".

Channel-feature-based methods focus on the colour channels present in the greyscale or RGB images. Selected channels are subsampled by a preset factor and aggregated into a pixel lookup table [24]. A weak classifier such as a decision tree or Adaboost classifier subsequently discerns intra- and inter-channel correlations to facilitate classification.

Finally, the NN-based methods use convolutional layers to capture spatial patterns in images and generalize common features across different faces. Powerful DNNs such as MTCNN [19], RetinaFace [25] use convolutional layers for feature extraction, and fully-connected layers for bounding box and face presence probability calculation.

Accompanying the rise in the number of NN-based face detectors, large-scale face detection benchmark datasets such as AFLW [26], PASCAL FACES [27] and WIDER FACE [16] are created for training and testing of models. These datasets contain hundreds of

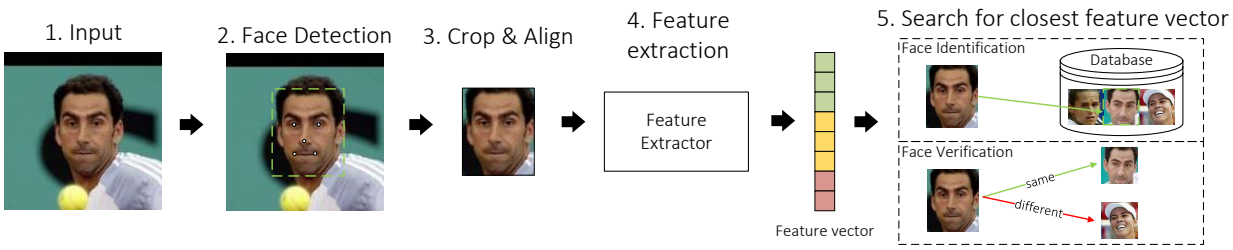


Figure 2.1: Face recognition pipeline.

thousands of data instances retrieved from search engines [16], or augmented from existing datasets [26]. The bounding boxes of the faces are then annotated by crowdsourcing. To enhance the generalization of models trained on them, these datasets take into account unconstrained conditions where images are captured with various poses, scales, face expressions, and occlusions.

2.1.2 Face Recognition

The face recognition pipeline consists of the processes of face detection, alignment, representation learning (feature extraction), and classification as shown in Fig. 2.1 [28]. Its application can be further divided into face verification, where the given face is compared with other faces in an existing face database (*"Is this the same person?"*), and face identification, where the given face is identified within a list of names (*"Who is this person?"*). These systems can be applied to real-life scenarios such as attendance monitoring [29], automated border control [30], and video surveillance [31].

Traditional face recognition systems utilize deterministic algorithms such as PCA and Linear Discriminant Analysis (LDA) to derive personal face features and decide the identity of the person. A famous example of the application of these systems is the Eigenface [32]. The essence of "eigenface" is to project the facial images onto a feature space, or the "face space", to encode the variation among different faces. A face detector determines the "faceness" of a figure by calculating the proximity between the image projec-

tion and the face space, and a face recognition model computes the similarity between the given face and the database to identify the person.

Due to the recent rapid development of deep networks, face recognition models now rely on deep Convolutional Neural Network (CNN) to capture the features of the human face depicted in the image. The most common approach is to directly look at the faces as aggregated pixels (similar to the NN-based face detector) instead of looking at engineered features (similar to the part-based face detector). The face pixels are transformed into lower-dimensional face embeddings and the distance or similarity score between the two embeddings is calculated. Researchers deploy state-of-the-art DNNs such as ResNet, and Inception Network as backbones of face recognition models and design loss functions to assist the models in capturing the facial features more efficiently. In face verification, the ArcFace model [33] proposed an Additive Angular Margin Loss to capture the geometric features of the face by viewing it as a hypersphere. Similarly, in face identification, the AdaFace model [34] combined the margin loss with image quality factors to enhance the model performance when encountering low-quality images. After the face embeddings are obtained, in face verification, a threshold is used for determining whether the two faces belong to the same person [2]. Meanwhile, in face identification, an additional fully connected layer is attached to perform multi-class classification of the face.

To ensure that models can achieve similar invariance to different face and surrounding conditions, the DNN models are trained on large datasets that contain instances with variations in pose, illumination, and other factors [2]. A majority of the face recognition datasets, such as Labeled Face in the Wild (LFW) [14], VGG-Face [3] are created from static personal photos of celebrities queried from search engines. Some more recent datasets [35] consider video pairs over images for better approximation of real-life tasks.

2.1.3 Deep Learning Based Face Recognition on Edge Devices

Currently, deep CNN has been deployed to perform face detection and recognition tasks with extremely low error rates. However, when implemented on edge devices, these

DNNs struggle with maintaining a balance between accuracy and computing resource (power, latency, and energy) consumption [36]. The always-on device consumes a massive amount of power and energy as the inference repeats over time. As a result, researchers are looking for more resource-efficient algorithms that are capable of maintaining state-of-the-art face detection and recognition performance with fewer arithmetic operations during inference.

The first set of solutions considers using lightweight systems as the backbone NN architecture for existing face recognition framework to reduce the computations performed during a single inference [37]. The MobileNets [38] are a set of efficient CNNs designed specifically for mobile vision applications that are resource-constrained. They employ depthwise separable convolutions to break down the traditional convolutional layers into pointwise and depthwise convolutions. This step drastically reduces the number of parameters in the layers and therefore results in less floating point operations consumed. Serving as a backbone, a later version of MobileNet, MobileNetV2 [36] is combined with the FaceNet framework for face recognition tasks. the combined structure achieves the same level of face recognition accuracy as FaceNet with a $7.5\times$ smaller parameter size.

As designing neural network architectures from scratch can be a demanding job for human experts, recent studies also focus on developing Neural Architecture Search (NAS) algorithms that are tailored to deep face recognition applications. NAS algorithms are based on reinforcement learning or evolutionary algorithms and they automatically search for better architecture designs by tuning the hyperparameters based on given metrics such as accuracy, latency, and power consumption. Zhu *et al.* [39] explores a design space with various magnitudes of connectivity between convolution layer blocks of a face recognition model using reinforcement learning. Their explored optimal design also achieved the same level of face recognition accuracy as FaceNet with $1.8\times$ fewer parameters.

As introduced in Section 2.1.2, the face recognition pipeline contains various stages. Researchers can then take advantage of this stage-wise operation of face recognition al-

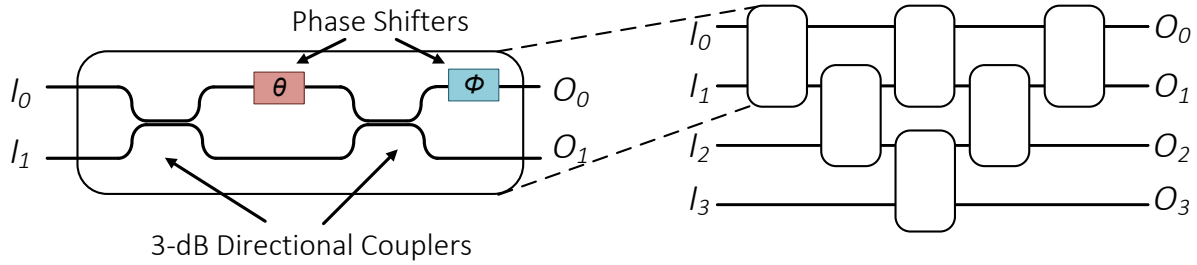


Figure 2.2: The configuration of a 2×2 optical processor as a part of a 4×4 ONN.

gorithms and consider shutting down part of the hardware when it is not in use. Bong *et al.* [8] proposed a system structure composed of an always-on face detection module and an event-driven CNN-based face recognition module. The Viola-Jones-based face detector constantly scans for a face in the environment and only triggers the face recognition module when it believes a valid face is present. The designed system fabricated on an Application-Specific Integrated Circuit (ASIC) chip consumes only 0.62 mW power to evaluate one face with a sacrifice of 2% drop in accuracy compared to state-of-the-art deep face recognition networks. Similarly, Synopsys designed a two-stage face recognition system with one "always-on" processor holding a low-resolution face detector and one event-driven processor for high-accuracy face detection and recognition [9]. They prove that the proposed stage-wise functionalities are implementable on existing commercial processors and can alleviate the low-power high-performance trade-off.

2.2 Optical Neural Networks

Silicon photonics is one of the emerging solutions for accelerating the Vector-Matrix Multiplication (VMM) operations, owing to its faster speed, lower energy consumption, and CMOS-compatible manufacturing capability [10, 40]. With information encoded in light waves, optical modulators such as MZIs, and Micro-Ring Resonators (MRRs) can be organized into linear optical processors and perform VMM by light interference. The output electromagnetic field intensity is expressed mathematically as the multiplication between

the input field intensity and the transfer matrix represented by the processor. These reconfigurable devices can map any arbitrary complex matrices to it by specifically designing the programming process. In this thesis, we focus on the use of MZIs as the main component of linear optical processors.

2.2.1 Transfer Matrix and Programming

The optical processors can be seen as the combination of multiple 2×2 reconfigurable units. Each 2×2 unit consists of two 3 dB directional couplers and two phase shifters, one on the internal arm (θ) and the other on the external arm (ϕ), as shown in Fig. 2.2 [11]. Theoretically, the 3 dB couplers equally split the input power to each output branch. To achieve an arbitrary split at the output ports of the unit, the internal phase shifter θ adjusts the phase difference between the two arms and the external phase shifter ϕ controls the phase difference between the two output ports (O_1, O_2). The transfer matrix of the 2×2 reconfigurable unit (D_{MZI}) is then formulated as a multiplication of the transformation matrices of each component and the final result is shown in Eqn (2.1).

$$D_{MZI} = je^{j(\frac{\theta}{2})} \begin{bmatrix} e^{j\phi}\sin(\frac{\theta}{2}) & e^{j\phi}\cos(\frac{\theta}{2}) \\ \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \end{bmatrix} \tag{2.1}$$

The designed phase shifts are programmed to phase shifters by applying a bias voltage to the optical waveguide. The phase shifters considered in this work take advantage of the thermo-optical effect. By applying the bias voltage to the phase shifters, thermal changes occur, and the refractive index of the material changes subsequently, altering the interference pattern.

2.2.2 Optical Processor Topology

Transformation matrices in larger sizes can be realized by organizing the 2×2 reconfigurable units into different topologies such that different $[D_{\text{MZI}}]$ are multiplied and con-

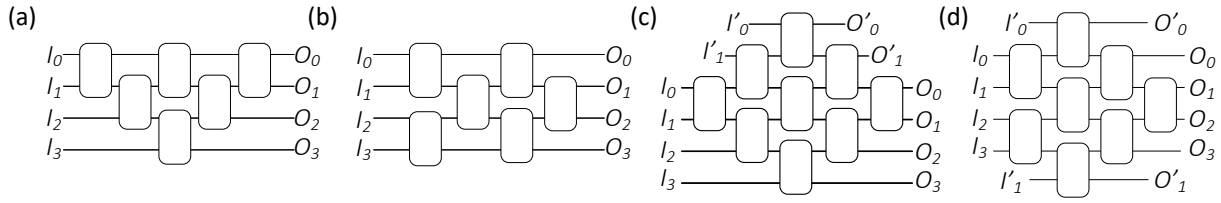


Figure 2.3: Topologies of optical processors: (a) Reck, (b) Clements, (c) Diamond, and (d) Bokun with 4 input-output ports.

catenated [11]. Commonly seen topologies are shown in Fig. 2.3, and their properties are listed in Table. 2.1. N denotes the number of input/output ports of the mesh. The processor depth is the maximum number of MZIs deployed on the longest input-output path.

The Reck [41] topology (Fig. 2.3(a)) consists of a triangular mesh of MZIs. The triangular shape allows for a more convenient sequential calibration of MZIs from the vertex to the rest of the mesh. The Clements [15] topology (Fig. 2.3(b)) deploys the same number of MZIs as Reck while organizing them into a rectangular shape. This change in shape shortens the processor depth and leads to lower loss experienced by the processor even when light propagates through the longest input-output path. The Diamond [42] topology (Fig. 2.3(c)), as stated in its name, has a symmetrical diamond shape and allows more consistent path losses between each input-output ports pair. The consistent path lengths ensure the optimal light intensity directed to the output ports and avoid the light intensity issues led by the tapered-out waveguides [42]. The topology also allows independent access to all MZIs as a result of the diagonal input-output paths and therefore, supports

Table 2.1: Summary of Optical Processor Topologies and Properties

Name	Number of MZIs	Processor Depth
Reck	$N(N-1)/2$	$2N-3$
Clements	$N(N-1)/2$	N
Diamond	$(N-1)^2$	$2N-3$
Bokun	$N(N+N/2-2)/2$	N

simpler and more accurate programming of phase angles to the phase shifters [43]. The Bokun [43] topology (Fig. 2.3(d)) further modifies the Diamond topology by keeping only the middle optical input-output ports for main optical paths and leaving the other ports for calibration purposes only. This modification further minimizes the processor depth and enhances the robustness of the model towards operation noises.

2.2.3 Non-ideal Operation of Optical Processor

Fabrication process variation is one of the major concerns on the accuracy of computations performed on optical processors. During fabrication, the waveguide dimensions are altered and the geometric shapes of optical components become different from the simulation. For the 3 dB couplers, their splitting ratios are no longer exact; for the phase shifters, the bias voltage for inducing the same phase shifts varies across devices and leads to imprecise mapping of phase angles to them. Moreover, thermal crosstalk during the programming of a phase shifter leads to unexpected heat-up of other waveguides, creating unintended phase changes to other phase shifters in the processor. The overall propagation loss or insertion loss of the waveguide also increases due to the roughness of sidewalls during fabrication, resulting in more attenuated signals.

All of the aforementioned fabrication imperfections and material restrictions can be summarized into two effects that directly lead to changes in the transfer matrix: the deviation in phase shifts (θ and ϕ) programmed to the phase shifters and the propagation loss defined at a per-MZI basis [42].

2.2.4 Silicon Photonics for Deep Learning

Over the past decade, there has been a growing amount of research work done on creating optical processors for accelerating the computation of neural networks. While the processors demonstrate their potential in increasing the inference speed with less power and better energy efficiency, the loss in computation accuracy compared to digital proces-

Table 2.2: Summary of Related Work Architectures and Performance

Paper	Training	Task / Dataset	Architecture		Accuracy [%]	
			Layers	Neurons	Trained	Actual
Shen *et al.* 2017 [1]	Ex-situ	Vowel Sound Recognition	2	4x4	91.8	76.7
Hughes *et al.* 2018 [44]	In-situ	Ring Separation	6	4x4	91.0	N/A
Shokraneh *et al.* 2019 [45]	Ex-situ	Linear Separation	1	4x4	98.9	72.0
Williamson *et al.* 2020 [46]	In-situ	MNIST	3	16x16	93.9	N/A
Zhang *et al.* 2021 [47]	Ex-situ	Iris	1	4x4	99.3	97.4
		MNIST	1*	4x4	93.5	N/A
Mojaver *et al.* 2023 [43]	In-situ	MNIST	1	10x10	70.0	N/A
			2		83.5	

*Contains two additional digital layers with size 784x4 and 4x10 before and after ONN

sors, and the scalability issues in implementing deep neural networks on the devices are yet concerns to be addressed. Table. 2.2 summarizes the accuracy of ONNs reported by recent work. The ex-situ training assumes the neural networks are trained with weights in the digital format as tunable parameters and the in-situ training assumes the ONNs are trained with phase angles programmed to the phase shifters. The trained accuracy is obtained from computer simulations while the actual accuracy is the inference results obtained on fabricated chips.

Shen *et al.* [1] demonstrated the practicality of implementing a digitally trained neural network on optical processors. They proposed a two-layer fully connected neural network architecture for vowel speech recognition tasks and mapped the ex-situ trained weights to the optical processors. The weight matrices are decomposed into two unitary matrices and one diagonal matrix using the Singular value decomposition (SVD). The unitary matrices are directly mapped to the optical processors and the diagonal matrix is implemented on a special diagonal matrix layer as shown in Fig. 2.4. Results have shown that the power consumption of the proposed optical processor is proportional to the number of neurons implemented and the inference speed is at least two orders of magnitude faster than state-of-the-art electronic systems. However, the system suffered severely from the degradation of classification accuracy as a result of the imperfect operation of devices and the finite precision that a phase can be set.

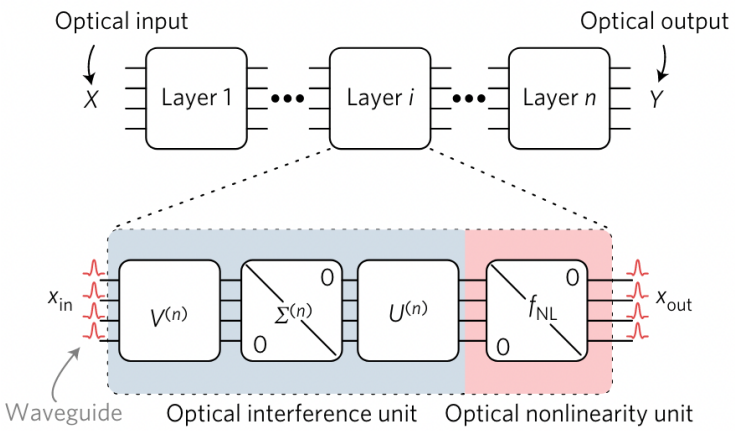


Figure 2.4: The architecture of ONN implemented in [1], with a general neural network layer decomposed into an optical interference unit and an optical non-linearity unit.

To take care of the finite precision of phase angles in the phase shifters, the concept of in-situ training is introduced such that the ONNs can be trained entirely on phase angles while keeping the directional couplers at 3 dB. Hughes *et al.* [44] proposed an in-situ backpropagation scheme based on Maxwell's equations. The gradient of loss functions is numerically calculated in terms of the phase shifts for weight updates. The proposed system is tested by training a single layer 3×3 fully connected neural network to implement the XOR gate. The network reached 100% accuracy after training for 400 iterations.

Activation functions play a critical role in neural networks for introducing non-linearity to the functions. Most of the current hardware-based activations are implemented electronically, challenging their incorporation within optical processors. Williamson *et al.* [46] introduced a reprogrammable electro-optic activation function that can be implemented within the optical system and achieved a ReLU-like response by tuning the parameters. Fully connected neural networks trained on such activation function can reach a maximum of 94% classification accuracy on the MNIST dataset.

To further enhance the robustness of ONN towards fabrication imperfections, Mourgias-Alexandris *et al.* [48] proposed a novel coherent neuromorphic photonic computing device [49] which contains a phase shifter followed by an amplitude modulating element per neuron and equipped it with noise-aware training models. They showed that by in-

jecting the experimentally obtained noise characteristics of the silicon photonics circuitry to the neural network during training, they were able to achieve an on-chip classification accuracy of $> 99\%$ on the MNIST dataset and 6 orders of magnitude faster speed than the cascaded MZI design.

More recent studies have suggested the possibility of implementing CNNs on optical processors with a similar operation to the GPUs [40]. However, the majority of the current investigations are still limited to fully connected networks and generated tasks such as simulating XOR gates, separating data points based on mathematical expressions, and handwritten digit recognition. In this work, we focus on a more practical scenario of a face detection system.

