DEEP LEARNING FOR SECURITY SYSTEMS FROM CAMOUFLAGED THREAT DETECTION TO QUANTUM KEY PROTECTION

A Thesis Submitted

In Partial Fulfilment of the Requirements for the Degree of

MASTERS OF TECHNOLOGY
IN
Data Science

Submitted by

Lavish Kumar

23/DSC/23

Under the supervision of

Dr. Shweta Meena

Assistant Professor, Department of Software Engineering

Delhi Technological University

Department of Software Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

JUNE, 2025

DEPARTMENT OF SOFTWARE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Lavish Kumar, Roll No - 23/DSC/23 student of M.Tech (Data Science), hereby certify that the work which is being presented in the thesis entitled "Deep Learning for Security Systems from Camouflaged Threat Detection to Quantum Key Protection" in partial fulfilment of the requirements for the award of degree of Master of Technology, submitted in the Department of Software Engineering, Delhi Technological University is an authentic record of my own work carried out during the period from Jan 2025 to May 2025 under the supervision of Dr. Shweta Meena.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other institute.

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor **Signature of External Examiner**

DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "Deep Learning for Security Systems from Camouflaged Threat Detection to Quantum Key Protection" which is submitted by Lavish Kumar, Roll No - 23/DSC/23, Department of Software Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Dr. Shweta Meena

Assistant Professor

Date:

Department of Software Engineering, DTU

DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to Dr. Shweta Meena for her continuous guidance and mentorship that she provided me during the project. She showed me the path to achieve our targets by explaining all the tasks to be done and explained to me the importance of this project as well as its industrial relevance. She was always ready to help me and clear my doubts regarding any hurdles in this project. Without her constant support and motivation, this project would not have been successful.

Place: Delhi — Lavish Kumar

Date: — (23/DSC/23)

Abstract

Because today's world is so interconnected and digital, it is now even more crucial to keep both physical and digital systems secure. With rapid growth and practical uses, deep learning is becoming very important in facing current security difficulties. Aspects of system security such as camouflaged object detection and quantum communication protection, are discussed in this thesis.

Detecting camouflaged objects remains a difficult problem. Many domains rely heavily on the usefulness of this field. It happens because the object we want to find looks similar to its background. There are many strategies and datasets being developed to deal with this issue and this field has emerged as a rapid growth point in image processing. We tested EfficientDet with SAM on NC4K and compared the results to what some existing models show. By analysing why the model failed, we have suggested areas for improvement in future projects. In this part, the thesis compares EfficientDet and SAM to various COD models and also examines how the new NC4K dataset performs.

This thesis further examines the topic of system security by exploring how deep learning can support quantum communication. In the second section, we study the use of neural networks for error correction in QKD. To study five architectures, a new dataset of 120,000 observations was created, where both noise probabilities and photon transmission rates varied. From these findings, it is clear that AI can improve exiting quantum communication protocols.

Contents

Candidate's Declaration . . . . . **i**

Certificate . . . . . **ii**

Acknowledgement . . . . . **iii**

Abstract . . . . . **iv**

Content . . . . . **vi**

List of Tables . . . . . **vii**

List of Figures . . . . . **viii**

1 INTRODUCTION **1**

1.1 Overview 1

1.2 Problem Statement 4

1.3 Dataset 5

1.3.1 NC4K 5

1.3.2 Qiskit Generated Dataset 6

1.4 Research Objective and Contribution 7

2 RELATED WORK **9**

2.1 Camouflaged Object Detection 9

2.1.1 Classical Models and Benchmarks 9

2.1.2 Advances in Dataset Creation 10

2.1.3 Recent Advancements 11

2.2 Quantum Key Distribution 12
2.2.1 Early Work 12
2.2.2 Machine Learning's Involvement in QKD 14

3 METHODOLOGY 15
3.1 Proposed Methodology for COD 16
3.1.1 Our Novel Approach 16
3.1.2 Existing Results 17
3.2 Proposed Methodology for QKD 18
3.2.1 QBER Significance 18
3.2.2 Data Generation 19
3.2.3 Model Selection 20
3.2.4 Performance Evaluation 23

4 EXPERIMENTAL SETUP 25
4.1 NC4K 25
4.2 EfficientDet 26
4.3 SAM 27
4.4 Qisbit 27
4.5 Google Colab 28

5 RESULTS AND DISCUSSION 30
5.1 Overall Outcome for NC4K Dataset 30
5.2 Overall Outcome for Quantum Key Prediction 31
5.2.1 Different Measurement Metrics 31
5.2.2 Error Distribution 32
5.3 Limitation 35