Chapter 3

Methodology

3.1 Face Recognition System Overview

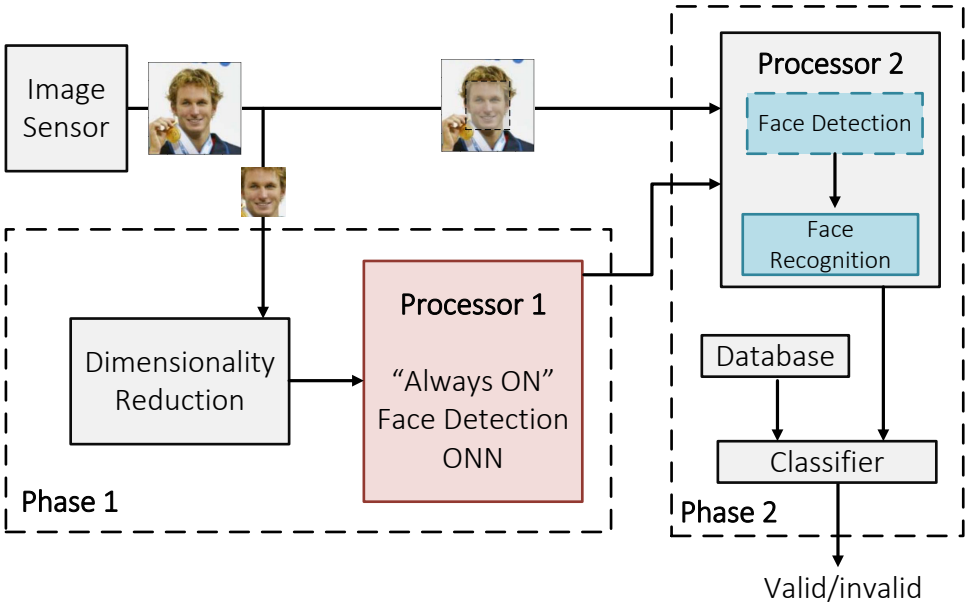


Figure 3.1: Flowchart for the two-phase electro-optic hybrid face recognition system. Processor 1 is the optical processor for accelerating face detection and processor 2 is the electronic processor for high-accuracy face recognition.

As shown in Fig. 3.1, the face detection and recognition system consists of two phases, with an always-on, low-power optical processor that performs detection, and an event-

driven high-performance processor that performs recognition. The system's goal is to achieve high end-to-end face recognition accuracy while consuming less power and energy compared with the same algorithm implemented on purely digital devices. To achieve high overall accuracy, apart from using a highly accurate face recognition model, the face detection model should minimize the number of its FN predictions, which directly fails the system.

In phase 1, a cropped region from the frame captured by the image sensor is first downsampled by a dimensionality reduction algorithm and the extracted features are then fed into the ONN. If a face is detected, processor 1 wakes up processor 2. Processor 2 takes the entire frame captured by the image sensor and executes a more accurate face recognition neural network.

We consider two practical scenarios for applying the proposed system to real-life conditions. The first scenario, **ONN-C**(entered), requires continuous monitoring of the surroundings and assumes the users are able to see and proactively align themselves in the center of the camera's field of view. This scenario aligns with assumptions made in [9] and applies to cases such as smart door locks or attendance recording. ONN-C can safely assume that the user appears in the middle and performs only one inference using a center crop in the image. Processor 2 uses a digital face detection DNN to find the face bounding box in the original frame.

The second scenario, **ONN-U**(n)**A**(ligned), considers the proposed system working with additional sensors, such as accelerometers. This scenario aligns with assumptions made in [8] and applies to use cases including laptop or smartphone unlocking. The sensors activate the ONN under certain conditions and the user's face location at this moment varies in the image. Therefore, ONN-UA subsamples image patches from the original frame using a sliding window and detects if a face is present.

After locating the face in the original frame, the face recognizer model compares it with an existing database. A face is recognized when the distance between its embedding and any embedding from the database is below a certain threshold [2].

3.2 Datasets

In this work, we consider two steps toward the proof-of-concept of using optical processors in deep learning on computer vision tasks, especially face detection. The first step consists of using well-established artificial benchmark datasets, such as MNIST, Fashion-MNIST, and CIFAR-10, to validate the feasibility of optical processors on computer vision tasks. Next, face detection and face recognition datasets are used to further demonstrate the potential of optical processors in image/face classification tasks in real-life scenarios.

3.2.1 Reference Datasets

The following datasets are selected for reference in comparing the performance between digitally trained NNs and ONNs.

MNIST [50]: The MNIST dataset contains images of handwritten digits from zero to nine in grayscale. Each image is 28×28 pixels. The dataset is balanced with 6,000 images per class in the predefined training set and 1,000 images per class in the test set.

Fashion-MNIST [51]: Similar to the MNIST dataset, the Fashion-MNIST dataset contains images from 10 classes of fashion items, such as t-shirts, trousers, and sneakers. Each greyscale image is 28×28 pixels. The training and test sets contain 60,000 and 10,000 instances that are uniformly distributed across the 10 classes.

CIFAR-10 [52]: The CIFAR-10 dataset contains images in 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck), each with three color channels (red, green, and blue). Each image is 32×32 pixels. Though the original CIFAR-10 dataset contains 10 independent classes, we rearranged the labels to make the classification binary by aggregating the original label of "airplanes", "cars", "ships", and "trucks" into a new group called "vehicles", and "birds", "cats", "deer", and "dogs" into a new group called "animals". The images originally labeled as "frog" and "horse" are removed from the data set to ensure the balance between data samples between the two classes. This reduces the total image number to 48,000 with 24,000 images in each category.

3.2.2 Face Detection Dataset

The WIDER Face dataset [53] comprises 32,203 images with 393,703 faces labeled with bounding boxes. There are also various scenarios present in the images, including different poses, facial expressions, and occlusions. The dataset is split into training, validation, and test sets with a ratio of $4:1:5$. The bounding box information in the test set is not publicly available; therefore, we use the training and validation sets for our evaluations.

In this work, we simplify the definition of face detection to a binary classification task: *is a large enough face present in the image?* As a result, we need to create a new set of datasets from the original WIDER face dataset. Moreover, subject to the selected dataset, we define a "large enough" face to be one with a height in pixels in the range $[50, 200]$.

A sliding window of size (image_size $\times$ image_size) pixels moves horizontally and vertically across the image with a step size of $\frac{\text{image\_size}}{2}$ pixels. Since for each image, we have the coordinates of the bounding boxes of each face, the starting point of the crop is the top left corner of the top leftmost bounding box. Similarly, the endpoint is the bottom right corner of the bottom rightmost bounding box. An extra $\frac{\text{image\_size}}{2}$ pixels is added to the four sides of the traversal range to ensure that faces on the edges can be fully captured and there is a variety of partial face positions in the image patches. We define the image_size to be 100 px, as this preserves the facial features in each cut.

Each image patch is labeled based on the Intersection over Union (IoU) value (Eqn. (3.1)) between the sliding window and the bounding boxes of the nearby faces. We adopt the WIDER Face methodology and set the threshold to be 0.5: any image with IoU ≥ 0.5 will be labeled '1' and otherwise '0'.

$$IOU = \frac{\text{Area of Overlap}}{\text{Area of Union}}. \tag{3.1}$$

A prominent problem as a result of the cropping algorithm is that there are many more image patches labeled "0" than those labeled "1". With the first 1,348 images from the original validation set, 66,659 image patches were created and only 8,546 of them are

labeled "1". We consider another sliding window approach to selectively remove label "0" images from the dataset to resolve the problem of the unbalanced dataset. Each time, the sliding window looks at 5 consecutive labels and if there is no label "1" present, it removes the first three labels and their corresponding image patches.

After the entire dataset is created, the dataset is split into training, validation, and test sets with a ratio of $7:2:3$. To ensure that image patches cut from a single image are not spread across different sets, these ratios are in terms of the number of original images. For each original image, the number of image patches created is dependent on the number of faces it contains, ranging from around 150 image patches with more than 20 targeted faces to around 10 images when there is only one face. Taking into account this difference, the dataset division at the image patch level is $4.5:1:2.7$.

3.2.3 Face Recognition Dataset

The Labeled Faces in the Wild (LFW) dataset [14] is a renowned dataset for face verification and recognition tasks. It contains 13,233 250×250 px images, with 5,479 distinct people depicted. The original dataset contains instances of faces that are grouped into pairs and labeled with the fact that the two faces belong to the same person. There are also different versions of the dataset with the faces in an unconstrained environment or funneled for face alignment.

The LFW dataset is only used as the test set in this work, as it contains fewer images compared to other academic benchmarks. Instead of grouping images into pairs, we consider only the face images of distinct people present in the dataset and create a database with these faces. The same set of modifications are applied to the images as the WIDER Face dataset. They are cut into patches with a 100×100 px sliding window and the face presence labels are determined by the IoU in the same way as the WIDER Face. To ensure the best generalization of neural network models over the two face detection and face recognition datasets, we apply min-max normalization to the cropped WIDER Face and LFW datasets to limit the pixel values in the range $[0, 1]$.

3.3 Dimensionality Reduction

Though low resolution, images we consider for this work are still too large to be directly fed to an ONN, recalling that the maximum number of inputs we considered is 64 [54]. Taking the CIFAR-10 images as an example, each image comprises $32 \times 32 \times 3$ pixels or 3072 features. Hence, we need a dimensionality reduction methodology to extract the most representative features of image information.

3.3.1 Principal Components Analysis

In general, PCA maps the n-dimensional data into a k-dimensional subspace ($k << n$) by finding the eigenvectors that best represent the feature distribution in an image, known as the principal components [32]. Normally, applying PCA requires a preprocessing step of normalizing the dataset, as shown in Eqn (3.2),

$$x_j^{(i)} = \frac{x_j^{(i)} - \mu}{\sigma_j}, \tag{3.2}$$

where $x_j^{(i)}$ is the i-th feature of data sample x_j, μ is the sample mean of all data points and σ_j is the standard deviation of the j-th data sample. However, since we have already normalized the image data, as discussed in Section 3.2.2, into the range $[0, 1]$, this additional preprocessing is omitted as we already know the apriori distribution of data points.

In the next step, the most representative eigenvectors are obtained by sorting the eigenvalues of the data in decreasing order and selecting the eigenvectors corresponding to the top k eigenvalues. Considering that the pixel arrays are not always square, we use SVD in place of eigenvalue decomposition,

$$W = U\Sigma V^T, \tag{3.3}$$

where U and V are orthogonal matrices and Σ is a diagonal matrix. The non-zero elements of Σ are the positive square roots of eigenvalues obtained from WW^T and W^TW,

known as the singular values. The higher the singular value, the more variance in data points can be seen in the corresponding eigenvector directions, and therefore, the corresponding eigenvector is more representative. To finish the transformation, the selected top-k eigenvectors are concatenated and multiplied with the original matrix.

3.3.2 Fast Fourier Transform

Fast Fourier Transform (FFT) converts the images to the frequency domain and reflects the magnitude of variations of image data in each direction. The 2D FFT is applied to the images after they are converted to greyscale (which will be discussed in Section 3.4):

$$c(k_x, k_y) = \Sigma_{m,n} e^{jk_x m + jk_y n} g(m, n), \tag{3.4}$$

where $g(m, n)$ is the pixel at location (m, n) mapping to the location k_x, k_y in the Fourier image, and $c(k_x, k_y)$ is the corresponding Fourier coefficient [46]. More specifically, (k_x, k_y) refers to the same location as (m, n) in the image, but its origin starts at the center of the image. The transformation defined in Eqn. (3.4) perserves the size of the image. Therefore, to reduce the dimensionality, we perform a center crop of size L to the transformed frequency domain image. The $L \times L$ cropped coefficients are stacked to form a $L^2 \times 1$ feature vector.

FFT preprocessing is particularly suitable for ONNs because of the use of Fourier optics including components such as the lens and spatial filters [46]. These devices can passively perform the transformations without spending extra power. Moreover, since ONNs operate on complex numbers, the complex-valued Fourier coefficients can be directly handled, without isolating the imaginary and real parts as done in the digital neural networks.

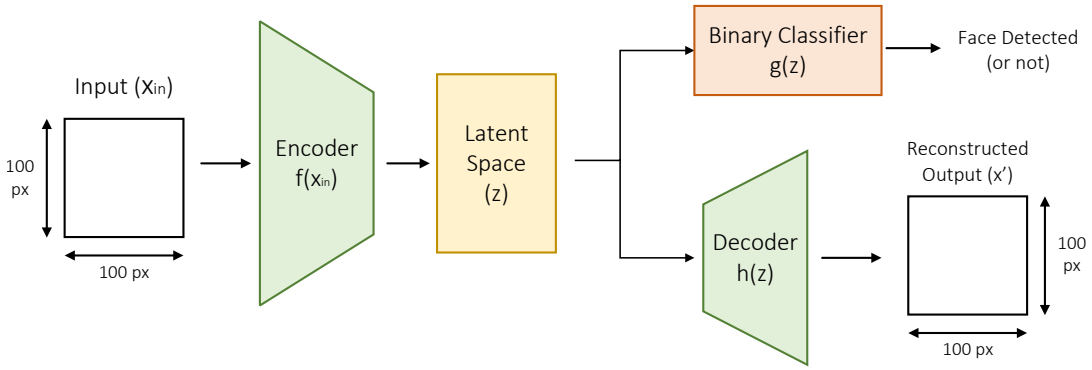


Figure 3.2: The DAE multi-tasking pipeline: latent space obtained from the encoder is sent to the decoder and a binary classifier simultaneously.

3.3.3 Autoencoders

An autoencoder is a type of unsupervised generative model for creating a set of outputs that are certain transforms of the input. In our case, we train the autoencoder with a target of the output being identical to the input after stages of transformation.

The autoencoder consists of two sets of neural networks, an encoder that applies a function f to the input x_{in} such that the output $z = f(x_{in})$ and a decoder that applies a function g to the output from encoder such that $x' = g(z) = g(f(x_{in}))$. The optimization goal is to minimize the distance between x_{in} and x', meaning that the reconstructed input is as similar as possible to the original input. The encoder is responsible for extracting the key properties from the input, which is also known as the latent space. The latent space is therefore smaller than the input space and can be used as the result of dimensionality reduction. The decoder reconstructs the input from the latent space by adding the dimensions in the reversed pattern.

To better train the the encoder for face detection, we also consider adding a binary classifier with function $y_{\text{pred}} = h(z)$ to the encoder. Backpropagation uses the loss resulting from both image reconstruction and face detection:

$$L_{\text{total}} = L_{\text{recons}}(x_{\text{in}}, x') + L_{\text{classify}}(y_{\text{true}}, y_{\text{pred}}). \tag{3.5}$$

In this work, we focus on one specific autoencoder architecture, DAE, with different layer compositions. The images from the reference datasets are fed to DAEs with either solely fully connected layers or convolutional layers when they are colored. Meanwhile, the face images are only trained on convolutional networks to best capture the spatial relationships between pixels and avoid massive fully connected neural networks which lead to overfitting concerns in training and potentially, scalability concerns during their implementations on edge devices. Each DAE has a 5-layer encoder; each layer reduces input size by half and doubles the number of channels when the image is not monochromatic. Its decoder mirrors this in reverse. The classifier for multi-tasking has two identical fully connected layers of $K \times K$ neurons, where K denotes the number of extracted features, and ReLU activation, and the multi-tasking pipeline is shown in Fig. 3.2. We apply early stopping during training with a patience window of 5 epochs.

3.4 Image Preprocessing - RGB to Greyscale

In the very first step of FFT for dimensionality reduction, we convert the colorful images to greyscale according to CCIR 601 [55]. The colors are first gamma corrected by applying a factor $\gamma = 2.2$ to each channel ($R' = R^\gamma$, $G' = G^\gamma$ and $B' = B^\gamma$). The values are then normalized to the range $[0, 1]$ instead of $[0, 255]$ in the original RGB image. Next, the lightness is computed by Eqn (3.6).

$$Y = 0.2126 \cdot R^\gamma + 0.7152 \cdot G^\gamma + 0.0722 \cdot B^\gamma, \tag{3.6}$$

Finally, the "luminance" of an image is calculated by

$$L = 116 \cdot Y^{\frac{1}{3}} - 16, \tag{3.7}$$

This L value is then the pixel intensity represented by the RGB channels.

3.5 False Negative Reduction

A key challenge in the implementation of two-phase models as we propose is phase 1 FNs: if a face is present but not detected, facial recognition fails by default. Therefore, the goal of our work is to minimize the number of FNs while maintaining the classification accuracy of ONN.

We considered two methods for this purpose. The first changes the IoU threshold defined for '1' labels when creating the dataset to count partial faces. In this work, two new datasets, with IoU $\geq$ 0.2, and 0.1 are created using the same routine as described in Section 3.2.2. The IoU $\geq$ 0.2 dataset allows half-faces to be counted towards valid faces while the IoU $\geq$ 0.1 dataset considers all partial faces to be valid. This allows for more positive labels within the same set of image patches and models trained on these datasets tend to predict positive labels more often than the one trained on the IoU $\geq$ 0.5 dataset.

The second method changes the weight assigned to each label class in the loss function. We penalize FN more severely, and the binary cross-entropy loss becomes

$$L_{\text{BCE}} = -\beta y \cdot \log(\hat{y}) - (1-y)\log(1-\hat{y}), \tag{3.8}$$

where β is a constant greater than 1, y is the ground-truth label and $\hat{y}$ is the output from classifier after Sigmoid activation

$$\hat{y} = \frac{1}{1+e^{-y'}}, \tag{3.9}$$

where y' is the output layer outcome. Consequently, the gradient of the loss function becomes

$$\frac{\partial L_{\text{BCE}}}{\partial w} = \text{conj}(-\beta y + (\beta - 1) \cdot y\hat{y} + \hat{y}) \cdot z, \tag{3.10}$$

where z is the input to the ONN and the complex conjugate is taken for complex-valued neural networks.

3.6 System Power and Energy Analysis

We take a probabilistic approach to modeling system behavior: the cost (in power, energy, or time) of one inference is decomposed into the cost of the ONN plus the cost of the digital DNN(s), times the probability of it being triggered ($p(\text{trigger})$).

In ONN-C, $p(\text{trigger})$ depends on the chance of a face appearing in the camera's field of view ($p(\text{face})$) and the probability of FP $p(FP)$ and True Positive (TP) decisions $p(TP)$ made by ONN: Hence,

$$p(\text{trigger}|N) = (p(TP) - p(FP)) \cdot p(\text{face}) + p(FP), \tag{3.11}$$

where N is the number of input features, and for one inference

$$C = C_1 + p(\text{trigger}|N) \cdot C_2, \tag{3.12}$$

where C is the power/latency/energy consumption of the entire system, C_1 and C_2 are the corresponding consumption of phases 1 and 2 in Fig. 3.1. In this work, we assume $p(\text{face}) = 1\%$, which corresponds to around 14 minutes per day.

In ONN-UA, we assume that faces are always present when the camera starts capturing its surroundings ($p(\text{face}) = 1$). Therefore, $p(\text{trigger}|N) = p(TP) =$ accuracy, since all the images in the ONN-UA face detection test set contain a face (corresponding to the $p(\text{face}) = 1$ assumption). A more detailed description of this test set will be provided in Section 4.3.2. Moreover, in this case, the cost of the ONN (C_1) is the cost of a single inference times the average number of inferences before the ONN detects a face ($n(\text{trial})$). Therefore,

$$C = C_1 \cdot n(\text{trial}) + p(\text{trigger}|N) \cdot C_2. \tag{3.13}$$

The actual values of C_1 and C_2 are obtained from established work. We estimate the power, energy, and latency of 1) the ONN using data provided in [46], and 2) Jetson Nano using [56]. Memory performance is estimated with CACTI-7.0 [57].

3.7 Performance Evaluation Metrics

The performance of the ONNs and the entire systems are evaluated based on their classification/recognition accuracy and power/latency/energy consumption as calculated in Section 3.6. All evaluations begin with plotting the confusion matrix of the test set to visualize the number of TP, FN, FP, and True Negative (TN) cases. The subsequent metrics are calculated to further quantify the classification performance.

3.7.1 Accuracy

The accuracy refers to the fraction of time when the model correctly predicts the ground-truth label. In binary classification cases, it is calculated using the equation

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + FN}. \tag{3.14}$$

More generally, for all classification cases regardless of class size,

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Number of test set instances}}. \tag{3.15}$$

Due to the multi-stage nature of the design in Fig. 3.1, we calculate three sets of accuracy for the system: the ONN face detection (phase 1) accuracy, the standalone face recognition (phase 2) accuracy, and the end-to-end face recognition accuracy. The former two sets of accuracy can be calculated by applying Eqn (3.14). Meanwhile, the confusion matrix of the entire system can be computed by combining the confusion matrices

obtained from phases 1 & 2 alone:

$$\text{TP}_{\text{total}} = \text{TP}_2,\ \text{FP}_{\text{total}} = \text{FP}_2, \tag{3.16}$$

$$\text{FN}_{\text{total}} = \text{FN}_1 + \text{FN}_2,\ \text{TN}_{\text{total}} = \text{TN}_1 + \text{TN}_2, \tag{3.17}$$

and the corresponding values are substituted into Eqn (3.14) to find out the overall accuracy. By observing Eqn (3.17), we can also find out that the FNs in the first stage directly counts towards the total system errors without any chance of correction by phase 2. Therefore, reducing FNs in ONN is critical to achieving high end-to-end face recognition accuracy as stated in Section 3.5.

3.7.2 F1 score

The F1 score (Eqn (3.18)) is the harmonic mean of precision (Eqn (3.19)) and recall (Eqn (3.20)) of the system. It provides a balanced measure of the model's performance and is particularly valuable for cases where we have unbalanced datasets between classes, in our case, the slightly unbalanced face detection dataset. This score will be only applied to binary classification scenarios.

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{3.18}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{3.19}$$

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{3.20}$$

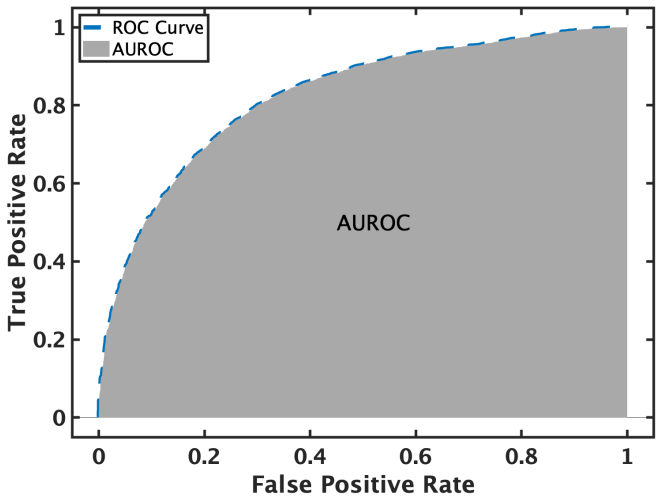


Figure 3.3: The ROC curve (blue) with AUROC area labeled in grey.

3.7.3 Area Under the Receiver Operating Characteristics (AUROC)

The Area Under the Receiver Operating Characteristics curve (AUROC) is another fundamental metric in binary classification tasks, used for evaluating the effectiveness of a model in performing the task. The evaluation begins with depicting the Receiver Operating Characteristics curve (ROC) (as shown in Fig. 3.3) that contrasts the True Positive Rate (TPR) (a synonym of Recall in Eqn (3.20)) and False Positive Rate (FPR) (Eqn (3.21)) of the model at all classification thresholds.

$$\text{FPR} = \frac{FP}{FP + TN}. \tag{3.21}$$

The classification threshold is the value of probability predicted by the model beyond which can be considered as a positive label. Lowering it will lead to more data instances classified as positive, hence leading to more TPs and FPs.

AUROC is the entire two-dimensional area enclosed by ROC and two axes. It can be seen as the probability that a model ranks a random positive data instance higher than a random negative one [58].

3.8 Pareto Optimality

The goal of our work is to find an electro-optic hybrid system that achieves the best trade-off between face recognition performance and the system's power/energy efficiency. Therefore, the design problem can be transformed into a multi-objective optimization problem using a grid search over the hyperparameters. The two objectives are 1) the high accuracy/low error rate of the end-to-end face recognition system, and 2) the low power/energy consumption of the system, averaged to one inference. The Pareto optimal solutions to this multi-objective optimization problem can then be described as the models whose performance cannot be surpassed by another in one objective without degradation in performance of the other [59].

In the ONN-C case, the first objective refers to the face recognition system accuracy while the second objective can be analogized by $p(\text{trigger}|N)$, as a result of the linear relationship between it and the total consumption of system shown in Eqn (3.12). More specifically, given a Pareto optimal solution model m in search space M, with error rate $E(m)$ and probability of triggering second phase $p(\text{trigger}|N, m)$, it dominates the other solutions $n \in M$ by

$$\begin{aligned} m \prec n \iff & E(m) < E(n) \ \wedge \ p(\text{trigger}|N, m) \leq p(\text{trigger}|N, n) \vee \\ & E(m) \leq E(n) \ \wedge \ p(\text{trigger}|N, m) < p(\text{trigger}|N, n). \end{aligned} \tag{3.22}$$

Multiple Pareto optimal solutions can coexist for the same optimization problem and by aggregating them, we obtain the Pareto optimal set S,

$$S = \{s \in S \mid \ s \in M, \ \nexists n \in S \text{ s.t. } n \prec m\}, \tag{3.23}$$

which are the model designs that fit our goal.

In ONN-UA, the first objective also refers to the face recognition system accuracy. However, in this scenario, $p(\text{face}) = 1$, and $p(\text{trigger}|N) = p(TP) = \text{accuracy}$. $p(\text{trigger}|N)$

is now identical to the metric we used for describing the first objective, and no comparisons can be made. Therefore, $n(\text{trial})$ will replace it to be another factor influencing the power and energy consumption pattern as indicated in Eqn (3.13).

Chapter 4

Experimental Setup

We conduct a set of experiments to prove: (1) the ability of ONNs in performing object/face detection tasks, and (2) the feasibility of the proposed multi-stage face detection and recognition system in both lossless and lossy environments. The stagewise experiments begin from the search for the best dimensionality reduction methods among the proposed ones in Section 3.3. The best dimensionality reduction method should yield the highest ONN accuracy based on the extracted features. Next, we use the selected dimensionality reduction method to preprocess data instances from the proposed datasets in Section 3.2 and train the ONNs on the extracted features. ONNs are configured with different combinations of hyperparameters and their classification performances, including accuracy, F1 score, and AUROC, on the validation and test sets are compared. Selected best-performing ONNs are combined with SOTA deep neural networks to complete the face recognition workflow. We compare the end-to-end face recognition accuracy and the power/energy consumption of the entire system with different hyperparameter settings and obtain the Pareto Optimal solution set subject to the perfect operating conditions. Finally, we inject non-ideal conditions to the Pareto Optimal solution set and observe each solution's response to the noise.

4.1 Simulation Framework

We perform both ex-situ and in-situ training in this work to evaluate the ONNs. The ex-situ training of ONNs assumes noise-free operating conditions with high arithmetic precision (64-bit) and directly updates the digital weights during backpropagation. Therefore, it is only used for investigating the effectiveness of dimensionality reduction methods. The simulations are implemented using PyTorch. To align the ex-situ trained ONNs with in-situ trained ones, the parameters used in neural networks are set to be in complex numbers and the bias per layer is eliminated. The random seeds for each experiment are fixed to allow fair comparison.

The in-situ training of ONNs uses phase shifts as the parameter to be updated during backpropagation. The current version of in-situ simulations is built on the Neuroptica [60] package.

4.2 Efficiency of Dimensionality Reduction

We begin by searching for the most efficient dimensionality reduction methods among PCA, FFT, and autoencoders. To achieve this, ONNs with a fixed set of hyperparameters are trained on the extracted features from each dimensionality reduction method with a set of random seeds. The averaged classification performances of ONNs then serve as indicators of dimensionality reduction method efficiency.

4.2.1 Experiment Workflow

Fournier and Aloise [61] first presented a comparison between the SVD-based dimensionality reduction methods, such as PCA and isometric feature mapping [62], and the autoencoders by testing their classification accuracies over the MNIST, Fashion-MNIST, and CIFAR-10 datasets. The extracted features from the dimensionality reduction methods are fed into a simple classifier model, K-Nearest Neighbors (kNN), to obtain the clas-

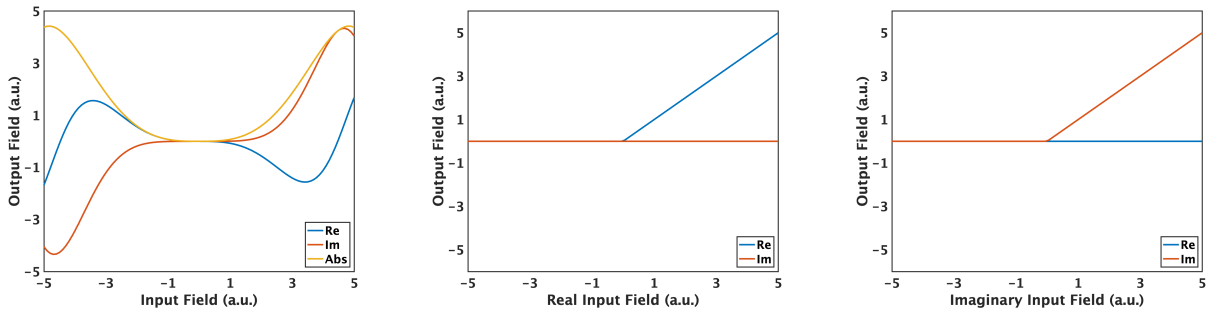
sification results. However, limited by the complexity and learning ability of KNN, the classification accuracy decreases below 50% as the complexity of the dataset increases and the subsequent comparisons become trivial.

Dimensionality Reduction on Reference Datasets

In this work, we follow a similar procedure as [61] for dimensionality reduction tests on the reference datasets and replace the KNN with ONNs using a fixed set of hyperparameters.

The parameter to be selected in PCA is the number of features (N). In this case, $N \in [10, 64]$. This not only considers the upper limit of the ONN scalability [54] but also ensures a sufficient number of features to be used for classification tasks. Similarly, the only factor involved in the FFT method is the half-feature length $L = \frac{\sqrt{N}}{2}$, which is the number of pixels to be cropped from the center of the image. This value is set to be in the range $[2, 4]$ for the same reason as above. When using autoencoders for dimensionality reduction, each encoder-decoder architecture was first trained with Adam optimizer [63] and the reconstruction loss function. An early stopping technique with a patience window size of 5 is applied, such that when the validation reconstruction loss does not decrease within 5 epochs, the training process stops. The upper limit of training epochs is set to 100. The transformed inputs are derived from the latent space, by reshaping it into a vector.

We directly input the reconstructed dataset after dimensionality reduction into both ex-situ and in-situ ONNs. In both cases, the classification network is fixed to 2 layers of N input and output neurons. We tested over the Reck, Clements, Diamond, and Bokun topology. The electro-optic non-linear function (Fig. 4.1a) [46] and the complex-valued ReLU (cReLU) function [64] (Fig. 4.1b for the real part and Fig. 4.1c for the imaginary part) are inserted between layers for activation and all neural networks are terminated with a squared normalization layer which imitates a photodiode input. For a k-class classification task, readings from the first k output ports are obtained. Each selected port



(a) EO activation **(b)** Re(cReLU) **(c)** Im(cReLU)

Figure 4.1: Activations functions to be used in ONNs. The cReLU function is separated into real and imaginary parts.

corresponds to a class in the dataset and its reading represents the probability of the given input belonging to the corresponding class. The Adam optimizer [63] and stochastic gradient descent are used in all neural network training procedures with a learning rate of 0.005 for 20 epochs. Each dataset is divided into training, validation, and test sets with a proportion of $8:1:1$, and each set is split into 10 equal batches during training.

Dimensionality Reduction on Face Detection Dataset

The test of dimensionality reduction methods on the face detection dataset follows the same procedure as the reference datasets. However, for each method, a fixed number of 64 features are extracted and used solely in the ex-situ training of ONNs with two fully connected layers of 64 input and output neurons. We leave the feature number selection to the later stages of hyperparameter tuning of face detection models. The first two neurons from the output layer of the ONN are used for the final classification decision. We apply early stopping during training with a patience window of 5 epochs and train up to 80 epochs.

4.2.2 Evaluation Metrics

Autoencoder Performance

The varied versions of autoencoders are first compared and the best-performing model is selected for further comparison with other dimensionality reduction methods. The test set reconstruction loss is the most intuitive indicator of the effectiveness of autoencoders. We also take into account the test accuracy of ex-situ ONNs trained on the extracted features and compare it with the accuracy of the auxiliary classifier if multi-tasking is used.

Dimensionality Reduction Effectiveness on Different Datasets

The MNIST and Fashion-MNIST datasets are designed for 10-class classification tasks and are artificially balanced among classes. Therefore, in this case, we only look at the overall test accuracy of ex-situ and in-situ models trained on the two datasets for comparison.

We redesigned CIFAR-10 dataset into a binary classification task. Therefore, in addition to the test set accuracy, we calculate the F1 score of the ONN predictions and compare them among methods. Since the new CIFAR-10 dataset is also artificially balanced between classes, we should expect close proximity between the accuracy and F1 scores.

The face detection dataset is not perfectly balanced despite having implemented the data selection method. Therefore, it is important to look at both the accuracy and the F1 score to evaluate the ONN's ability to discriminate between both classes. Moreover, the AUROC of each model is also compared to evaluate the ONN performance under different decision threshold values.

4.3 Classification Performance

4.3.1 Hyperparameter Tuning in Face Detection Model

The face detection stage is implemented in the optical domain with the ONNs. The most effective dimensionality reduction method selected from the previous stage is used for

Table 4.1: Hyperparameters Tuned during Face Detector Training

Hyperparameter	Values/Category
Number of Layers	1, 2, 3
Features	8, 16, 32, 64
Activation	Electro-Optic (EO)
Dataset	IOU $\geq \{0.1, 0.2, 0.5\}$
Topology	Bokun, Clements, Diamond, Reck
β	1, 1.2, 1.4, 1.6, 1.8

generating the extracted features for training, validation, and test sets. Next, these features are fed to the ONNs for training and evaluation. Similar to the ex-situ training, the first two output port (ports O_0 and O_1 in Fig. 2.3) readings are used for the binary decision of a face present or absent. After each training iteration, backpropagation is done by calculating the weight update values in phase angles using the approach in [44]. 10 splits are applied to the dataset to create batches, and cross-entropy loss is used with stochastic gradient descent.

The hyperparameters to be tuned (as summarized in Table. 4.1) include the number of layers ($[1, 3]$), the input/output size of each layer ($[8, 64]$), and the optical processor topology (Reck, Clements, Diamond, and Bokun). EO activation is used for all models, as it has actual hardware implementation data allowing us to perform power/energy consumption analysis. The models are trained on datasets with different IoU thresholds or different weights ($\beta \in [1, 2]$) assigned to the positive class. A fixed learning rate of 0.005 is applied to all cases.

In the ONN-UA cases, the cropped images are applied in series as the sliding window cuts across the original image, starting from the top left to the bottom right corner. We not only determine whether the face detector can detect a valid face but also calculate the average number of sliding window cuts ($n(\text{trial})$) required before the face detector signals a valid face. In the ONN-C cases, the center-cropped original image is sent to the detector.

For one image, ONN-UA tries multiple times to detect a face while ONN-C relies on one shot.

4.3.2 Evaluation Metrics for Face Detectors

As summarized in Table 4.2, all the ONN face detection models are tested on three different sets:

1. the **local** test sets which are the corresponding test sets of different IoU thresholds,
2. an **In-distribution (ID)** IoU ≥ 0.5 test set (with the same data distribution as the training set), and
3. an **Out-of-distribution (OOD)** test set, which is created from the LFW dataset with IoU ≥ 0.5 cropping.

As a result of the label '0' elimination step in the local and ID test sets, the stored images are no longer consistent sliding window crops. Therefore, they will not be applied to the ONN-UA scenario which requires sequential image input. The composition of OOD set also differs in the two scenarios: in ONN-UA, there are two sets of labels considered, as shown in Fig. 4.2. At a macro level, all the images from the original LFW dataset contain a face, meaning that the ground-truth labels are all "1". This corresponds to the assumption that $p(\text{face}) = 1$. However, during the face searching process of ONN, it cuts out image patches that may not contain a face. Therefore, at a micro level, the test set contains label-"0" no-face patches and label-"1" with-face patches. For the reporting of model performance, we calculate both the macro-level and micro-level accuracy: the

Table 4.2: Summary of Face Detection Test Set Information

Name	Source Dataset	Cropping IoU threshold	Scenarios Tested
Local	WIDER Face	[0.1, 0.2, 0.5]	ONN-C
ID	WIDER Face	[0.5]	ONN-C
OOD	LFW	[0.5]	ONN-C & ONN-UA

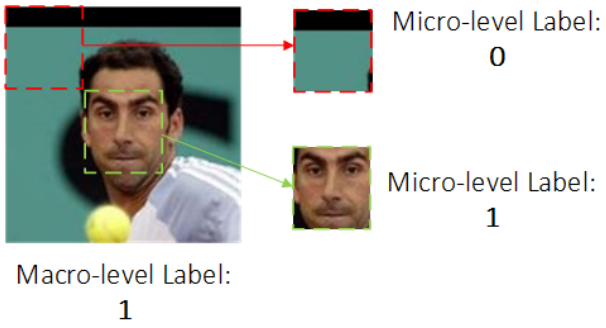


Figure 4.2: Different Levels of Labeling in the Face Detection Dataset of ONN-UA

macro-level accuracy is the number of 250×250 px images on which ONN successfully detects a face after scanning through them, and the micro-level accuracy is the number of 100×100 px image patches on which the ONN makes the correct decision on whether there is a face. As we assume the entire face detection system is triggered when a user is present (i.e., $p(\text{face}) = 1$) for this scenario, the macro-level test set accuracy will be exactly $p(\text{trigger}|N)$ as computed in Section 3.6. In ONN-C, the label '1' images are created from a center crop, and the label '0' images are created from partial-face images from surrounding cuts and no-face images from IoU ≥ 0.1.

For each model with a different combination of hyperparameters, we calculate its accuracy, F1 and AUROC scores. A model is trained five times with different random seeds and the evaluation metrics obtained from each seed are averaged. At this stage of investigation, we do not pick out the best-performing models, instead, we only eliminate a few combinations of hyperparameters that only yield random-guess level ($\leq$60% ID test accuracy) classification performance or have shown strong overfitting trend during the training process.

The ONN-C and ONN-UA face detectors consider the same set of training hyperparameters and training data. Therefore, we assume that the ID test performance on ONN-C face detectors is representative enough such that we can rely on it to eliminate infeasible hyperparameter sets for both use cases.

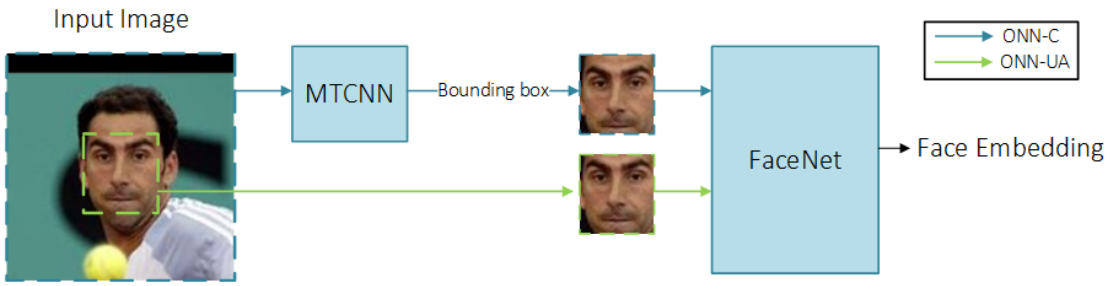


Figure 4.3: The Workflow of Tests on the Face Recognition and Entire System

4.3.3 Combining Face Detection with Face Recognition Models

In the next step, we combine the ONN face detectors with digital face recognition models and evaluate the performance of the entire event-driven system, using the workflow shown in Fig. 4.3. Devising a state-of-the-art model for face recognition is beyond the scope of this work: therefore, we adopt FaceNet [2] pre-trained on the VGGFace2 dataset. The set of hyperparameters for FaceNet is therefore fixed.

At this stage, we use a test set that is created in the same way as the OOD test set from face detection stage; however, to avoid confusion with the face detection results, we simplify its name to **"LFW test set"**. Furthermore, the 100×100 px LFW test set created from cropping is called the "new LFW test set" and the original 250×250 px LFW dataset is called the "original LFW test set".

We first obtain the face embeddings of the image patches marked positive by ONN. The new LFW test set is applied to all selected models from face detection and all the images that are marked as "face present" are stored. In the ONN-UA cases, the first positive-marked new LFW test set instance is directly sent to FaceNet to obtain the face embedding. In the ONN-C case, we use Multi-Task Cascaded Convolutional Neural Networks (MTCNN) [19] to figure out the 100×100 px bounding boxes of the old LFW test set instances that correspond to the positive-marked new LFW test set instance. The bounding box regions are cut out and sent to FaceNet for face embeddings.