6 CONCLUSION AND FUTURE SCOPE 37
6.1 Overall Conclusion 37
6.2 Future Scope 39

List of Tables

3.1 Performance of different models on the NC4K dataset 17

5.1 Performance comparison of models over different evaluation metrics 31

5.2 Computational performance of models 35

List of Figures

1.1 Distribution of data points in the generated dataset 3

1.2 Image segmentation of camouflaged animals 6

1.3 Distribution of data points in the generated dataset 7

2.1 Performance of BiRefNet 11

2.2 Performance of various models on the CAMO dataset 12

3.1 Feature Pyramid Network architecture 16

3.2 Network architecture of EfficientDet 17

3.3 Correlation Heatmap 20

4.1 A picture of a fish from the NC4K dataset 25

4.2 Image of an instance of a fish from the NC4K dataset 26

4.3 SAM being used for masking an image 28

5.1 Results of EfficientDet with SAM on NC4K 31

5.2 Error Distribution chart of MLP 33

5.3 Error Distribution chart of LSTM 33

5.4 Error Distribution chart of GRU 33

5.5 Error Distribution chart of CNN 34

5.6 Error Distribution chart of Autoencoders 34

Chapter 1

INTRODUCTION

1.1 Overview

Two security-focused areas, Camouflaged Object Detection (COD) and Quantum Key Distribution (QKD), are analysed in this research for the potential use of Deep Learning. There are unique issues associated with every domain. The challenge lies in telling what's concealed from other items. Errors in quantum transmission need to be fixed as well. Both works used enhanced AI models to improve detection, while maintaining consistency, in spite of differences. It becomes easier to find solutions that grow with your enterprise. There are adaptable solutions included within security systems as well. With this double domain method, we analyse how smart systems can detect security risks. They protect areas from danger in real life and from digital breaches. By examining concealed object detection and quantum key distribution at the same time, we recognise how AI can impact fields beyond its own. It helps people to notice the finer points of things.

Camouflaged Object Detection (COD) is a crucial and essential task in Visual Computing, specifically Computer Vision (CV) due to the great resemblance between the target entity and its surrounding environment. The traditional object detection techniques fail for this task due to the great resemblance between the targeted entity and its surrounding environment. Hence, image segmentation plays an important role in COS to overcome this challenge.

After that, we turn them into a form that is simpler to learn about. Each part of a mixture is called out by its own characteristic label. It helps when you are trying to distinguish the camouflaged object from its surroundings.

QKD proved to be an impressive addition to cryptologic protocols because it uses quantum physical rules to achieve exceptional security [1] [2]. Traditional encryption depends on uncertain methods, while QKD relies on quantum entanglement and the principle that you can't copy a quantum state [3] [4]. QKD is expected to be stable, yet it experiences many difficulties, mainly because error correction is influenced greatly by noise, lost photons and possible eavesdropping. Abnormal losses, unwanted noise and device imperfections during quantum sharing can result in bits being out of order in the key given to Alice and Bob. If the issue is ignored, the security of the key will suffer and the protocol will not be as efficient. For this reason, fixing errors is vital in QKD since it decides if the key that was exchanged is safe to use.

The Cascade Protocol fixes these errors by running many parity cheques on the data sent [5]. Although this algorithm is efficient, too much computational and communication workload can block its ability to work in real time and scale smoothly. Using the newest innovations in optical communication along with QKD equipment allows for quick transfers of data and a higher number of photons to travel, so this method is practical where distance is high or signals can be muddled, as found in satellite connexions. Still, these situations makes it easier to recognise where the system is not working as efficiently as possible [6], [7]. In these conditions, delays and not enough resources can stand in the way of main reconciliation processes [8]. If QKD is to be used effectively in today's high-volume networks, cloud networks and IoT, it needs to include good and adjustable error correction methods.

Fig. 1.1 shows the overall working of the autoencoder model on both Alice's and Bob's sides, with a focus on predicting the QBER and integrating it with the Cascade protocol for error correction [9].

The journey begins with everything Alice does. She obtains a series of shifted data X_A which is the raw key bits that result from the key reconciliation with Bob. Autoencoder

extracts the transformed data from here on. It is important to have this latent representation so a meaningful Quantum Bit Error Rate (QBER) can be derived. QBER serves as a required measure to support the beginning of error correction. Following encoding, Alice sends her latent data as a message to Bob over a quantum channel.

When working with Bob, he assumes the data is sifted, so he sees it as X_B. Eve has the same data as Alice does. When Bob has the obtained latent vector, he uses the decoder function from the autoencoder. It reconstructs and strengthens the first main idea. Cascade utilises Alice's QBER prediction while setting up its protocols.

They then proceed to use error correction and finish the process by making the final secret key. To cheque that their keys are identical and safely generated, they use a randomly picked hash function.

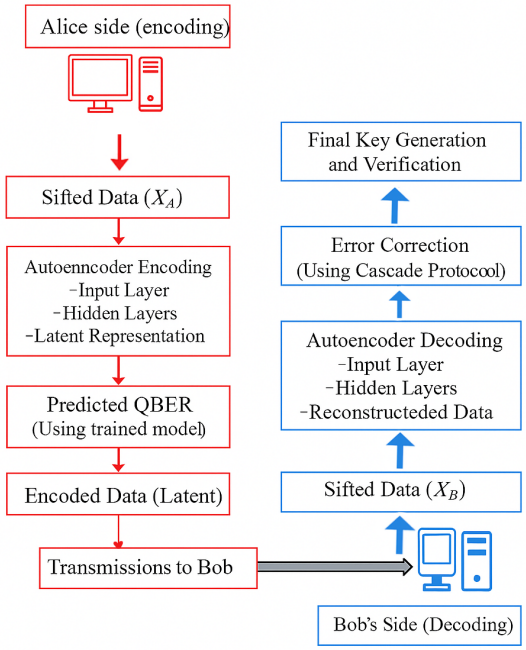

Figure 1.1: Distribution of data points in the generated dataset

As Artificial Intelligence (AI) and Machine Learning (ML) advance, their importance in improving complex networks and cryptography is now more obvious. Some of the Deep Learning models mentioned such as RNNs, CNNs and Autoencoders, allow for streamlining error correction and making a QKD system more secure.

1.2 Problem Statement

As Artificial Intelligence (AI) and Machine Learning (ML) advance, their importance in improving complex networks and cryptography is now more obvious. Some of the Deep Learning models mentioned such as RNNs, CNNs and Autoencoders, allow for streamlining error correction and making a QKD system more secure.

Without filling this gap, COD research moves more slowly. There are many excellent algorithms that reach good results on the NC4K dataset, but they must be tested. For the first time, we apply EfficientDet and SAM on the NC4K dataset to try and make a difference. Let's apply EfficientDet and SAM on NC4K to support the future study of corrupted objects and learn valuable things. The analysis will show how accurately the model detects different forms of camouflage and suggest ways to make it better.

While COD helps identify hidden elements in natural pictures, QKD resolves the issues and misalignment caused by noise in transmission using quantum technology. Still, each problem needs the ability to understand patterns that are usually hidden from regular algorithms. While QKD ensures great security, the error correction step remains a major issue because scalability, security threats and computational problems get in the way. Cascade Protocol mainly uses rounds of exchanging information that cause more time and excess computation power to be spent, compared to other methods [5], [10]. When photons are sent faster, the possibility of error goes up as well. Secure key exchange in telecommunications over high photon rates calls for an even faster and shiftable correction system. The difficulties can be improved by applying Neural Networks to forecast and enhance the optimization of error correction services in QKD. The use of AI models in this paper lets us reduce computational efforts, fix errors more easily and secure the Cascade Protocol which makes QKD better fit for practical use. Thanks to NNs being good at pattern recognition, our technique deals with bad bits in a quicker and easier way, helping to reach higher throughput.

1.3 Dataset

1.3.1 NC4K

A significant and extensive dataset is needed for successful Camouflaged Object Detection (COD). For years, a range of datasets have been created to solve the issue which leads to significant progress in research on COD and COS.

A well-known dataset for building COD models is CAMO which was first released in 2017 [11]. This information is oriented toward Camouflaged Object Segmentation (COS), an issue where the goal is to separate the object from its environment. The data includes 1250 photographs split into two data sets, one for training and one for testing, with examples of both camouflaged animals and camouflaged man-made objects. The ground truth information was provided in COCO JSON which did not include a great deal of detail. Then, the CAMO++ dataset [12] appeared in 2021 and it is most likely bigger than CAMO and offers detailed pixel segmentation annotations.

In particular, the COD10K team [13] developed an important dataset called COD10K in 2020 which has continued to be used and updated and is often used to measure the performance of COD algorithms. The dataset includes 10,000 images and the subjects covered include land animals, aquatic life and objects that mask their appearance in nature, made by humans. Each image in the dataset has Bounding Boxes, Categories and Attributes, labels for Objects and Instances and annotation of Edges.

We have used the NC4K dataset [14] for this paper. This dataset is comparatively new and has nearly 4,000 images. The images vary across various fields, from naturally camouflaged objects to artificially camouflaged objects. Its ground truth is similar to the CAMO++'s ground truth. As this is a new dataset there is relatively less work done on this dataset. Some instances of the NC4K dataset can be seen in Fig. 1.2.

As the NC4K dataset tests segmentation models with camouflage from real cameras, making quantum datasets of your own will require learning algorithms to handle errors in quantum communication. Due to progress in QKD, there is little data available which means researchers must generate new datasets to encourage further study in quantum

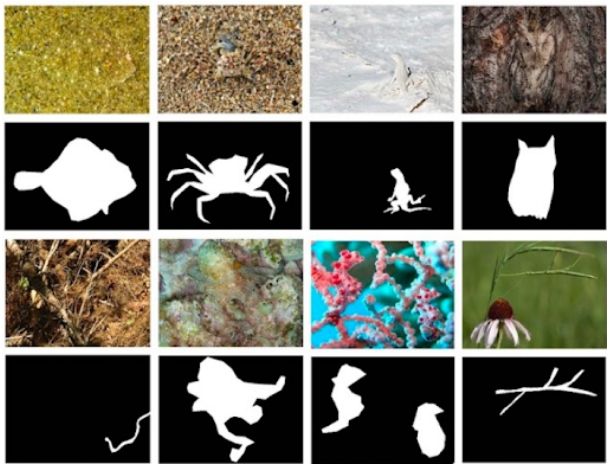

Figure 1.2: Image segmentation of camouflaged animals

communication. While datasets are available, they mainly highlight certain QKD topics such as taking measurements of amplitude in thermal state QKD [15], reviewing QBER values across many signal losses [16] and showing plans for space-to-ground QKD [17]. While these datasets are useful for many QKD areas, they miss the main focus of this study which is to optimise error correction in the Cascade Protocol using neural networks.