The real-time face embeddings are pair-wise compared with existing embeddings in the database. The Euclidean distance between the two embeddings is used as a metric for

proximity and we select the most appropriate threshold value based on the accuracy of comparisons.

4.3.4 Evaluation Metrics for Face Recognition and Entire System

The database of face embeddings for phase 2 is created by computing the embeddings of all images in the 100×100 px center-cropped face-aligned original LFW test set. To avoid repetitive inference during the grid search over the hyperparameters, the real-time face embeddings of both the new LFW test set used in ONN-UA and the MTCNN-cropped original LFW test set used in ONN-C are calculated in advance to runtime. To further save storage space, the binary outcome of whether these new face embeddings match with any of the embeddings in the database is computed and stored in a look-up table format with the same sequence as the test set. The accuracy of stage 2 can be simply calculated by summing all elements in the look-up table. Finally, we compute the end-to-end system accuracy by combining the standalone face detection and face recognition results based on the confusion matrix constructed by Eqns (3.16) and (3.17).

Using the confusion matrix from face detection, we can also calculate $p(\text{trigger}|N)$ of each model. Combined with the entire system accuracy, we project all the models to the two-dimensional objective space as described in Section 3.8. The models in the Pareto optimal solution set are considered the best models under perfect operation conditions.

4.4 Effect of Imperfect Devices

As described in Section 2.2.3, we focus on two device imperfection factors, the deviation in phases programmed to phase shifters and the propagation loss per MZI. We apply two sets of sensitivity analyses to the model with different levels of device imperfection.

The first set of sensitivity analysis looks at the impact of topology on the resilience of the ONN models towards device imperfection. We focus on the ONN-C scenario during our tests to save simulation time by avoiding the repetitive inference of sliding windows.

During sensitivity analysis, the distribution of phases programmed to phase shifter in radians is modeled with Gaussian distributions, such that $\theta \sim N(\theta_i, \sigma_\theta)$ and $\phi \sim N(\phi_i, \sigma_\phi)$ for the internal and external arms. θ_i and ϕ_i are the phase shifts obtained from the in-situ training of ONNs under perfect conditions. The values for σ_θ and σ_ϕ are uniformly sampled from the interval $[0, 1]$. Similarly, the propagation loss per MZI in dB is sampled from the interval $[0, 1]$ and applied to the system by altering the transfer matrix

$$D'_{MZI} = je^{j(\frac{\theta}{2})} \begin{bmatrix} 10^{-\text{loss}/10} & 1 \\ 1 & 10^{-\text{loss}/10} \end{bmatrix} \cdot \begin{bmatrix} e^{j\phi}\text{sin}(\frac{\theta}{2}) & e^{j\phi}\text{cos}(\frac{\theta}{2}) \\ \text{cos}(\frac{\theta}{2}) & -\text{sin}(\frac{\theta}{2}) \end{bmatrix}. \tag{4.1}$$

Different pairs of [σ_θ, σ_ϕ] and [$\sigma_\theta = \sigma_\phi$, loss] are applied to the trained model and we monitor the changes in accuracy, number of FN and FP cases, and resultant $p(\text{trigger}|N)$. For each σ_θ or σ_ϕ value, we sample 20 different θ and ϕ values from the corresponding Gaussian distribution, apply them to the system, and take the average of the results. To quantitatively analyze the robustness of each model, we introduce a Figure of merit (FoM) where we count the number of [σ_θ, σ_ϕ] or [$\sigma_\theta = \sigma_\phi$, loss] pairs that lead to less than 10% drop in the ONN or system accuracy with respect to the perfect condition accuracy. Models with a greater count are considered more robust as they tolerate a wider range of errors while achieving the same level of classification performance.

The second set of sensitivity analyses focuses on the Pareto optimal solutions selected from the perfect operating conditions. We assume a fixed insertion loss of 0.6 dB/MZI [65] and apply the analysis to both ONN-C and ONN-UA scenarios. The programmed phase is still modeled with a Gaussian distribution $(\theta, \phi) \sim N((\hat{\theta}, \hat{\phi}), (\sigma_\theta, \sigma_\phi))$ where $\hat{\theta}, \hat{\phi}$ are phases obtained after training in radians, and $\sigma_\theta, \sigma_\phi \in [0, 0.5]$ are the deviations (errors). The device imperfections are injected into the trained model to obtain the variation in the ONN accuracy, system accuracy, power, and energy consumption.

Chapter 5

Results and Discussion under Perfect Operating Conditions

To prove that the proposed design can save power and energy consumption while maintaining a high level ($\geq$ 90%) of face recognition accuracy, we perform a grid search over the dimensionality reduction methods and the identified hyperparameters in Table. 4.1 with the assumption of perfect operating condition. This assumes ideal fabrication conditions, negligible insertion loss or attenuation of light in waveguides, and the precise mapping of trained phase shifts to the on-chip phase shifters.

5.1 Dimensionality Reduction

Our discussion of results begins with the effectiveness of dimensionality reduction methods. The same set of dimensionality reduction methods are applied to both reference datasets (MNIST, Fashion-MNIST, and modified CIFAR-10) with different numbers of classes used for classification, and our ultimate goal of the face detection dataset. The numerical metrics, including classification accuracy, and F1 score, are compared among methods. We also reconstruct the images after transformations and compare them with the original images.

5.1.1 Comparison of Dimensionality Reduction Methods on Reference Datasets

Table. 5.1 shows all metric values obtained from using dimensionality reduction methods on the reference datasets with the maximum number of allowable input features and different combinations of hyperparameters. The entries in Italics indicate the result obtained from ex-situ training.

For both Clements and Reck topologies, a maximum number of 64 features can be used for successful classification as defined by the model correctly updating the weights and achieving more than random-guess level performance. The Bokun topology has an upper bound of 16 features for successful classification. Anywhere beyond the threshold leads to a model that experiences zero-gradient issues during backpropagation, meaning that there is no weight update during training, and eventually the model can only perform random guesses on the data instances. A similar problem occurs with the Diamond

Table 5.1: Performance of Dimensionality Reduction Methods on Reference Datasets

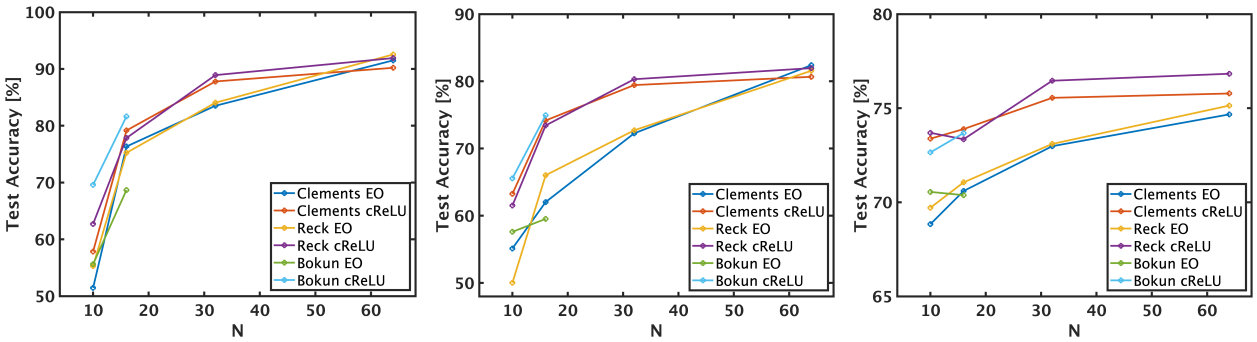
Dimensionality Reduction Method	Trainable Parameter	Topology	Features	Activation	MNIST	Fashion-MNIST	CIFAR-10	
					Test Accuracy [%]	Test Accuracy [%]	Test Accuracy [%]	F1
PCA	Weight	-	64	EO	*95.98*	*85.87*	*77.71*	*0.759*
	Phase Shift	Clements	64	EO	91.51	**82.40**	71.65	0.683
				cReLU	90.20	80.67	75.78	0.770
		Reck	64	EO	**92.55**	81.57	75.14	0.750
				cReLU	91.91	81.96	**76.83**	**0.752**
		Bokun	16	EO	68.67	59.53	70.38	0.430
				cReLU	81.62	74.95	73.86	0.716
FFT	Weight	-	64	EO	*97.31*	*88.58*	*66.64*	*0.664*
	Phase Shift	Clements	64	EO	91.07	**74.60**	**63.72**	0.616
				cReLU	90.17	70.49	60.81	0.593
		Reck	64	EO	**91.92**	73.98	63.16	**0.647**
				cReLU	91.12	59.30	61.39	0.609
		Bokun	16	EO	74.44	54.91	54.63	0.332
				cReLU	77.62	63.00	50.95	0.445
Autoencoder	Weight	-	64	EO	*90.40*	*79.90*	*70.91*	*0.699*
	Phase Shift	Clements	64	EO	**70.75**	**68.00**	47.96	0.538
				cReLU	67.48	67.18	**57.05**	**0.688**
		Reck	64	EO	61.92	57.76	55.77	0.673
				cReLU	60.20	64.29	47.80	0.096

Table 5.2: Results from Autoencoder Training on CIFAR-10

Dataset (Image Shape)	Layer Type	Reconstruction Loss	Test Accuracy [%]	Number of parameters
CIFAR-10 (3*32*32)	FC	193742.36	70.91	3842048
	Conv	733857.13	70.88	43624

mesh and more severely, the model learns nothing even when only four features are extracted. This limits the topology's ability to perform multi-class classification tasks on datasets such as MNIST and Fashion-MNIST. It is therefore removed from all subsequent discussions.

In general, as the complexity of the dataset instances increases, the accuracy of both ex-situ and in-situ ONNs drops. PCA and FFT have similar classification accuracy when trained with MNIST task. However, as the task becomes harder, the discrepancy in accuracy between the two methods expands to more than 13%, and PCA becomes a better choice for dimensionality reduction. Moreover, we observe that models trained with EO activations more frequently outperform their counterparts trained with cReLU activations. This suggests that the EO activation, which is tailored to the ONN designs [46], better improves the learning ability of in-situ trained models.



(a) MNIST **(b)** Fashion-MNIST **(c)** Modified CIFAR-10

Figure 5.1: The variation in ONN accuracy due to change in the number of input features with PCA as the dimensionality reduction method.

For both MNIST and Fashion-MNIST datasets, only one DAE architecture, the one with only fully connected layers, is used. When it comes to the CIFAR-10 case, two DAE architectures, one with only fully connected layers and the other with only convolutional layers are used. Table. 5.2 compares results obtained from two architectures on CIFAR-10. The fully connected based DAE shows stronger reconstruction ability by lower reconstruction loss and slightly higher ex-situ ONN test accuracy. However, this comes at the cost of $88.1\times$ more parameters used in the encoder and decoder. When comparing the selected DAE performance with the other two methods, it performs worse in MNIST and Fashion-MNIST even with ex-situ training, but reaches the same level of accuracy on the CIFAR-10 dataset. Moreover, there is a huge gap between the ex-situ ONN and the in-situ ones ($\geq 10\%$), indicating that the features extracted in the latent space of DAEs are hard to be captured by ONN.

Although models with PCA and FFT exhibit relatively high accuracy using the maximum allowable number of input features, their corresponding performance degrades significantly when using fewer features. As shown in Fig. 5.1, with PCA, models trained on MNIST and Fashion-MNIST have their accuracy drops below 60% when only 10 features are used (which also correspond to the number of classes in the dataset). The accuracy drop magnitude is smaller in the modified CIFAR-10 dataset as for binary classification, 10 input features are still sufficient for the model to make its decision. A similar trend can be observed in models trained with FFT, as shown in the analysis of Appendix B. This trend emphasizes the importance of input feature number selection for each method and the potential negative impact brought by the scalability issue of ONN to its accuracy.

5.1.2 Comparison of Dimensionality Reduction Methods on Face Detection

Next, we look at the effect of dimensionality reduction methods applied to the face detection task with IoU ≥ 0.5 dataset. Only the ex-situ ONNs are used to ensure the best

Table 5.3: Performance of Ex-Situ Face Detection ONNs on Different Dimensionality Reduction Methods

Method	Classifier	Activation	Accuracy [%]		F1	AUROC
			Validation	Test		
PCA	ONN	EO	58.45	57.37	0.581	0.607
		cReLU	**77.00**	**69.31**	**0.707**	**0.741**
FFT	ONN	EO	51.45	54.52	0.595	0.567
		cReLU	73.90	61.50	0.626	0.652
DAE	ONN	EO	75.78	66.68	0.666	0.723
		cReLU	73.22	67.17	0.694	0.719
DAE + multi-tasking	*MLP*	*ReLU*	*75.18*	*67.68*	*0.697*	*0.737*
	ONN	EO	72.63	67.93	0.693	**0.741**
		cReLU	72.62	68.23	0.697	0.736

ability of models trained and help distinguish whether the poor performance of face detector sources from the dimensionality reduction method or the model itself.

The classification performance of models with different dimensionality reduction methods is summarized in Table. 5.3. The Multilayer Perceptron (MLP) indicated in Italic is the auxiliary digital classifier for multitasking. Overall, the model trained with PCA and cReLU activation outperforms the rest in all three metrics. The difference between the best model and the DAE-based models is small, less than 2% in accuracy, 0.13 in F1 score, and 0.22 in AUROC. Moreover, the DAE-based model shows a more uniform performance between models with EO and cReLU activation. Since the EO activation is designed for in-situ training conditions, it results in low accuracy in ex-situ conditions. The training of the DAEs mitigates this problem by providing a better starting point for the classifiers by performing more complex transformations between the input image and the output feature vector. However, this comes at a cost of more than $25\times$ more CPU time spent on transformation than PCA and FFT. Therefore, we use PCA for the rest of our experiments.

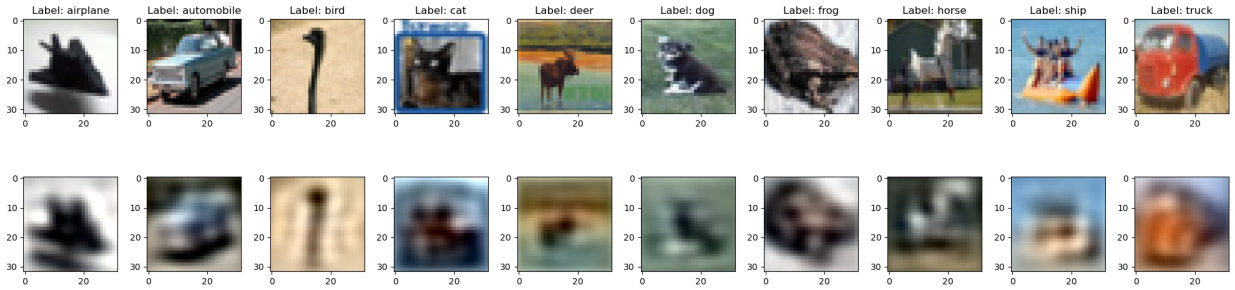


Figure 5.2: Original CIFAR-10 images (upper row) and their reconstruction (lower row) using PCA transformation

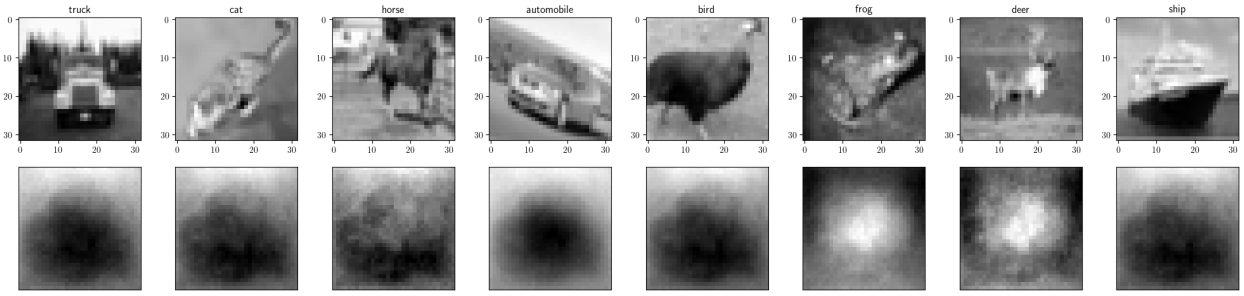


Figure 5.3: CIFAR-10 images (upper row) and their reconstruction (lower row) from the DAE with fully-connected layers (figures are converted to greyscale before training)

5.1.3 Comparison of Dimensionality Reduction Methods by Visualization

Another effective way of assessing the ability of dimensionality reduction methods is to inverse transform the extracted features and observe the similarity between the original and reconstructed images. This technique applies to both PCA and DAE. Here, we visualize the reconstructed CIFAR-10 images from PCA in Fig. 5.2, from DAE with fully connected layers in Fig. 5.3, and from DAE with convolutional layers in Fig. 5.4 with $N = 64$ features extracted during the transformation. More similar visualizations of MNIST and Fashion-MNIST can be found in Appendix B. From direct observation, PCA best keeps both the shape of the object depicted and the color channels present in CIFAR-10 images. Though DAE with convolutional layers can capture mostly the rigid shape of the objects, the coloring is largely missing from the reconstructions due to the limited number of pa-

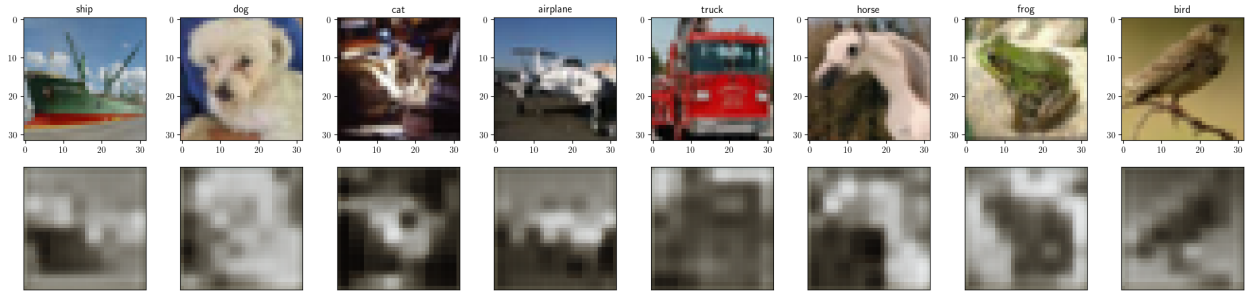


Figure 5.4: CIFAR-10 images (upper row) and their reconstruction (lower row) from the DAE with only convolutional layers

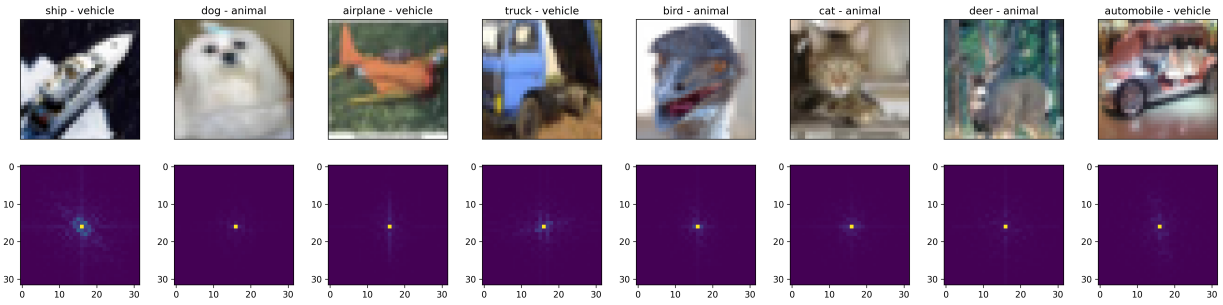


Figure 5.5: CIFAR-10 images before (upper row) and after (lower row) Fast Fourier Transform

rameters in the encoder-decode structure. The DAE with fully connected layers performs the worst among the three models. The figures are converted to greyscale prior to the training to save parameters and more importantly, the model can only perform a saliency mask level reconstruction of the object depicted in the figure.