1.3.2 Qiskit Generated Dataset

Using Qiskit, a free and open quantum computing framework developed by IBM, we engineered a dataset to help us study this research gap [18]. People in quantum research use Qiskit for both improving quantum circuit designs and reducing errors. We have organised a dataset that was created to support AI error correction in quantum key distribution. The data includes 120,000 records and each record covers transmission rates from 1,000 to 12,000 kbps along with noise probabilities between 0.01 and 0.15. For a reliable comparison of different neural networks in QKD error correction, we chose attributes such as QBER, Sifted Key Length, SNR and Final Key Length. A plot of the data can be found in Fig. 1.3.

This dataset is made to give AI models reliable and realistic experience of working in QKD systems, making the study useful for the real world. Unlike current datasets, this one provides an entire assessment setup for ML in QKD error correction. Access to these datasets is needed to advance research in AI-based cryptography and improve how QKD

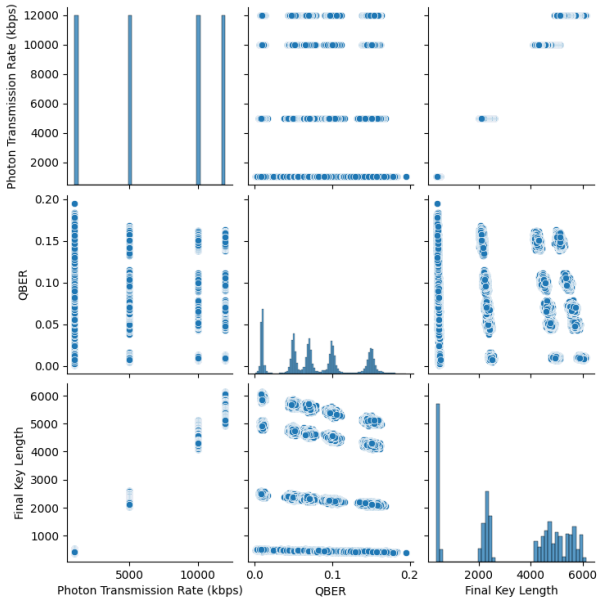

Figure 1.3: Distribution of data points in the generated dataset

protocols are used in practise.

Running the model on over 120,000 examples of data with different transmission conditions increases its potential for general use. Because of this, the model works well in simulated situations with different noisy conditions. Thanks to simulation, we can easily adjust and increase experiments which is practical given how challenging and costly it is to operate physical QKD setups.

1.4 Research Objective and Contribution

For COD, we utilised the EfficientDet algorithm. EfficientDet is a model that is recognised for achieving a good balance of results and power efficiency. In 2020, Google Research first introduced this model. It distinguishes itself by having successful object detection with less energy consumption than current best models. Once this is done, it tends to fit well on devices with few computer resources. The essential features Strengthening Networks and BiFPN are important parts of its design.

The Segment Anything Model (SAM), created by Meta AI, has been another tool we've worked with [19]. Through this model, objects can be parted in images, even though those objects had not been observed before during training. Thanks to Zero-shot Segmentation,

it identification objects and separate them into regions without needing extra training data for them. The architecture contains an Image Encoder, Prompt Encoder and Mask Decoder.

In our experiments with QKD, we have studied how ML can be used in QKD error correction by training and comparing different neural network architectures. Our main objective is to use deep learning models to make it easier, scale up the error correction process and increase its security. We address error patterns, choose better QKD settings and refine key reconciliation by using Autoencoders, CNN, GRU, LSTM and MLP.

In this article, the performance of several neural network designs is compared in QKD error correction using a newly introduced dataset. LSTMs and GRUs help catch connexions between elements in a series, whereas Autoencoders reduce the number of dimensions to allow more accurate and efficient correction methods using data. By looking at these models, the research tries to find the best design for minimising QBER and producing the longest final keys. By making AI use more efficient and secure, we hope to create QKD systems that are easily used in today's cryptography. Our goal with optimising Cascade is to make latency less, increase accuracy and lower the amount of information that must be exchanged using classical methods. In turn, this style produces scalable AI-supported QKD that might be used for real-time, high-speed routines in environments with limited resources.

Chapter 2

RELATED WORK

In this section, we have discussed about the related work done in the field of Camouflaged Object Detection as well as Quantum Key Distribution.

2.1 Camouflaged Object Detection

2.1.1 Classical Models and Benchmarks

These days, there are big advancements in COD, mainly because of developments in image segmentation, machine learning and computer vision. Many approaches and techniques have been put into practise to solve this problem. The literature on COD is summarised in the text below. We paid close attention to major techniques, datasets and methods. Additionally, we examine research connected to SAM and zero-shot learning. After examining the research, we have found where there are opportunities to make further progress in COD.

Among the first models proposed for object detection in camouflaged settings are BASNet [20] and EGNet [21]. They were each added in the 2019 update. BASNet designed a new refine architecture meant for detecting salient objects and that pays attention to boundaries. Residual refinement and densely supervised Encoder-Decoder play key roles in its process. As a result, salient object detection's boundary quality improved thanks to this development. The network guides its learning of image conversion using IoU, BCE

and SSIM as part of a combined loss function at various levels of image hierarchy.