The visualization of the FFT method is different, as we have a transformation followed by cropping. The direct inverse transformation will not provide any insight into how the two steps work collectively. Therefore, in Fig. 5.5, we visualize the Fourier coefficients obtained after the transformation of the CIFAR-10 images. For each image from a different class, there are some distinctive patterns that can be found in the frequency domain plot. Although some of them are hard to distinguish by humans, it is expected that NNs can effectively capture the nuances and perform correct classification.

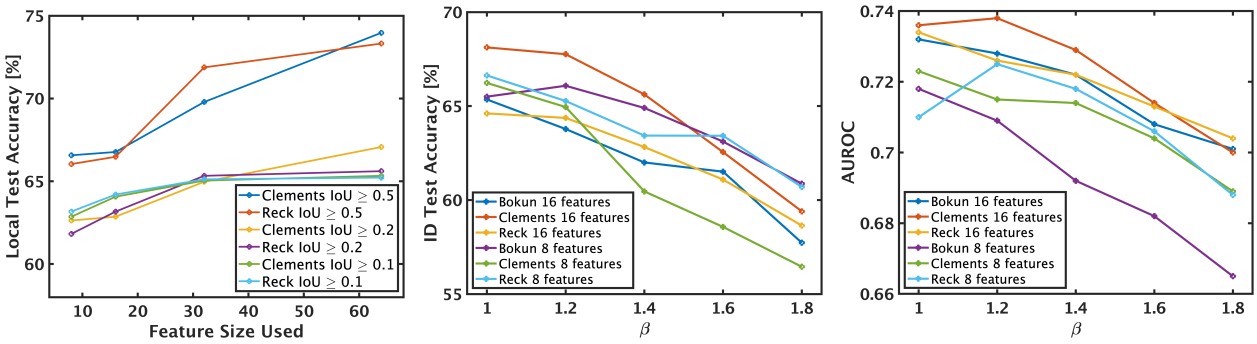
5.2 Hyperparameters Tuning and Selection for ONN

5.2.1 Topology, Neural Network Width, and Depth

The optical processor topology determines the size of the neural networks implemented. Similar to the issues we encountered with reference datasets, while Reck and Clements meshes can support as large as 64×64 neurons, the Bokun mesh experienced zero-gradient issues during backpropagation when its input size grows beyond 16. The Diamond mesh consistently experiences zero-gradient issues and it is therefore eliminated from the rest of the discussions.

The feature size (N) for ONN input/output refers to the width of the neural network implemented. As shown in Fig. 5.6a, the accuracy of each model on its local test set increases by up to 8% from 8 to 64 features as the neural network receives more information for decision-making. However, this increase is mostly due to the perfect operating condition we assumed during the simulation. Considering actual fabrication constraints, more MZIs in the optical processor leads to higher propagation loss and stronger noise [12]. Therefore, for our final evaluation, only $N = 8$ and $N = 16$ are considered.

The depth of the neural network corresponds to the number of optical processors connected in series. Since only fully connected layers are used, the models are prone to



(a) Feature size vs. Accuarcy **(b)** β vs. Accuracy **(c)** β vs. AUROC

Figure 5.6: Change in ONN Accuracy and AUROC as a result of the hyperparameter search.

overfitting if too many neurons are used. In our case, the 2-layer network demonstrates the best performance among all models trained. Single-layer networks suffer from underfitting due to the insufficient number of parameters, and 3-layer networks start to overfit within 10 epochs.

5.2.2 Effectiveness of False Negative Reduction

Both redefined dataset and weighted class methods successfully reduce the FN rate in all models based on their performance on the IoU ≥ 0.5 test set. However, as evidenced by the huge gap in test accuracy between models trained with IoU ≥ 0.5 dataset and the other two in Fig. 5.6a, the learning ability of the models is limited by the complexity of the dataset, which increases as IoU threshold drops. The weighted class methods result in better test accuracy and the AUROC score. Fig. 5.6b and Fig. 5.6c depict the variation of test accuracy and AUROC in each ONN as β increases. The best $\beta = 1.2$ and 1.4 where the models experienced less than $\pm 3\%$ fluctuation in ID test set accuracy. Notice that in Section 4.3.2, we indicated that this stage intends to eliminate the infeasible solutions. Therefore, the combinations of hyperparameters being eliminated are: 1) IoU ≥ 0.1 dataset with 8 features, and 2) IoU ≥ 0.5 datasets trained with $\beta \geq 1.6$.

5.2.3 OOD Test Set Performance

Moving on from the hyperparameter tuning, we tested the models on the OOD test dataset. Table. 5.4 and Table. 5.5 records the test accuracy, F1 score, and $p(\text{trigger}|N)$ of all selected models under the ONN-C assumptions. On average, models with $N = 16$ outperform the ones with $N = 8$ in terms of accuracy and F1 score. Considering $p(\text{face}) = 1\%$, $P(\text{trigger}|N = 16) = 0.388 \pm 0.09$ and $P(\text{trigger}|N = 8) = 0.360 \pm 0.14$. Moreover, in this scenario, the relationship between accuracy or F1 and $p(\text{trigger}|N)$ is not linear. A high face detection accuracy (e.g., the Clements model with IoU ≥ 0.1 set and 16 features) may denote a situation of high FNs and low FPs, which is indeed detrimental to the entire

Table 5.4: OOD Results with Modified IoU

Trained Dataset	Topology	N	ONN-C		
			Accuracy [%]	F1	p(**trigger**$\|N$)
IoU $\geq$ 0.2	Reck	16	71.23	0.734	0.331
	Bokun		**76.38**	**0.790**	0.332
	Clements		71.48	0.737	**0.336**
	Clements	8	69.85	0.711	0.221
	Reck		**73.70**	**0.750**	**0.294**
	Bokun		70.56	0.707	0.272
IoU $\geq$ 0.1	Reck	16	75.93	0.787	0.331
	Bokun		73.00	0.770	**0.390**
	Clements		**78.08**	**0.799**	0.297
IoU $\geq$ 0.5	Bokun	16	74.70	0.782	**0.417**
	Clements		**76.00**	0.790	0.380
	Reck		75.98	**0.791**	0.394
	Clements	8	66.83	0.694	0.270
	Reck		**70.80**	**0.733**	**0.350**
	Bokun		68.23	0.679	0.262

Table 5.5: OOD Results with Different β

Trained Dataset	Topology	β	N	ONN-C		
				Accuracy [%]	F1	p(**trigger**$\|N$)
IoU $\geq$ 0.5	Clements	1.2	16	**77.93**	**0.810**	**0.386**
	Clements	1.4		75.08	0.792	0.438
	Reck	1.2		75.43	0.790	0.412
	Reck	1.4		72.55	0.775	0.479
	Bokun	1.2		77.18	0.804	0.400
	Bokun	1.4		76.50	0.799	0.403
	Reck	1.2	8	71.33	0.718	0.278
	Reck	1.4		**75.28**	**0.768**	0.296
	Clements	1.2		69.93	0.701	0.276
	Clements	1.4		72.95	0.759	0.379
	Bokun	1.2		69.73	0.714	0.322
	Bokun	1.4		69.53	0.747	**0.499**

system accuracy. Notably, p(trigger$|N$) values with the weighted class method are larger than those of the other methods, indicating its more prominent effect on reducing FNs but with a sacrifice of increasing FPs which leads to more frequent activation of the second phase.

Recall that in Section 4.3.2, we defined the ONN-UA test set to contain two sets of labels, each leading to a set of macro-level and micro-level accuracy calculated for the system. We report both sets of accuracy for our model comparisons. As shown in Table. 5.6 and Table. 5.7, the macro-level accuracy of ONN is significantly higher than the micro-level accuracy, as the later inferences can mitigate the FNs made by the prior ones. The average number of n(trial) is 12.22 when $N = 16$ and 12.83 when $N = 8$. Models taking in more features are more sensitive to partial faces and use fewer trials to locate a face. Moreover, there is an almost linear relationship between macro-level accuracy and n(trial). A higher macro-level accuracy usually denotes a low n(trial), indicating that the ONN performs better at identifying partial faces and fewer searches of ONN on the test set images reach the maximum number of image patches and fail to find a face.

Table 5.6: OOD Results with Modified IoU

Trained Dataset	Topology	N	ONN-UA		
			Micro-level Accuracy [%]	Macro-level Accuracy [%]	n(trial)
IoU $\geq$ 0.2	Reck	16	74.70	**98.40**	12.38
	Bokun		75.15	98.15	12.37
	Clements		**76.10**	98.30	12.38
	Clements	8	77.43	97.45	12.57
	Reck		**79.38**	**97.50**	12.62
	Bokun		77.88	93.40	13.26
IoU $\geq$ 0.1	Reck	16	76.03	97.70	12.45
	Bokun		73.40	**98.90**	12.23
	Clements		**78.55**	97.50	12.42
IoU $\geq$ 0.5	Bokun	16	74.20	99.60	12.18
	Clements		75.78	**99.65**	12.16
	Reck		**75.83**	99.55	12.15
	Clements	8	73.93	98.45	12.59
	Reck		**75.93**	**98.80**	12.46
	Bokun		75.88	93.80	13.46

Table 5.7: OOD Results with Different β

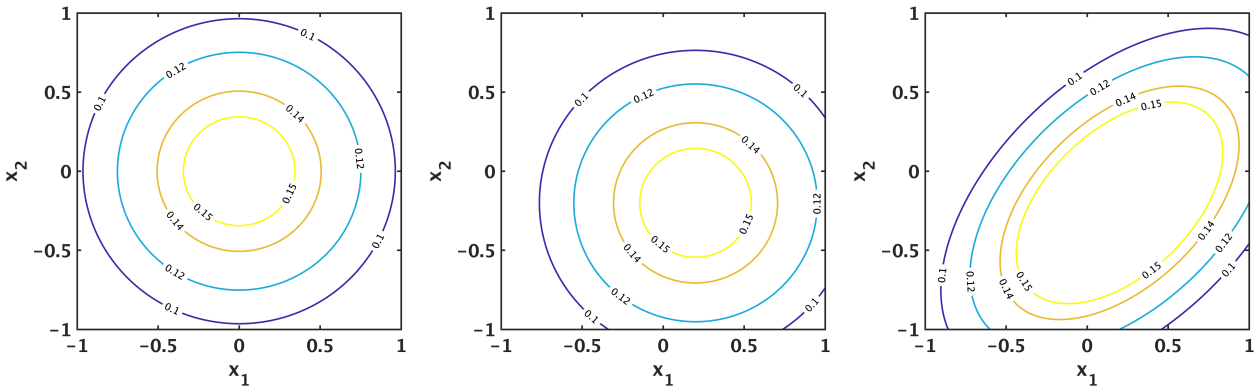
Trained Dataset	Topology	β	N	ONN-UA		
				Micro-level Accuracy [%]	Macro-level Accuracy [%]	n(trial)
IoU $\geq$ 0.5	Clements	1.2	16	**76.00**	**99.65**	**12.10**
	Clements	1.4		72.73	99.80	12.08
	Reck	1.2		75.08	99.60	12.15
	Reck	1.4		71.95	99.80	12.11
	Bokun	1.2		75.83	99.75	12.09
	Bokun	1.4		74.48	99.75	12.07
	Reck	1.2	8	77.93	94.70	13.23
	Reck	1.4		**77.98**	96.85	12.73
	Clements	1.2		77.40	94.00	13.32
	Clements	1.4		77.20	96.80	12.82
	Bokun	1.2		77.00	**99.00**	12.39
	Bokun	1.4		75.75	98.50	12.47

5.2.4 Sliding Window Pattern and n(trial)

In ONN-UA, we assumed the sliding window traversal from the top-left corner to the bottom-right corner. Since all the faces in the LFW dataset have been aligned in the middle of the image, it takes an average of 12.49 out of 25 inferences for all the models in Table. 5.6 and Table. 5.7 to find a face.

There are also many other traversal methods such as random selection, clockwise-spiral (CW-S), and counter-clockwise-spiral (CCW-S) search from the center, which can be effective under different $p(\text{face})$ assumptions. Indeed, if we consider the centered-face feature of the LFW dataset and start our search for a face in the center of an image and spread out spirally, it takes only about 2 inferences for the ONN to successfully locate a face.

However, in reality, the face of the user can potentially occur at any location in the range of a camera. To obtain a holistic view of which traversal method works the best, we modeled the occurrence of a face in a 250×250 px image range with a 2-dimensional Gaussian distribution $X \sim N(\mu_{\text{face}}, \Sigma_{\text{face}})$. $X = (x_1, x_2)$ and we sample $x_1, x_2 \in [-1, 1]$ with



(a) $\mu = [0\ 0], \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ **(b)** $\mu = [0.2\ -0.2], \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ **(c)** $\mu = [0.2\ -0.2], \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$

Figure 5.7: The probability density function of the 2D Gaussian distribution used for simulating the face positions in a frame, with different means and covariance matrices.

250 uniformly distributed points. As shown in Fig. 5.7, the mean vector $\mu_{\text{face}} = E[X] = [E[x_1], E[x_2]]^T$ controls the most likely position of face occurrence and the covariance matrix Σ_{face} controls the direction of spreading of the probability density function. Therefore, for a pixel location $X = (x_1, x_2)$,

$$p(\text{face}) = \frac{1}{2\pi}\det(\Sigma_{\text{face}})^{-\frac{1}{2}}\exp(-\frac{1}{2}(X - \mu_{\text{face}})^T\Sigma_{\text{face}}^{-1}(X - \mu_{\text{face}})) \tag{5.1}$$

To begin with, we assume a perfect face detection model that achieves 100% accuracy in distinguishing faces from partial or absent faces. For each traversal method, we then calculate the average of $p(\text{face})$ in the 100×100 px patches it cuts out. When the average probability reaches a threshold of 0.15, we assume a face is identified and the traversal stops. The threshold is determined by assuming a 2D standard Gaussian distribution (Fig. 5.7a) imitating the center-aligned LFW dataset and finding out the threshold required by the top-left to bottom-right (TLBR) traversal to reach the 12.49 out of 25 inferences for searching a face. We then calculate the average number of patches (considered as $n(\text{trial})$) across methods after repeating with 20,000 different combinations of μ_{face} and Σ_{face}. In the random selection method, we also test with 10 different random seeds for the selection and take the average of $n(\text{trial})$.

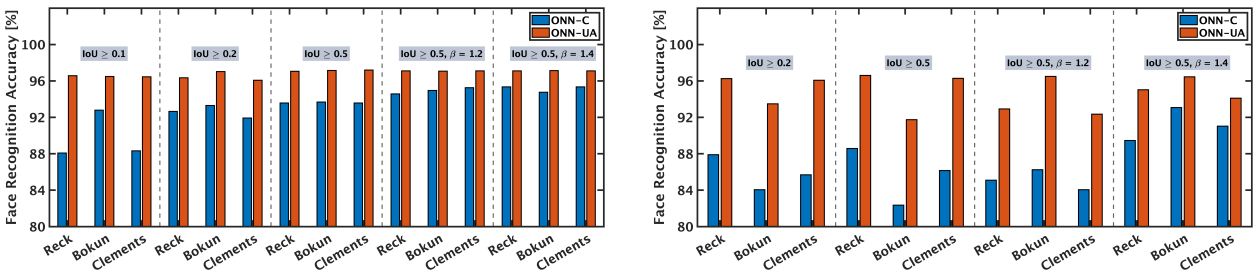
Table 5.8: Required $n(\text{trial})$ for Identifying A Face at Random Location

Method	TLBR	Random	CW-S	CCW-S
$n(\textbf{trial})$	12.50	12.99	**12.19**	12.36

Results from the simulation are shown in Table. 5.8. While the TLBR gradually converges to an average number of 12.5, the CW-S with a clockwise spiral cut from the center gives the fewest number of inferences. Therefore, for future considerations, using different traversal techniques can slightly reduce the power and energy consumption of the ONN face detection.

5.3 Complete System Evaluation

The faces marked "positive" by the ONN face detector are passed to the digital processor with DNN to perform face recognition. In the ONN-C scenario, using the 100×100 px bounding boxes detected by MTCNN, FaceNet itself can achieve an accuracy of 97.3% with a Euclidean distance threshold of 1.0. In the ONN-UA scenario, the 13th image patch in the middle of the image leads to a FaceNet accuracy of 98.4% with the same Euclidean distance threshold as ONN-C. Moving one cut to the left or right (the 12th and 14th image patch) will reduce the accuracy by around 2%. The rest of the image patches lead to a face recognition accuracy $\leq 60\%$.



(a) $N = 16$ **(b)** $N = 8$

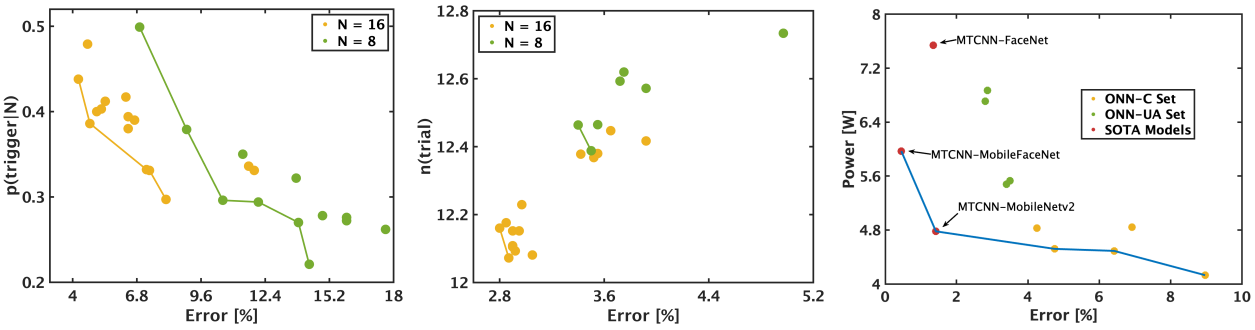
Figure 5.8: Full system (end-to-end face recognition) accuracy of models with different numbers of input features.

5.3.1 Entire System Accuracy

Combining the results from two phases, we evaluate the performance of the entire system design. Since the two use cases use the same final face recognition dataset, we can compare the performance between them directly.

Fig. 5.8 shows the entire system face recognition accuracy of all selected models. In ONN-C, the highly accurate FaceNet successfully boosted the accuracy of the entire system from an average of 74.5% and 72.0% in face detection with $N = 16$ and $N = 8$ to 93.2% and 87.0%. However, the performance gap between the two feature sizes is still huge and very few models with $N = 8$ (2 out of 12) can reach more than 90% face recognition accuracy.

In ONN-UA, the entire system accuracy is indeed lower than that of the ONN face detector, as more faults are made in the second phase. Those faces detected earlier than the 12th cut or later than the 14th cut can hardly be recognized since they contain only partial face features. Despite the drop in accuracy, on average, ONN-UA outperforms ONN-C by 3.9% with $N = 16$ and 7.84% with $N = 8$. The performance gap between the numbers of input features is also reduced to 2% on average.



(a) ONN-C **(b)** ONN-UA **(c)** ONN vs. SOTA Design

Figure 5.9: The projection of the trained models to the design space.