EGNet was introduced by Wang et al. [21] to address the challenge of coarse object boundaries in salient object detection by upgrading the relationship between prominent entity information and salient edge information that is complementary. To continually describe these two types of complementing information in a single network, the model included an edge guidance network (EGNet).

The models SiNet [22] and PraNet [23] made their appearance in 2020. They brought about important improvements in the fields of COD and medical image segmentation. This approach uses a straightforward system to address the problem of COD. Wang et al. built a Search Identification Network (SINET) that performed well on all tested COD tasks. This model relies on the newly released dataset COD10K which features 10,000 densely labelled images.

2.1.2 Advances in Dataset Creation

There are times when objects are so well hidden by their surroundings that they are hard to differentiate, so SINet-V2 was created to handle this. The authors presented the COD10K dataset which contains images of 10,000 camouflaged objects that appear in different real-world views. All together, there are 78 different categories in the guide, from animals to man-made objects. The information in the dataset covers object types, edges, difficult points and particular instances, making it the most annotated dataset on COD as of now. SINet-V2 shows improved results on every dataset used for testing. As a result, SINet-V2 outlined what is next for the area of COD.

In 2021, Le et al. [24] put forward a new COD task known as camouflaged instance segmentation that seeks to disassemble camouflaged objects in images into understandable elements. To support their investigation, the authors enlarged the CAMO dataset and added CAMO++ which contains more and varied images with labels for every pixel grouped by importance. Applying CAMO++ data, the study evaluated recent instance segmentation methods and set a benchmark for disguised instance segmentation. Even better performance was achieved using a Camouflaged Fusion Learning (CFL) framework.

The method is available for anyone to access on the project page and helps with research on camouflaged instance segmentation.

Improvements in camouflaged object detection and foreground segmentation have been made with EVPv2 [25], ZoomNeXT [26] and BiRefNet [27]. In 2023, EVPv2 became available and acts as a joint solution for many foreground segmentation tasks, e.g., SOD, Defocus Blur Detection and COD. In this work, a new model, Explicit Visual Prompting (EVP), was proposed and it improved the techniques used in pre-training and prompt-tuning that come from NLP. Compared to other parameter-efficient fine-tuning solutions working on many datasets, EVP showed better results by fine-tuning parameters that focus on what is unique in each image's visual part. The results of BiRefNet are shown in Fig. 2.1.

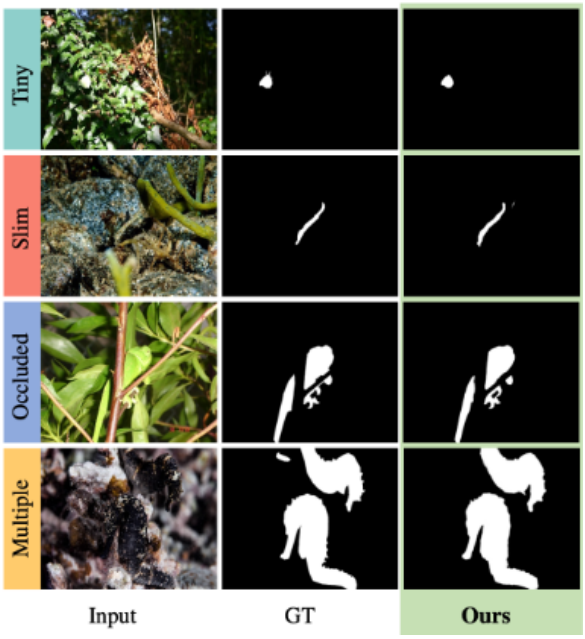

Figure 2.1: Performance of BiRefNet

2.1.3 Recent Advancements

ZoomNeXT [26] was also introduced in 2023 and addresses the complexity of camouflaged object detection by proposing an effective unified collaborative pyramid network. The model works by zooming strategy to learn discriminative mixed-scale semantics and explores subtle clues between targeted objects and background surroundings. Addition-

ally, ZoomNeXT introduces a simple yet effective regularisation called uncertainty awareness loss to support predictions with higher confidence in candidate regions. Its task-friendly framework surpassed the existing state-of-the-art methods in image and video-camouflaged object detection benchmarks.

In 2024, BiRefNet [27] appeared and offers a new way to improve DIS by using bilateral reference images. Two modules, called the Localization Module (LM) and the Reconstruction Module (RM) with Bilateral Reference (BiRef) are used alongside each other to assist in object localization by making use of global semantic information as well as for rebuilding objects. A new method was added to help the model pay more attention to details. BiRefNet manages to surpass task-specific methods on all the benchmarks tested. All their performance results are displayed in Fig. 2.2.