Table 5.9: Performance of Pareto Optimal Set in ONN-UA

Scenario	Trained Dataset	Topology	β	N	ID	Phase 1 Performance		Total Performance		p(trigger$\|N$)/ n(trial)	Consumption		
						Accuracy [%]	F1	Accuracy [%]	F1		Power [W]	Latency [ms]	Energy [mJ]
ONN-UA	IoU ≥ 0.5	Clements	1	16	C1-16U	65.82	0.738	**97.20**	**0.971**	0.972/12.16	6.71	8.81	63.82
		Bokun	1.4		B4-16U	**74.48**	**0.895**	97.13	**0.971**	0.971/12.07	6.87	8.80	63.80
		Reck	1	8	R1-8U	75.93	0.944	**96.60**	0.965	0.966/12.46	5.48	8.76	63.38
		Bokun	1.2		B2-8U	**77.00**	**0.944**	96.50	0.964	0.965/12.39	5.53	8.75	63.33
DNN	N/A							98.42	0.984	N/A	7.54	20.71	117.87

Table 5.10: Performance of Pareto Optimal Set in ONN-C

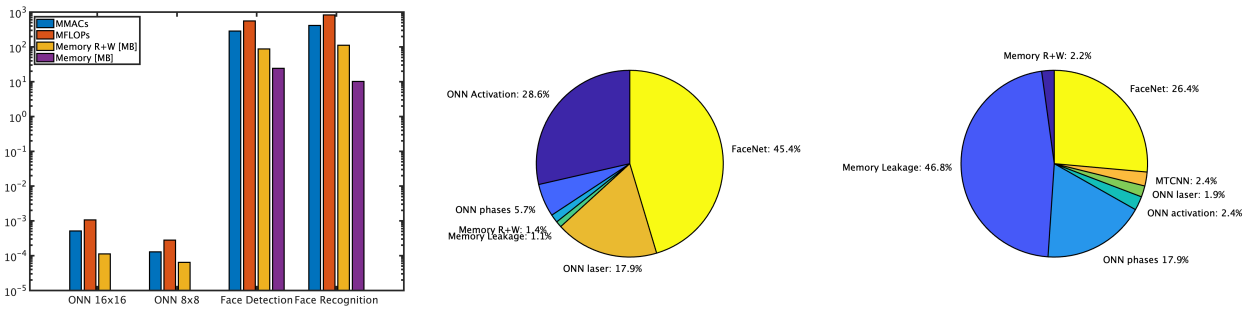
Scenario	Trained Dataset	Topology	β	N	ID	Phase 1 Performance		Total Performance		p(trigger$\|N$)/ n(trial)	Consumption		
						Accuracy [%]	F1	Accuracy [%]	F1		Power [W]	Latency [ms]	Energy [mJ]
ONN-C	IoU ≥ 0.5	Clements	1	16	C1-16C	76.00	0.790	93.58	0.932	0.341/1	4.49	7.22	40.37
		Clements	1.2		C2-16C	**77.93**	**0.810**	95.25	0.951	0.346/1	4.52	7.31	40.87
		Clements	1.4		C4-16C	75.08	0.792	**95.75**	**0.956**	0.401/1	4.83	8.46	47.43
		Clements	1.4	8	C4-8C	**72.95**	**0.759**	91.03	0.902	0.344/1	4.13	7.27	40.57
		Bokun	1.4		B4-8C	69.53	0.747	**93.08**	**0.870**	0.465/1	4.84	9.78	54.84
DNN	N/A							98.42	0.984	N/A	7.54	20.71	117.87

5.3.2 The Pareto Optimal Set of Designs

A highly accurate model may come at the cost of higher power and energy consumption. Therefore, to find out the most balanced design between the two objectives, we project the trained models to the two-dimensional design spaces defined in Section. 3.8. Fig. 5.9a and Fig. 5.9b depicts the results of the search. The Pareto-optimal solutions are connected with lines for each value of N and their corresponding hyperparameters and performance are recorded in Table. 5.9 and Table. 5.10 by filtering out the models with accuracy below 93% and 90% for $N = 16$ and $N = 8$. In all cases, high accuracy in Phase 1 never implies a high overall accuracy but smaller $p(\text{trigger}|N)$ and $n(\text{trial})$. ONN-UA methods show higher accuracy as later inferences can continuously correct mistakes made in earlier ones.

5.4 Power, Energy, and Latency Estimation

Estimated power, energy, and latency estimations are reported in the last three columns of Table. 5.9 and Table. 5.10. The DNN case in both tables assumes MTCNN and FaceNet run continuously. In the budgeting analysis of ONN power consumption, we account for a 10 dBm C-band laser with wall plug efficiency of 10% [66]. This optical power is



(a) Operations by Stage **(b)** C1-16U **(c)** C4-8C

Figure 5.10: The breakdown of arithmetic operations, memory operations, and power consumption by stage (ONN phases refer to the programming power of phase shifters in ONN).

sufficient to split and feed all input ports of a 16×16 and a 8×8 ONN. Moreover, for the programming of the phase shifters, we assume a uniform power consumption and the heater efficiency is 1.42 mW/π [67].

The ONN-UA models, assuming more frequent FaceNet wakeup ($\geq$96%), exhibit higher power and energy consumption than ONN-C. Within the Pareto optimal solution set, ONN-UA requires approximately 12.2 image patches to identify a valid face during sliding window traversal from top-left to bottom-right. However, even the most inefficient model saves 10.9% of power and 46% of energy compared to the DNN model. Optical processors operate at GHz rates and consume less power, resulting in the ONN's serial subsampling taking less time and energy than its electronic counterpart. The most efficient ONN-C model achieves a significant twofold reduction in power and energy, accompanied by a 7% accuracy drop. Due to the low frequency of face appearance, consistently powering down the demanding DNN face recognition leads to substantial savings in power and energy.

To better compare the effectiveness of the proposed models, we estimated the power consumption of several SOTA face recognition models [37] designed for edge devices with the same device and compared them in Fig. 5.9c. The blue line connects the Pareto optimal solutions for all designs. Notably, the original MTCNN-FaceNet is not Pareto-

optimal. However, when incorporated into the ONN always-on system we propose, several different designs are selected as the Pareto-front.

We further break down the power consumption of the entire system into different phases. As shown in Fig. 5.10a, the DNNs, including MTCNN for face detection and FaceNet for face recognition, takes up the majority of floating point operations and memory accesses. Therefore, in the most power-saving ONN-C case, as shown in Fig. 5.10c, the memory leakage power and FaceNet computations take up more than 72% of the power consumption. However, when it comes to the more accurate ONN-UA case, shown in Fig. 5.10b, the elimination of MTCNN significantly relaxes the requirement for memory. The repetitive inferences of ONN, leading to repetitive laser input and EO activations, account for more than 45% of the power consumption, which is almost at the same level as FaceNet.

Chapter 6

Results and Discussion with Lossy Environment

With the presence of device imperfections discussed in Section 2.2.3, the output port field intensity of ONN can be largely altered and the decisions made by it will change subsequently. Therefore, we perform sensitivity analysis on the selected models from the perfect operating stage and report their performance against noise.

6.1 A Statistical Approach of Estimating Effect of Noise

Before delving into details of how the proposed system accuracy fluctuates in terms of FN and FP increase or decrease, we first quantify the significance of FN and FP at different phases of the proposed system. This will help us understand why the increase or decrease in faulty cases may or may not decrease or increase the accuracy values at different phases.

We begin by optimistically assuming that the prediction made by the second-phase processor is uniform across all data instances. This means that all data instances have the same level of "difficulty" to the classifiers and they all belong to the "easy-to-learn" region in dataset cartography [68]. The model makes consistent and correct predictions

on the same data instance across time. Therefore, we can safely interpret the accuracy as a probability of whether a data instance can be correctly classified. Although this may deviate from the real-life case where certain data instances are harder to distinguish, it provides insight into how the noise will affect the accuracy of ONN and the system and how they are connected with the variations in FN and FP.

Suppose that under perfect operating conditions, the ONN model prediction contains FN_0 FN cases and FP_0 FP cases. This means that we also have $TN_0 = 2000 - FP_0$ TN cases and $TP_0 = 2000 - FN_0$ TP cases, considering our OOD test set contains 4,000 images that are equally distributed over the positive and negative classes.

The accuracy of ONN can be expressed as

$$\text{ONN\_Acc}_0 = \frac{TN_0 + TP_0}{TN_0 + FP_0 + FN_0 + TP_0} = 1 - \frac{FN_0 + FP_0}{4000}. \tag{6.1}$$

The system accuracy can then be expressed as

$$\text{System\_Acc}_0 = \frac{TN_0 + TP_0 \cdot p + FP_0 \cdot p}{4000} = \frac{2000(1+p) - p \cdot FN_0 + (p-1)FP_0}{4000}, \tag{6.2}$$

where p is the standalone accuracy of (MTCNN +) FaceNet, interpreted as the probability of an image being correctly classified.

Suppose that when an imperfect condition is applied, it leads to ΔFN cases of variation in FN and ΔFP cases of variation in FP. The ONN model now creates $FN_1 = FN_0 + \Delta FN$ FN cases and $FP_1 = FP_0 + \Delta FP$ FP cases.

The ONN accuracy now becomes

$$\text{ONN\_Acc}_1 = 1 - \frac{FN_0 + FP_0 + \Delta FN + \Delta FP}{4000} = \text{ONN\_Acc}_0 - \frac{\Delta FN + \Delta FP}{4000} \tag{6.3}$$

Consequently, the system's accuracy becomes

$$\begin{aligned}\text{System\_Acc}_1 &= \frac{2000(1+p) - p \cdot (FN_0 + \Delta FN) + (p-1)(FP_0 + \Delta FP)}{4000} \\ &= \text{System\_Acc}_0 - \frac{p \cdot \Delta FN + (1-p)\Delta FP}{4000}.\end{aligned} \tag{6.4}$$

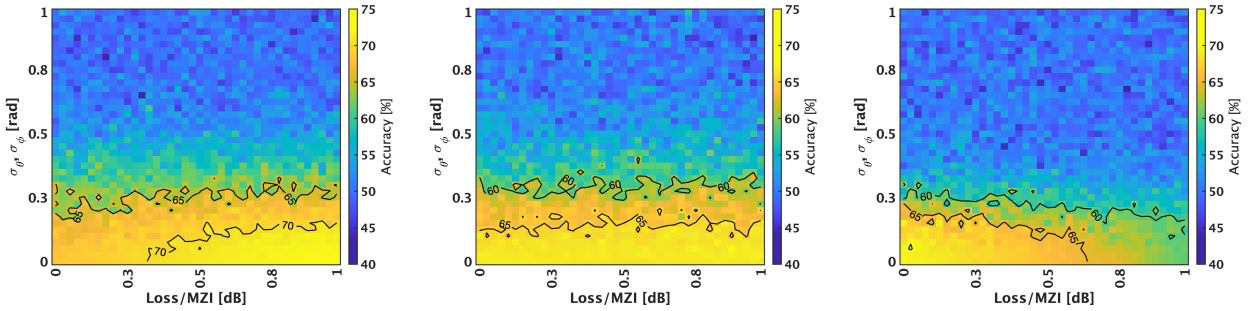
The ONN accuracy drops as ΔFN and ΔFP increases. To reverse this trend, at least one of the variations needs to be negative, with its magnitude larger than the other. However, for the system accuracy, the case becomes more complex. Suppose that we have $p = 0.973$ which corresponds to the accuracy of our current phase 2 system,

$$\begin{aligned}\text{System\_Acc}_1 &= \text{System\_Acc}_0 - \frac{0.973 \cdot \Delta FN + (1-0.973)\Delta FP}{4000} \\ &= \text{System\_Acc}_0 - \frac{0.973 \cdot \Delta FN + 0.007\Delta FP}{4000}.\end{aligned} \tag{6.5}$$

When $\Delta FN > 0$ and $\Delta FP \geq 0$, the accuracy drops, and conversely, when $\Delta FN \leq 0$ and $\Delta FP < 0$, the accuracy increases. In case of $\Delta FN \geq 0$ and $\Delta FP > 0$, $\frac{|\Delta FP|}{|\Delta FN|} > \frac{p}{1-p}$ is required for the accuracy to grow. Similarly, when $\Delta FN < 0$ and $\Delta FP \geq 0$, $\frac{|\Delta FP|}{|\Delta FN|} < \frac{p}{1-p}$ is required for the accuracy to grow. When $p = 0.973$, $\frac{p}{1-p} = 139$, the system requires 139 fewer FP cases to compensate for 1 additional FN case to maintain the accuracy. The function $\frac{p}{1-p}$ has an asymptote at $p = 1$. Therefore, as $p \to 1$, the result of the fraction further grows to infinity. This observation indicates that, while the ONN is equally sensitive to the FN and FP cases, the system accuracy is more susceptible to variations in FN cases, especially when the accuracy of the digital face recognition system is high. This observation also emphasizes the need for methods to reduce FN cases.

6.2 System Operation with Noise

Among the hyperparameters we considered for this work (Table. 4.1), the topology and the number of features are the factors that make a difference in the robustness of ONN against imperfect operations. In this section, we consider the three effective topologies



(a) Bokun **(b)** Clements **(c)** Reck

Figure 6.1: Change in ONN Accuracy of different topologies with 8 features after loss and phase angle deviations applied.

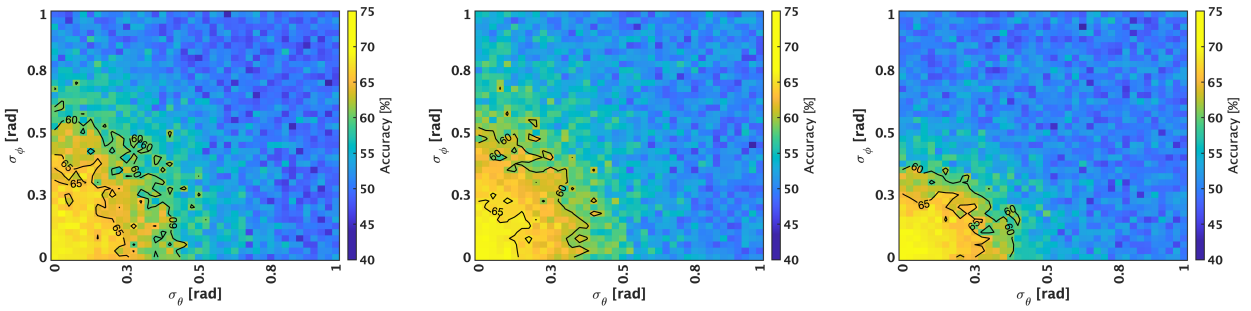
(a) FN **(b)** FP

Figure 6.2: Change in the number of FN and FP of ONN with Bokun topology due to imperfect operation

found in our grid search and test the impact of imperfect operation for models with different topologies trained on the same IoU ≥ 0.5 set without FN reduction under the ONN-C assumption with $N = 8$ and $N = 16$.

6.2.1 Effect of Imperfect Operations on Face Detection

Recall that the face detection stage is fully implemented in the optical domain, Fig. 6.1 indicates the variation in 8×8 ONN accuracy concerning the levels of noise in the form of light propagation loss per MZI and the deviation in phases programmed to phase shifters.



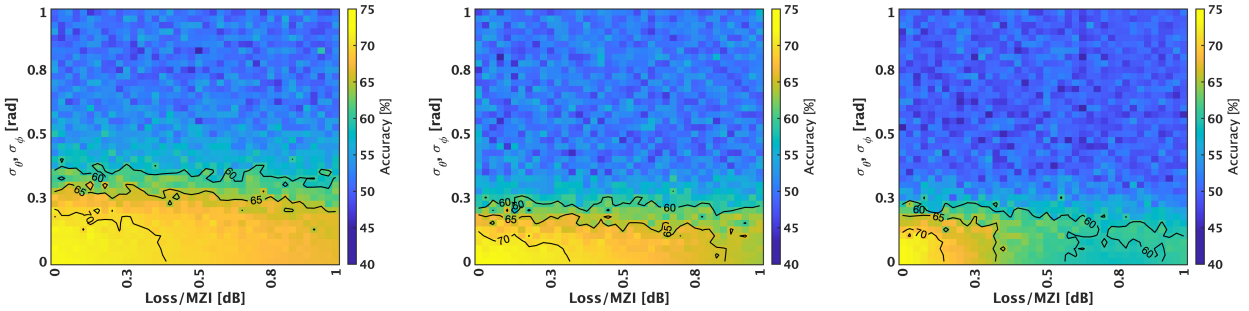
(a) Bokun **(b)** Clements **(c)** Reck

Figure 6.3: Change in ONN Accuracy of different topologies with 8 features after different phase angle deviations on the internal/external arms applied.

In Fig. 6.1, the programming error has a more profound impact on the ONN accuracy, as seen by the faster conversion of the grid coloring from yellow to blue along the y-axis, indicating a more rapid and larger magnitude of decrease in accuracy. Moreover, in Fig. 6.1a and Fig. 6.1b, the Bokun and Clements topology are more robust against the increase in signal attenuation and phase shifter programming error. Their contours of 65% accuracy are $2.54\times$ and $1.50\times$ larger than that of Reck.

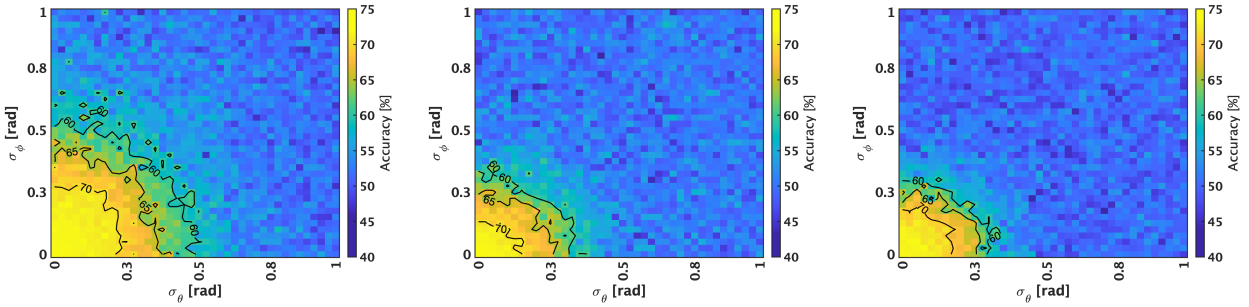
The impact of propagation loss on the ONN accuracy can be bidirectional, depending on the topology. The Reck mesh experienced relatively 13.6% decrease in ONN accuracy when propagation loss increases from 0 to 1 dB per MZI with no phase programming error. Under the same condition, the Clements topology only experienced an absolute range of 0.4% fluctuation in accuracy, and the Bokun mesh even displayed an ascending trend in accuracy. As we take a closer look at the FN and FP cases in it (Fig. 6.2), there is an underlying increasing trend in FN cases while a decreasing trend in FP cases. When the magnitude of the FP decrement outweighs the FN increment, we observe an increase in ONN accuracy.

In Fig. 6.3, the deviation in internal/external arm phase shifter programming has an almost equal impact on the performance of ONN, as seen by the contours indicating same-accuracy levels in a quarter-circle shape. All three topologies tested have shown similar patterns in the performance degradation as noise level increases, with Bokun



(a) Bokun **(b)** Clements **(c)** Reck

Figure 6.4: Change in ONN Accuracy of different topologies with 16 features after loss and phase angle deviations applied.



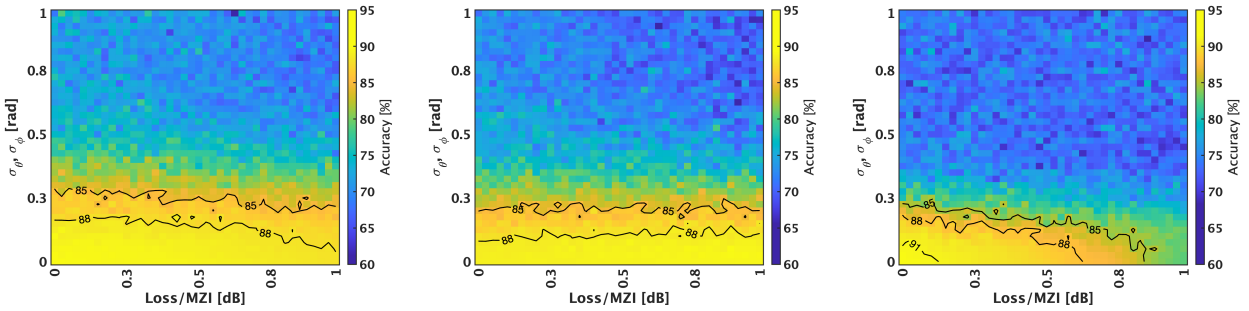
(a) Bokun **(b)** Clements **(c)** Reck

Figure 6.5: Change in ONN Accuracy of different topologies with 16 features after different phase angle deviations on the internal/external arms applied.

mesh outperforming the rest two with a large area enclosed by the 65% accuracy contour.