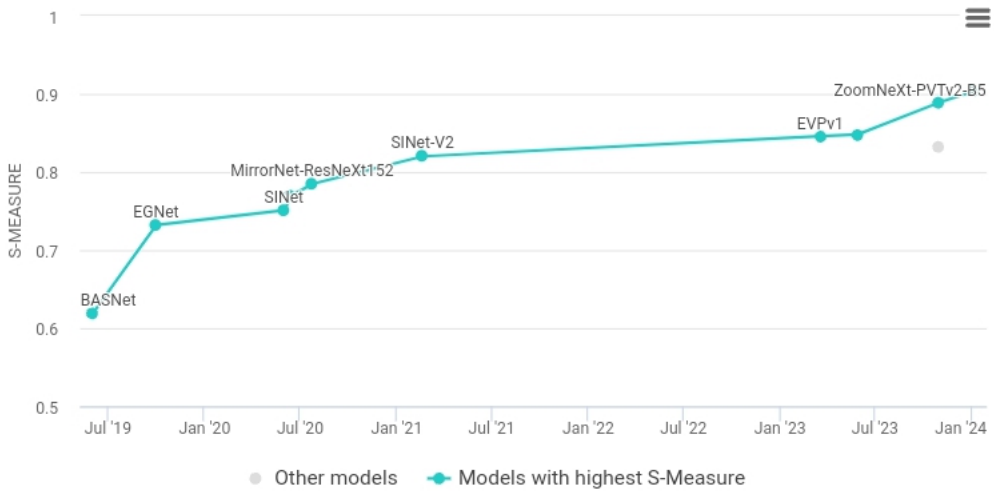

Figure 2.2: Performance of various models on the CAMO dataset

2.2 Quantum Key Distribution

2.2.1 Early Work

The basis of QKD lies in the first steps taken in quantum cryptography research and practise. A short while later, in 1997, Ekert designed the E91 protocol that used quantum entanglement and Bell's theorem to detect anyone listening in and developed a further

important method for quantum cryptography [28]. Under these rules, Bennett created and introduced the B92 protocol in 1992. Different from BB84 protocol, where four orthogonal quantum states are involved, B93 relies only on two similar quantum states, so the setup is easier, but the process is more vulnerable to interference. It created a simple yet clever method of quantum key distribution to show that quantum ideas play a big role in cryptography. In parallel, a toolkit was built to model real-life QKD systems so that they could address common experiments and highlight how these implementations might be tested in practise [29]. A QKD model was created by SeQureNet which included photon loss, imperfect detectors and key effects as limitations [30]. Using models made it easier for researchers to create working protocols. The improvement in QKD led researchers to review its issues and find that its ineffective error correction, scaling challenges and low data rate are the main things keeping QKD from being widely used [31].

Next, a major innovation in error reconciliation made it possible for Alice and Bob to minimise the information shared with Eve while fixing errors in their key. Initially, the Cascade protocol was the main technique for handling errors in coding, but Low Density Parity Cheque (LDPC) codes were introduced later and now provide very good performance in high noise conditions [32]. Earlier ways of resolving these tasks could only deal with noisy channels, whereas these new protocols solve them much more efficiently and on noiseless channels [33]. In these experiments, China and the US have demonstrated that QKD is reliable over longer ranges and in free space which establishes its secure use in such areas.

Recent progress in QKD is possible thanks to ML techniques that have solved many difficulties in quantum communication [34], [35]. A variety of ML models are now used to predict how much noise will occur in long-distance QKD systems [36]. Among these models are ones such as Support Vector Machines (SVMs), Linear Regression and others. Each type of deep learning model is designed to address specific prediction activities, including changes in trends, variations in hardware noise and effects due to environmental instability. In addition, many additional approaches have been developed to fine-tune methods for estimating key rates and to improve how errors are corrected. Unlike tradi-

tional methods, ML models quickly respond to new quantum channel conditions and cut down the number of keys discarded during QKD communication.

In QKD, the Quantum Bit Error Rate (QBER) explains how well the quantum channel works and if anyone might be listening to the data. ML is being applied in several studies to improve error correction and the prediction of QBER in QKD [37], [38]. With the use of different ML techniques, QBER has been roughly calculated and the key reconciliation processes have been smoothened, leading to faster implementations of QKD [39]. Using neural networks trained on simulation and experiment results, QBER prediction is accurate for different channel and system scenarios. Seeking to improve the Cascade Protocol, an autoencoder approach is presented along with high training accuracy of 99% [9].

2.2.2 Machine Learning's Involvement in QKD

Several times, researchers have shown that faster and more efficient processing is possible with ML added to QKD systems compared to older methods [40]. In detail, one approach made convergence 40% faster and required up to 30% fewer computations. We observe that both convergence time and computational complexity are greatly decreased [41]. It is clear from the data that machine learning can improve QKD by fixing some challenges and creating more adaptable networks. By employing several neural networks, we calculated the key lenght in QKD and we created a fresh dataset to teach the models used in QKD research. In this collection, more than 10,000 data points have been collected, covering channel states, numbers of photons and the key lengths that can help create later QKD prediction models.

ML is now recognised for its growing impact on security and efficiency in QKD, thanks to these practise experiments. Uniting ML models and QKD technology significantly speeds up error fixing, sharpens security defences and makes moving keys more efficient. ML can detect possible spying or device tampering due to oddity finding and still remain truthful when different threats occur.