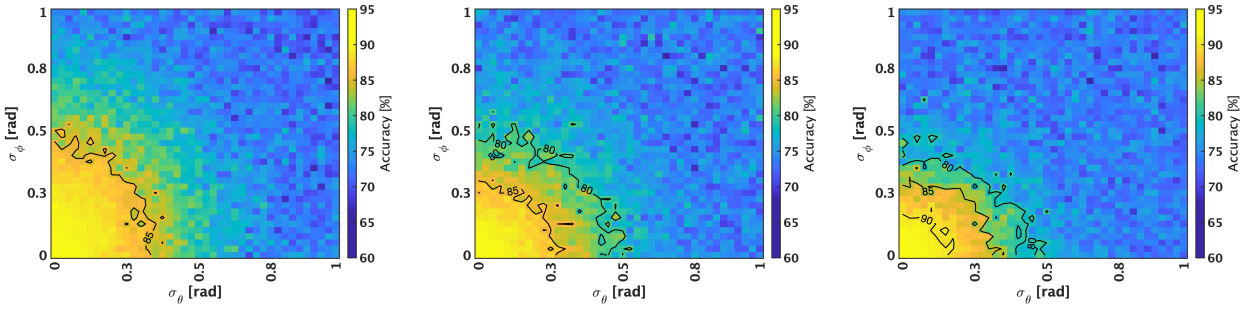
Similarly, Fig. 6.4 and Fig. 6.5 indicate the variation in 16×16 ONN accuracy with different levels of imperfection conditions. The impact of propagation loss per MZI is still less significant than that of the phase programming deviations, as seen by the quarter-elliptical shape of contours in Fig. 6.4, with the major axis lying on loss/MZI. However, as a result of the larger size of meshes, more MZIs are placed on the critical path where light couples through and this leads to a stronger effect of signal attenuation [12]. The accuracy of ONNs no more has a chance to increase along the x-axis. When we decompose the error cases of ONN, there is indeed a slight decreasing trend in FN accompanied by a more

significant increasing trend in FP. Since FP and FN cases are placed with equal weight in contributing to ONN accuracy, as calculated in Section. 6.1, the ONN accuracy still drops when the imperfection magnitude increases. The internal and external arm phase shifter programming deviation still shows a similar impact on the ONN performance, as depicted in Fig. 6.5. Bokun mesh outperforms the rest two with $2.02\times$ and $3.30\times$ larger area enclosed by the 70% accuracy contour.



(a) Bokun **(b)** Clements **(c)** Reck

Figure 6.6: Change in System Accuracy of different topologies with 8 features after loss and phase angle deviations applied.



(a) Bokun **(b)** Clements **(c)** Reck

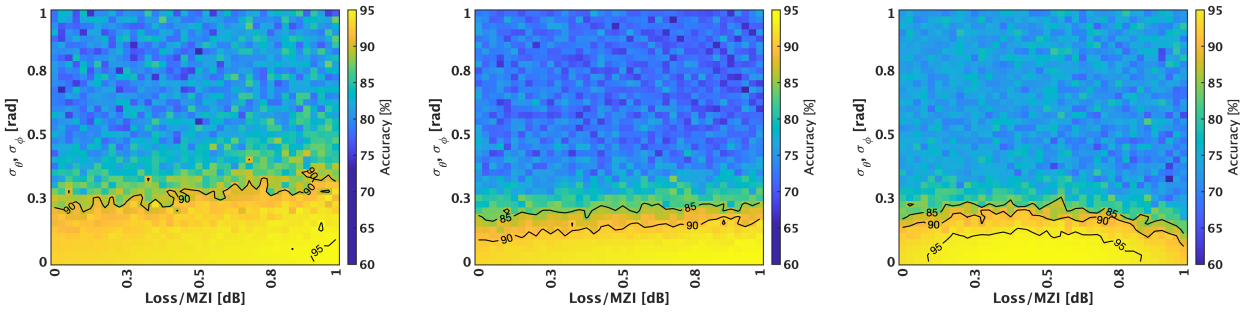
Figure 6.7: Change in System Accuracy of different topologies with 8 features after different phase angle deviations on the internal/external arms applied.

6.2.2 Effect of Imperfect Operations on Face Recognition

The full system accuracy can be changed as a result of the non-idealities in ONN. Whether this impact on accuracy is positive or negative depends on either more FN or FP cases in face detection are generated.

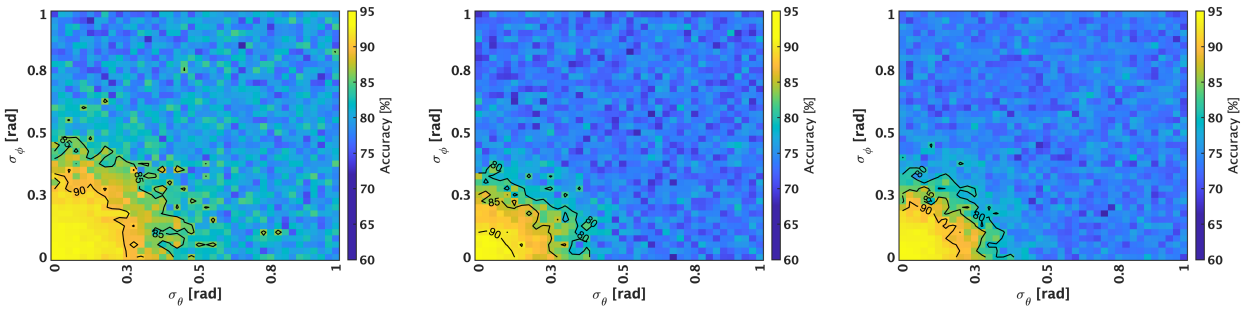
As shown in Fig. 6.6 and Fig. 6.7, with 8 input features, models built on all three topologies have shown a decreasing trend in accuracy when the intensity of noise increases. The Reck topology, although starting from the highest accuracy with perfect conditions, experiences a more significant drop in accuracy when the propagation loss per MZI increases. Without phase programming error, its accuracy drops below 85% after the propagation loss reaches 0.88 dB per MZI, while the other two topologies maintain more than 88% accuracy within this range. Furthermore, despite the fact that the Bokun topology exhibits an increasing trend in ONN accuracy with the increase of propagation loss when the phase programming deviation is low, its system accuracy still drops as the increase in FN in ONN has a more weighty impact on the entire system accuracy, as quantified in Section 6.1.

There are more diverse trends in the accuracy variation when the number of input features increases to 16, as shown in Fig. 6.8 and Fig. 6.9. Though the phase shifter programming errors still significantly degrades the system accuracy, the highest system accuracy



(a) Bokun **(b)** Clements **(c)** Reck

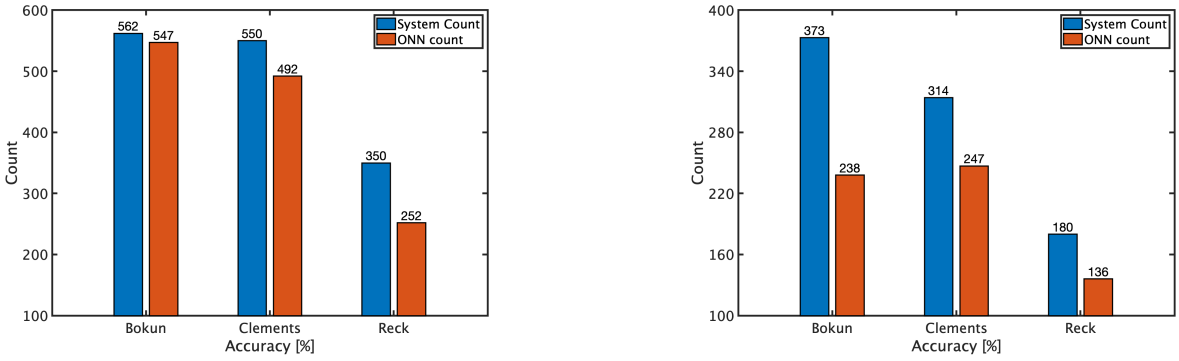
Figure 6.8: Change in System Accuracy of different topologies with 16 features after loss and phase angle deviations applied.



(a) Bokun **(b)** Clements **(c)** Reck

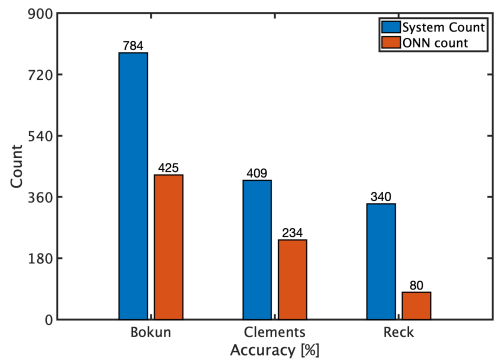
Figure 6.9: Change in System Accuracy of different topologies with 16 features after different phase angle deviations on the internal/external arms applied.

no more appears in the bottom left corner which corresponds to the perfect operating condition when propagation loss is considered. According to Section 6.1, one more FN case can mitigate the impact of $139\times$ more FP cases in the entire system evaluation. Therefore, although the magnitude of decrease in FN is less than the magnitude of FP increment in ONN, the entire system accuracy indeed increases. For models built on Reck topology, the FN case number decreases at first but starts to increase after Loss/MZI reaches 0.5 dB. Therefore, in Fig. 6.8, the total system accuracy decreases after this point, leading to the 95% accuracy contours lying in the middle of the x-axis.

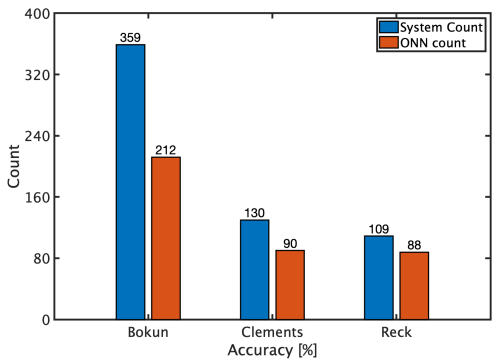


(a) Loss and Phase Angle deviations **(b)** Phase Angle deviations

Figure 6.10: Number of data points representing ONN/System Accuracy within 90% of the perfect condition accuracy with 8 features.



(a) Loss and Phase Angle deviations



(b) Phase Angle deviations

Figure 6.11: Number of data points representing ONN/System Accuracy within 90% of the perfect condition accuracy with 16 features.

6.2.3 Comparison of Topology Robustness Against Imperfect Operation

To quantify the performance of each topology on imperfect operating conditions, we define the FoM as the number of data points (or area) representing an ONN/system accuracy above 90% of the perfect condition accuracy. Compared to a fixed accuracy threshold, this FoM takes into account the different starting accuracy of the noise injection and, therefore, ensures the fairness of comparison.

Fig. 6.10 and Fig. 6.11 shows the results obtained by applying FoM to all models investigated in the Section 6.2.1 and Section 6.2.2. Regardless of the input feature size, there is a clear pattern that the Bokun mesh outperforms the other two. The FoM counts of Clements topology remain close to the Bokun topology when $N = 8$, but the performance gap is enlarged to 47.8% fewer counts in system accuracy when N increases, considering both propagation loss and phase shifter programming deviations. The Reck topology behaves even worse, as the gap between its performance and the Bokun topology grows from at least $1.6\times$ to at most $5.3\times$ fewer FoM counts as N increases. Indeed, this performance difference can be attributed to the underlying mesh designs. In the designs of Bokun and Clements mesh, they minimized and balanced the optical path lengths to

enhance the topology's robustness against optical loss. The Reck mesh, however, has an asymmetrical shape and suffers from imbalanced optical loss between two output ports.

6.3 Best Model Performance with Noise Applied

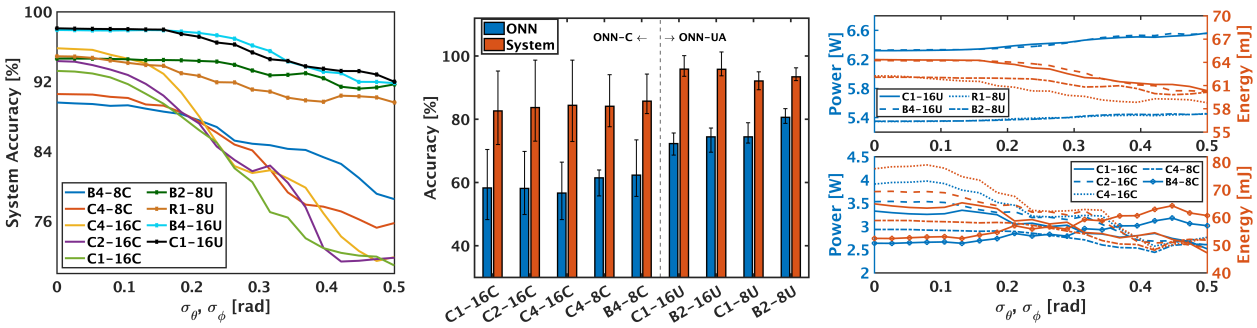
In this section, we subject the selected optimal models to imperfect operating conditions calibrated by real fabrication results.

6.3.1 Model Performance in Classification

Fig. 6.12a shows the range of system test accuracy obtained with $\sigma_\theta, \sigma_\phi \in [0, 0.5]$. Since we fixed the value of Loss/MZI, there is no more increasing trend in the system accuracy of all models. As σ_θ and σ_ϕ grow, programmed phase shifts are more likely to more significantly deviate from the trained value. Consequently, the accuracy of all models decreases. The least resilient model, C1-16C, experienced more than 23% drop in accuracy value as σ_θ and σ_ϕ grows to 0.5. Comparing across models, 8×8 meshes are more robust to noise than 16×16 ones, as seen by the smaller range of best-worst case accuracy variation in Fig. 6.12b. Fewer MZI units lead to less accumulation of error [12]. Similarly, the ONN-UA model demonstrates stronger resilience to imperfection, as the repetitive inference steps allow for iterative error correction.

6.3.2 Model Power and Energy Consumption

The propagation loss and phase errors also influence $p(\text{trigger}|N)$ and $n(\text{trial})$, leading to changes in power and energy use. Fig. 6.12c depicts the variation in power and energy due to phase errors. Most models experience a decrease in both as more FNs occur. The only exceptions are the power of ONN-UA models and Bokun $N = 8$ model with $\beta = 1.4$. In the earlier cases, the increase in $n(\text{trial})$ outweighs the savings from more FNs, and in



(a) Accuracy vs. $(\sigma_\theta, \sigma_\phi)$ **(b)** Accuracy Distribution **(c)** Power/Energy vs. $(\sigma_\theta, \sigma_\phi)$

Figure 6.12: Change in system accuracy, power, and energy consumption due to imperfect operation conditions.

the latter case, the increase in FP precedes FN so Processor 2 is indeed triggered more frequently.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this work, we aim to tackle the challenges aroused between the tight power budget and high accuracy requirement for real-time face recognition from both algorithmic and hardware architectural sides. The conventional face recognition pipeline is split into two phases, one with an always-on face detector and the other being an event-driven face recognition network. To further reduce the power consumption and inference latency of the face detector, we introduced the energy-efficient ONNs which is an emerging technology for computing matrix-vector multiplication in the optical domain. This multi-stage electro-optic hybrid face recognition system with an always-on face detector achieves the goal of reducing power, latency, and energy consumption by running energy-efficient ONNs continuously during face detection and only waking power-hungry high-accuracy DNNs when faces are present.

To verify the correctness and efficiency of the proposed design, we explored two real-life scenarios of the system deployment, one requiring a subsampling process for locating faces and one assuming center-aligned faces. In a lossless environment, the most accurate model arises from the first scenario, reaching 97.2% accuracy in the LFW dataset with two-layer 16×16 optical processors for face detection and pretrained FaceNet for face

recognition. Compared to the same DNN implementation in electronics, it achieves an 11% reduction in power. The most power- and energy-efficient model arises from the second scenario, with 45.2% and 65.6% reduction in power and energy but a worst-case 7% drop in accuracy.

However, real-world optical device operation encompasses various aspects of imperfection. The light coupling in the waveguide attenuates due to the optical insertion loss and there is a constant imprecise mapping of trained phases to actual phase shifters due to static factors such as fabrication variation and dynamic factors such as thermal crosstalk or other sources of noise. Therefore, we took into account the imperfections in a sensitivity analysis and discovered that there is a constant tradeoff in the FN and FP cases of the ONN face detector. The tradeoff not only affects the accuracy values of different phases of face recognition but also alters the entire system's power and energy consumption. Fixing the propagation loss to a realistic value of 0.6 dB per MZI, the most accurate model identified with perfect operating assumptions experienced a worst-case 6.5% absolute system-level accuracy drop as the phase programming error increased. Accompanied by the drop in accuracy, we observe more FN cases made by ONNs and this eventually brings down the power and energy consumption of the system as the power-consuming digital DNN is activated less frequently.

7.2 Future Work

7.2.1 False Negative Reduction

In this work, we have already considered two methods for false negative reduction, the modified IoU threshold and the weighted class methods. Although two methods have demonstrated the ability to reduce FN cases, they are only used during the training process of ONN and therefore consider only lossless operating environment. For future work, FN reduction methods should also take care of the lossy environment during the actual operation of ONN. For instance, for asymmetric topologies such as Reck, placing

the output port for indicating label "0" or no face present at those experiencing more propagation loss to ensure a higher chance of its reading lower than that of label "1". Moreover, noise-aware training with existing FN reduction methods is another avenue to be considered.

7.2.2 Noise Resilient Model

Current designs of ONN are susceptible to imperfect operating conditions, especially when the neural network sizes increase. Therefore, developing noise-resilient models is crucial for ensuring optimal system performance in real life. Noise-aware training is one of the options for such development. Indeed, prior work in [48] has already attempted noise-aware training in fully connected layers implemented on ONN as part of the electronic CNN. Modeling the device imperfection by injecting Gaussian white noise, they were able to achieve $\geq 99\%$ accuracy in MNIST classification. For higher fidelity noise-aware training, we need to devise more precise modeling of the noise or imperfection faced by ONN. Apart from noise-aware training, other conventional techniques such as quantization and regularization on the ONN weights shall also be considered. The quantization process allows ONN to adapt to the limited voltage source resolution and the regularization technique can be deployed to penalize more on weight matrices which leads to a susceptible ONN to the imperfect operating conditions [69].

7.2.3 Hardware Implementation and Hardware Comparisons

The current work serves as a proof-of-concept of using the electro-optic hybrid system for face detection and face recognition. The experimental results are obtained from simulations calibrated with empirically obtained data. In the future, potential hardware implementation of the system should be considered. This includes:

1. A direct deployment of the proposed system with fabricated optical meshes and the proposed mobile GPU (Jetson Nano). This will include additional design in the in-

terconnect and electronic control circuits for the synchronization and orchestration of the components. Subsequently, we can perform more precise power, latency, and energy consumption analyses based on the actual measurements.

2. Implementation of the equivalent software on other hardware platforms. The same neural network architecture as the optical face detection can be performed on existing digital platforms, such as CPUs and FPGAs, and emerging technologies for more thorough performance analysis. The event-triggered DNN face recognition model can also be implemented on a wider spectrum of hardware platform selection, including more powerful GPU systems.

7.2.4 Resolve Fairness Concerns in Face Recognition Systems

Algorithmic fairness concerns have existed for a long time in computer vision machine learning algorithms that directly interface with humans [70]. The algorithms can discriminate against a certain person or a certain group of people based on their age, gender, and demographic features. Algorithmic bias can be sourced from the datasets on which the neural networks are trained. In this work, the ONNs are trained on the WIDER face dataset, which has been proven to entail unbalanced distribution over genders, age, and skin tone groups [71]. Additionally, the pre-trained FaceNet we used was trained on the VGGFace2 dataset, which is yet another unbalanced dataset among age and race groups [72]. Whether the aggregation of such two models in our design will deteriorate the algorithmic fairness remains a question.

To address this issue, future work should consider either selecting a fair dataset such as [73] or looking for ways to augment the existing dataset in use. Other algorithmic fairness mitigation strategies should also be considered and incorporated into the ONN training pipeline.

Appendix A

Accuracy Variation in FFT

As shown in Table. A.1, the accuracy of ONN model decreases as a result of the reduced Half-feature Length (L) in the FFT method.