Chapter 3

METHODOLOGY

This work examines two technologies that may differ in their implementations but still follow the same scientific methods: Camouflaged Object Detection (COD) and Quantum Key Distribution (QKD). Whereas COD works on improving visual perception with object detection and segmentation in places with low visibility, QKD adds to quantum cryptography by making secure key exchanges more accurate. However, both Computer Vision and Medical Imaging use advanced statistics, loose neural structures and measure effectiveness using metrics that target accuracy. Because we share the same principles, our larger objective is to use machine learning advances to help make accuracy and reliability better in areas where traditional systems struggle.

For this first stage, EfficientDet and SAM are applied to the NC4K dataset to improve the location and outline of camouflaged objects in high definition images. In the second part, we perform real QKD experiments with Qiskit and gather a range of results which we then analyse with various neural networks to find ways to minimise the QBER. Both methods combine deep learning with the unique challenges in visual and quantum areas so as to provide practical and advanced solutions that meet higher accuracy, security and efficiency levels.

3.1 Proposed Methodology for COD

3.1.1 Our Novel Approach

We use the NC4K dataset as the base for our approach. All of the 4,121 high resolution pictures contain pre-labelled segmented images which accurately highlight where the hidden camouflaged objects are. By having prepared ground truth, we didn't need to go through every image ourselves and mark out objects for training.

Our work relies on the D5 version of the EfficientDet model that provides an optimal level of accuracy and efficiency. Before being used, the D5 model examines a large set of images to help with feature extraction. EfficientDet gets the NC4K dataset to train itself on recognising camouflaged objects from their ordinary surroundings.

To know how EfficientDet works we first have to draw light on Feature Pyramid Network (FPN). It works on the standard idea of executing the algorithm on numerous resolutions of the same image in the hope of catching both small and large scale phenomena. Iqbal et al. presented Fig. 3.1 in their paper [42] that in FPN they use feature maps on different resolutions instead of the different resolution images. In Fig. 7, the traditional backbone of CNN is represented by the bottom up and the feature fusion at different scales is represented by the top down. The concept after the lateral connections was to join low resolution feature maps that are rich in features with less meaningful feature maps with high resolution.

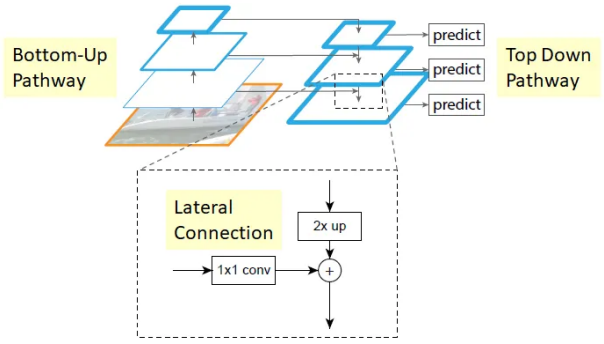

Figure 3.1: Feature Pyramid Network architecture

According to Fig. 3.2 presented by Niu et al. [43] in their paper, the BiFPN acts as

the feature network, that continuously applies bottom up and top down bi feature fusion. These mixed features are given to a box network and class to generate bounding box predictions and object class. They are shared at all levels of features equally.

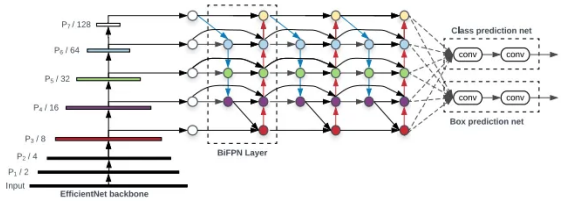

Figure 3.2: Network architecture of EfficientDet

3.1.2 Existing Results

We detected objects using EfficientDet, but also used SAM to enhance the segmentation of concealed items. SAM's zero shot segmentation model benefits from both the images and the known ground truth masks of the NC4K dataset. Working together, SAM and EfficientDet helped SAM find a way to connect image features and object borders, possibly leading to improved accuracy of those segmentation masks produced by EfficientDet.

For these tasks, IoU and Dice coefficient are commonly accepted evaluation metrics, so we used them. IoU shows what parts of the predicted mask and the ground truth match and Dice coefficient cheques how similar their union and intersection are.

We aim to achieve an accurate and robust system for camouflaged object detection in diverse natural environments by implementing the combined strengths of EfficientDet and SAM. Some comparative results on the NC4K dataset are presented in Table 3.1.

Table 3.1: Performance of different models on the NC4K dataset

Model	**S Measure**	**Weighted F Measure**	**MAE**	**Year**
BiRefNet	0.915	0.890	0.0023	2024
ZoomNeXt PVTv2 B5	0.903	0.863	0.028	2023
ZoomNeXt PVTv2 B4	0.900	0.865	0.028	2023
ZoomNeXt ResNet 50	0.874	0.816	0.037	2023
SINetV2 Res2Net 50	0.847	0.770	0.048	2021

3.2 Proposed Methodology for QKD

This section is divided into four subsections: QBER Significance, Data Generation, Model Selection, and Performance Evaluation.