Table A.1: Accuracy Variation with the Half-feature Length (L) in FFT

Dataset (Image Shape)	Trainable Parameter	Topology	Activation	L = 4		L = 3		L = 2	
				Validation Accuracy [%]	Test Accuracy [%]	Validation Accuracy [%]	Test Accuracy [%]	Validation Accuracy [%]	Test Accuracy [%]
MNIST (28*28)	Weight	-	EO	97.27	97.31	96.44	96.42	91.67	91.68
	Phase Shifter Angles	Clements	EO	87.85	91.07	90.03	90.25	75.42	74.48
			cReLU	89.68	90.17	89.84	90.22	78.84	78.20
		Reck	EO	90.37	91.92	90.71	91.00	79.23	78.62
			cReLU	88.97	91.12	90.02	90.08	80.99	80.13
		Bokun	EO	N/A	N/A	N/A	N/A	73.38	74.44
			cReLU	N/A	N/A	N/A	N/A	77.05	77.62
Fashion- MNIST (28*28)	Weight	-	EO	88.60	88.58	86.88	86.90	82.30	82.39
	Phase Shifter Angles	Clements	EO	73.22	74.60	73.96	70.98	66.52	47.75
			cReLU	67.62	70.49	71.14	62.77	63.46	60.30
		Reck	EO	73.40	73.98	75.77	72.55	69.95	68.20
			cReLU	66.18	59.30	72.20	69.92	65.72	61.45
		Bokun	EO	N/A	N/A	N/A	N/A	46.07	54.91
			cReLU	N/A	N/A	N/A	N/A	60.27	63.00
CIFAR-10 (3*32*32)	Weight	-	EO	67.95	66.64	68.60	66.93	65.55	66.93
	Phase Shifter Angles	Clements	EO	64.33	63.72	63.12	65.08	59.39	60.48
			cReLU	60.90	60.81	62.71	60.02	56.61	53.10
		Reck	EO	64.62	63.16	63.76	64.58	56.81	56.62
			cReLU	61.82	61.39	63.26	58.55	58.54	54.73
		Bokun	EO	N/A	N/A	N/A	N/A	51.82	54.63
			cReLU	N/A	N/A	N/A	N/A	50.02	50.95

The only exception applied to the case of Reck models with cReLU activation trained on Fashion-MNIST dataset. The $L = 4$ case, corresponding to 64 features used as ONN input, exhibits a trend of overfitting. Therefore, its test accuracy is exceptionally low.

Appendix B

Reconstructed Images from Dimensionality Reduction

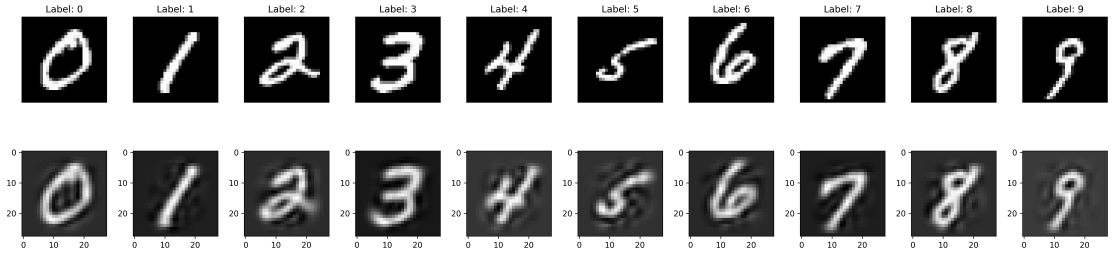


Figure B.1: MNIST images (upper row) and their reconstruction (lower row) with PCA

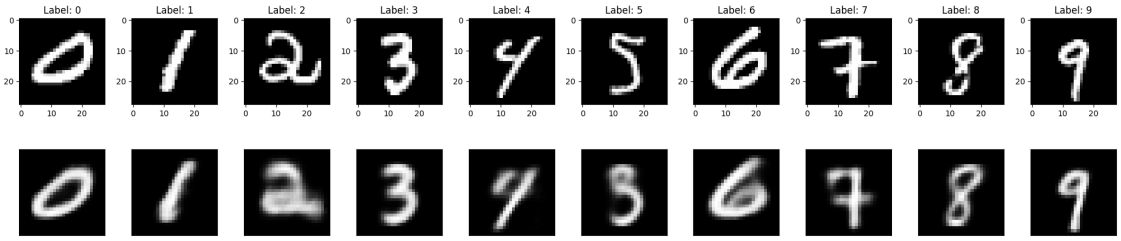


Figure B.2: MNIST images (upper row) and their reconstruction (lower row) from the DAE with fully connected layers

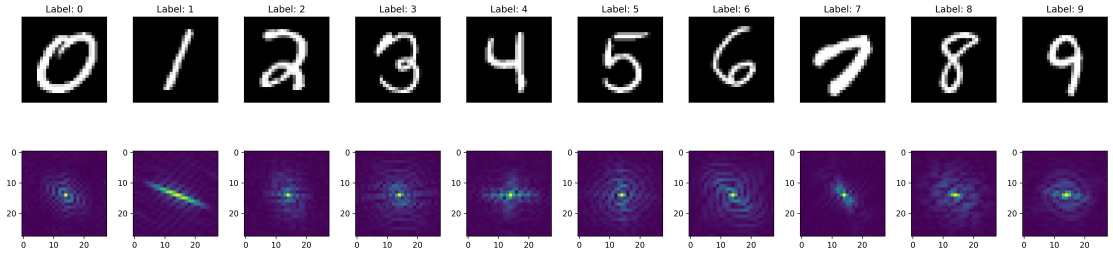


Figure B.3: MNIST images before (upper row) and after (lower row) Fast Fourier Transform

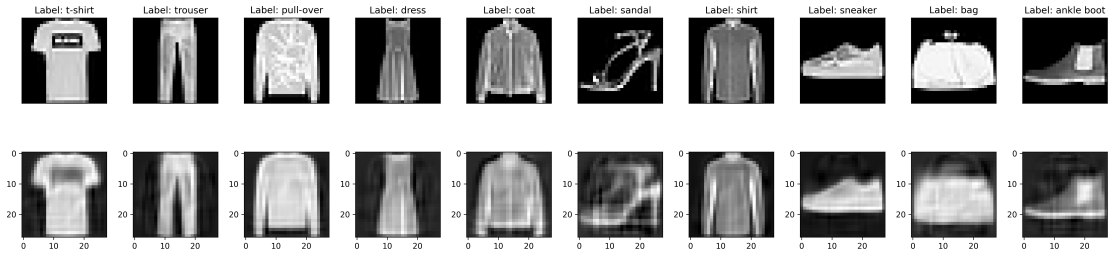


Figure B.4: Fashion-MNIST images (upper row) and their reconstruction (lower row) with PCA

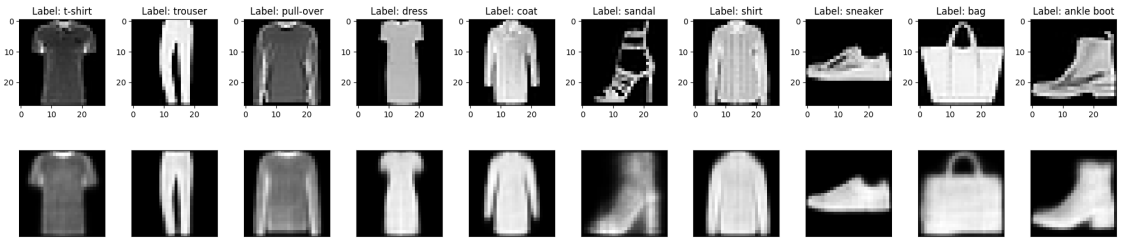


Figure B.5: Fashion-MNIST images (upper row) and their reconstruction (lower row) from the DAE with fully connected layers

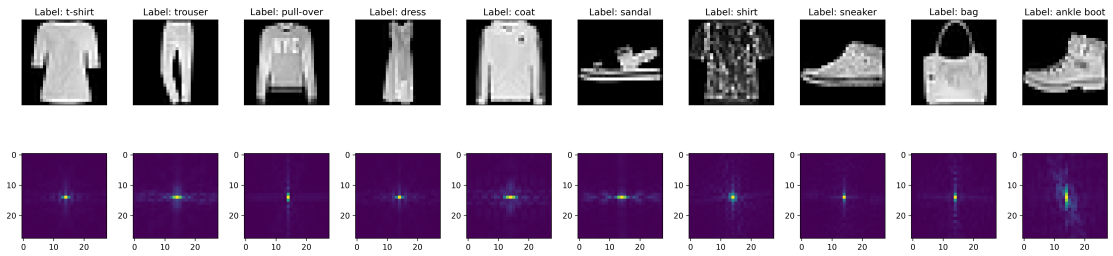


Figure B.6: Fashion-MNIST images before (upper row) and after (lower) Fast Fourier Transform

Bibliography

[1] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature photonics*, vol. 11, no. 7, pp. 441–446, 2017.

[2] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 815–823.

[3] O. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *BMVC 2015-Proceedings of the British Machine Vision Conference 2015*. British Machine Vision Association, 2015.

[4] M. Zulfiqar, F. Syed, M. J. Khan, and K. Khurshid, "Deep face recognition for biometric authentication," in *2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, Swat, Pakistan, 2019, pp. 1–6.

[5] S. Jafri, S. Chawan, and A. Khan, "Face recognition using deep neural network with" livenessnet"," in *2020 International Conference on Inventive Computation Technologies (ICICT)*. Coimbatore, India: IEEE, 2020, pp. 145–148.

[6] M. Z. Khan, S. Harous, S. U. Hassan, M. U. Ghani Khan, R. Iqbal, and S. Mumtaz, "Deep unified model for face recognition based on convolution neural network and edge computing," *IEEE Access*, vol. 7, pp. 72 622–72 633, 2019.

[7] C. Li, R. Wang, J. Li, and L. Fei, "Face detection based on yolov3," in *Recent Trends in Intelligent Computing, Communication and Devices: Proceedings of ICCD 2018*. Springer, 2020, pp. 277–284.

[8] K. Bong, S. Choi, C. Kim, D. Han, and H.-J. Yoo, "A low-power convolutional neural network face recognition processor and a cis integrated with always-on face detector," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 1, pp. 115–123, 2018.

[9] J. Campbell, "Case study: Facial detection and recognition for always-on applications," Synopsys, 2021.

[10] T. F. de Lima, H.-T. Peng, A. N. Tait, M. A. Nahmias, H. B. Miller, B. J. Shastri, and P. R. Prucnal, "Machine learning with neuromorphic photonics," *Journal of Lightwave Technology*, vol. 37, no. 5, pp. 1515–1534, 2019.

[11] F. Shokraneh, M. S. Nezami, and O. Liboiron-Ladouceur, "Theoretical and experimental analysis of a 4 × 4 reconfigurable mzi-based linear optical processor," *Journal of Lightwave Technology*, vol. 38, no. 6, pp. 1258–1267, 2020.

[12] A. Shafiee, S. Banerjee, K. Chakrabarty, S. Pasricha, and M. Nikdast, "Loci: An analysis of the impact of optical loss and crosstalk noise in integrated silicon-photonic neural networks," in *Proceedings of the Great Lakes Symposium on VLSI 2022*, 2022, pp. 351–355.

[13] F. Ashtiani, A. J. Geers, and F. Aflatouni, "An on-chip photonic deep neural network for image classification," *Nature*, vol. 606, no. 7914, pp. 501–506, 2022.

[14] G. B. Huang, M. Mattar, H. Lee, and E. Learned-Miller, "Learning to align from scratch," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[15] W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walmsley, "Optimal design for universal multiport interferometers," *Optica*, vol. 3, no. 12, pp. 1460–1465, 2016.

[16] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5525–5533.

[17] M. T. Rahman and N. Kehtarnavaz, "Real-time face-priority auto focus for digital and cell-phone cameras," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 1506–1513, 2008.

[18] M. Abdel-Mottaleb and L. Chen, "Content-based photo album management using faces' arrangement," in *2004 IEEE International Conference on Multimedia and Expo (ICME)(IEEE Cat. No. 04TH8763)*, vol. 3. IEEE, 2004, pp. 2071–2074.

[19] J. Xiang and G. Zhu, "Joint face detection and facial expression recognition with mtcnn," in *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, 2017, pp. 424–427.

[20] R. Angulu, J. R. Tapamo, and A. O. Adewumi, "Age estimation via face images: a survey," *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, pp. 1–35, 2018.

[21] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," 2010.

[22] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, pp. 137–154, 2004.

[23] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Faceness-net: Face detection through deep facial part responses," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 8, pp. 1845–1859, 2017.

[24] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," in *IEEE international joint conference on biometrics*. IEEE, 2014, pp. 1–8.

[25] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5203–5212.

[26] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 2879–2886.

[27] J. Yan, X. Zhang, Z. Lei, and S. Z. Li, "Face detection by structural models," *Image and Vision Computing*, vol. 32, no. 10, pp. 790–799, 2014.

[28] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014, pp. 1701–1708.

[29] J. Harikrishnan, A. Sudarsan, A. Sadashiv, and R. A. Ajai, "Vision-face recognition attendance monitoring system for surveillance using deep learning technology and computer vision," in *2019 international conference on vision towards emerging trends in communication and networking (ViTECoN)*. IEEE, 2019, pp. 1–5.

[30] J. S. del Rio, D. Moctezuma, C. Conde, I. M. de Diego, and E. Cabello, "Automated border control e-gates and facial recognition systems," *computers & security*, vol. 62, pp. 49–72, 2016.

[31] K. W. Bowyer, "Face recognition technology: security versus privacy," *IEEE Technology and society magazine*, vol. 23, no. 1, pp. 9–19, 2004.

[32] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Maui, HI, USA, 1991, pp. 586–591.

[33] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[34] M. Kim, A. K. Jain, and X. Liu, "Adaface: Quality adaptive margin for face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 750–18 759.

[35] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *CVPR 2011*. IEEE, 2011, pp. 529–534.

[36] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 4510–4520.

[37] S. Chen, Y. Liu, X. Gao, and Z. Han, "Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices," in *Biometric Recognition: 13th Chinese Conference (CCBR) 2018*. Springer, 2018, pp. 428–438.

[38] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[39] N. Zhu, Z. Yu, and C. Kou, "A new deep neural architecture search pipeline for face recognition," *IEEE Access*, vol. 8, pp. 91 303–91 310, 2020.

[40] Q. Cheng, J. Kwon, M. Glick, M. Bahadori, L. P. Carloni, and K. Bergman, "Silicon photonics codesign for deep learning," *Proceedings of the IEEE*, vol. 108, no. 8, pp. 1261–1282, 2020.

[41] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, "Experimental realization of any discrete unitary operator," *Physical review letters*, vol. 73, no. 1, p. 58, 1994.

[42] F. Shokraneh, S. Geoffroy-Gagnon, and O. Liboiron-Ladouceur, "The diamond mesh, a phase-error-and loss-tolerant field-programmable mzi-based optical processor for optical neural networks," *Optics Express*, vol. 28, no. 16, pp. 23 495–23 508, 2020.

[43] K. H. R. Mojaver, B. Zhao, E. Leung, S. M. R. Safaee, and O. Liboiron-Ladouceur, "Addressing the programming challenges of practical interferometric mesh based optical processors," *Opt. Express*, vol. 31, no. 15, pp. 23 851–23 866, Jul 2023.

[44] T. W. Hughes, M. Minkov, Y. Shi, and S. Fan, "Training of photonic neural networks through in situ backpropagation and gradient measurement," *Optica*, vol. 5, no. 7, pp. 864–871, 2018.

[45] F. Shokraneh, S. Geoffroy-Gagnon, M. S. Nezami, and O. Liboiron-Ladouceur, "A single layer neural network implemented by a 4×4 mzi-based optical processor," *IEEE Photonics Journal*, vol. 11, no. 6, pp. 1–12, 2019.

[46] I. A. D. Williamson, T. W. Hughes, M. Minkov, B. Bartlett, S. Pai, and S. Fan, "Reprogrammable electro-optic nonlinear activation functions for optical neural networks," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–12, Jan. 2020.

[47] H. Zhang, M. Gu, X. Jiang, J. Thompson, H. Cai, S. Paesani, R. Santagati, A. Laing, Y. Zhang, M. Yung *et al.*, "An optical neural chip for implementing complex-valued neural network," *Nature communications*, vol. 12, no. 1, p. 457, 2021.

[48] G. Mourgias-Alexandris, M. Moralis-Pegios, A. Tsakyridis, S. Simos, G. Dabos, A. Totovic, N. Passalis, M. Kirtas, T. Rutirawut, F. Gardes *et al.*, "Noise-resilient and high-speed deep learning with coherent silicon photonics," *Nature Communications*, vol. 13, no. 1, p. 5572, 2022.

[49] G. Mourgias-Alexandris, A. Totović, A. Tsakyridis, N. Passalis, K. Vyrsokinos, A. Tefas, and N. Pleros, "Neuromorphic photonics with coherent linear neurons using dual-iq modulation cells," *Journal of Lightwave Technology*, vol. 38, no. 4, pp. 811–819, 2020.

[50] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[51] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[52] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[53] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 2016, pp. 5525–5533.

[54] C. Ramey, "Silicon photonics for artificial intelligence acceleration: Hotchips 32," in *2020 IEEE Hot Chips 32 Symposium (HCS)*, 2020, pp. 1–26.

[55] J. Schanda, *Colorimetry: Understanding the CIE System*. Wiley, 2007. [Online]. Available: https://books.google.ca/books?id=uZadszSGe9MC

[56] S. P. Baller, A. Jindal, M. Chadha, and M. Gerndt, "Deepedgebench: Benchmarking deep neural networks on edge devices," in *2021 IEEE International Conference on Cloud Engineering (IC2E)*, 2021, pp. 20–30.

[57] S. Thoziyoor, J. H. Ahn, M. Monchiero, J. B. Brockman, and N. P. Jouppi, "A comprehensive memory modeling tool and its application to the design and analysis of future memory hierarchies," in *2008 International Symposium on Computer Architecture*, 2008, pp. 51–62.

[58] R. Hong, T. Kohno, and J. Morgenstern, "Evaluation of targeted dataset collection on racial equity in face recognition," in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, 2023, pp. 531–541.

[59] Z. Yin, W. Gross, and B. H. Meyer, "Probabilistic sequential multi-objective optimization of convolutional neural networks," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 1055–1060.

[60] B. Bartlett, M. Minkov, T. Hughes, and I. A. D. Williamson, "Neuroptica: Flexible simulation package for optical neural networks," https://github.com/fancompute/neuroptica, 2019.

[61] Q. Fournier and D. Aloise, "Empirical comparison between autoencoders and traditional dimensionality reduction methods," in *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, Sardinia, Italy, 2019, pp. 211–214.

[62] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[63] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2014.

[64] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," in *2018 6th International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada, 2018.

[65] A. Das, G. Zhang, H. R. Mojaver, and O. Liboiron-Ladouceur, "Low loss 8 × 8 silicon photonic banyan switch," in *2020 IEEE Photonics Conference (IPC)*, 2020, pp. 1–2.

[66] M. Al-Qadasi, L. Chrostowski, B. Shastri, and S. Shekhar, "Scaling up silicon photonic-based accelerators: Challenges and opportunities," *APL Photonics*, vol. 7, no. 2, 2022.

[67] A. Masood, M. Pantouvaki, G. Lepage, P. Verheyen, J. Van Campenhout, P. Absil, D. Van Thourhout, and W. Bogaerts, "Comparison of heater architectures for ther-

mal control of silicon photonic circuits," in *10th International Conference on Group IV Photonics*, 2013, pp. 83–84.

[68] S. Swayamdipta, R. Schwartz, N. Lourie, Y. Wang, H. Hajishirzi, N. A. Smith, and Y. Choi, "Dataset cartography: Mapping and diagnosing datasets with training dynamics," *arXiv preprint arXiv:2009.10795*, 2020.

[69] J. Gu, Z. Zhao, C. Feng, H. Zhu, R. T. Chen, and D. Z. Pan, "Roq: A noise-aware quantization scheme towards robust optical neural networks with low-bit controls," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2020, pp. 1586–1589.

[70] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Conference on fairness, accountability and transparency*. PMLR, 2018, pp. 77–91.

[71] Y. Yang, A. Gupta, J. Feng, P. Singhal, V. Yadav, Y. Wu, P. Natarajan, V. Hedau, and J. Joo, "Enhancing fairness in face detection in computer vision systems by demographic bias mitigation," in *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, 2022, pp. 813–822.

[72] V. Albiero, K. Zhang, and K. W. Bowyer, "How does gender balance in training data affect face recognition accuracy?" in *2020 ieee international joint conference on biometrics (ijcb)*. IEEE, 2020, pp. 1–10.

[73] K. Karkkainen and J. Joo, "Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1548–1558.