3.2.1 QBER Significance

QBER tells us the amount of bits in the quantum key that were not correct. It is important to understand QBER if you want to detect and correct errors. Also, when the error number rises, it may suggest someone is listening in. There are a variety of sources that create errors in QKD. Such threats consist of disturbances from channels, changes in the environment, faults with detectors and unlawful eavesdropping. Signal loss and a broadened wavelength are both causes of channel noise. Environmental change focuses on upheavals due to temperature swings and vibrations. Dark counts and problems with timing precision are present in detector problems. The disturbances in eavesdropping are intentional. All of this adds more mistakes to the process. When QBER is high, the safety and convenience of sharing the key call for stronger error correction approaches to secure the final key that's used. For example, when the QBER in BB84 is above 11%, we should expect that the protocol is compromised and key distillation cannot proceed safely.

$$QBER = \frac{N_{\text{error}}}{N_{\text{total}}} \tag{3.1}$$

where N_{error} is the number of incorrect bits detected in the sifted key and N_{total} is the total number of transmitted key bits.

The Cascade Protocol is widely used in quantum key distribution because it corrects errors with successful and repetitive parity cheques [44]. Even though Cascade is effective, newer methods such as LDPC codes are faster and more flexible which becomes apparent in high speed scenarios. As a result, traditional methods for reconciliation like Cascade are not practical for fast quantum communication, since they require many calculations [45]. With Neural Networks, it is possible to make good estimates of QBER and enhance

error correction, not requiring several repetitions. If deep neural networks, especially feedforward or convolutional types, are used, the model can detect complex patterns in bit error records, so it can provide real time estimates and determine suitable solutions for correcting errors.

3.2.2 Data Generation

Since we couldn't find any openly available quantum key distribution data, we built our own dataset using Qiskit, an open source framework provided by IBM. The reason for scarcity is that quantum communication systems are easily influenced and collecting data in experiments is very complicated. This means simulations are a useful way to get many different, replicable training cases without disturbing the real world. With Qiskit, we can act on quantum-based privacy schemes and test circuits which helps us obtain a dataset that matches our needs. Running the BB84 protocol through Qiskit's circuits made it possible to see a basic but useful version of how qubits are transmitted and measured in a noisy environment. We examined quantum key distribution using 120,000 samples, varying both the transmission rates and the noise probabilities within the ranges pointed out above. We have 7 different attributes in this dataset. A heatmap was also produced to visualise relations between inputs which confirmed that Noise Probability and Final Key Length have a strong negative association, supporting the focus of our model on these factors. You can see this in Fig. 3.3

These attributes are Trial Number to uniquely identify a QKD simulation instance, Photon Transmission Rate (in kbps) to represent the data transmission speed in quantum communication, Noise Probability to define the likelihood of an error occurring in the quantum channel, Quantum Bit Error Rate (QBER) is the primary error metric used for modelling training, Shifted Key Length is the number of key nits remaining after basis reconciliation, SNR represents the quality of the received quantum signal, defined as:

$$SNR = \frac{P_{\text{signal}}}{P_{\text{noise}}} \tag{3.2}$$

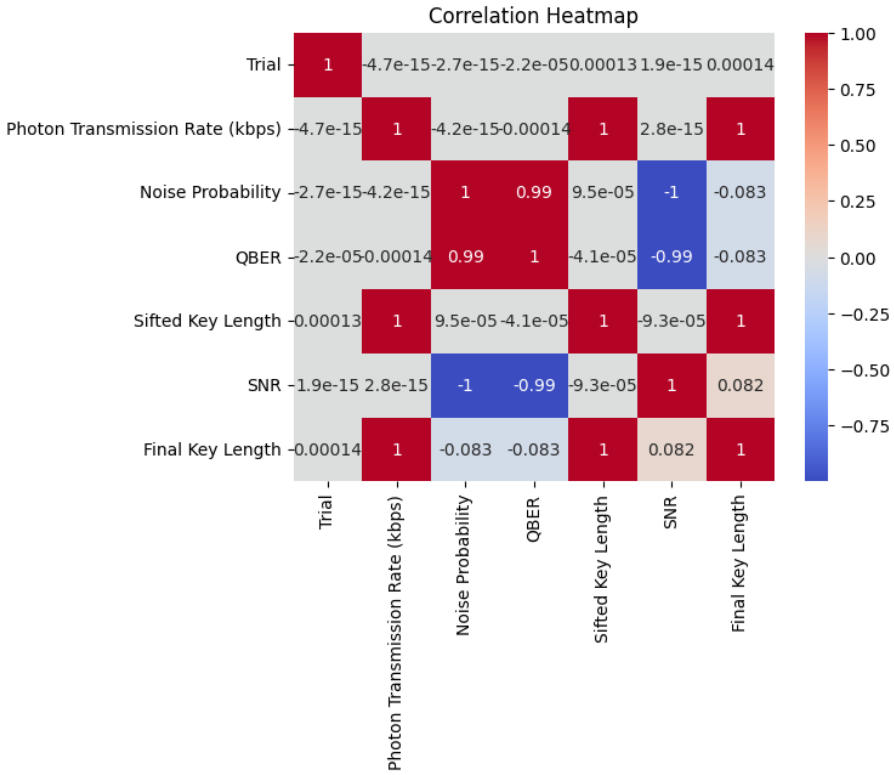

Figure 3.3: Correlation Heatmap

where P_{signal} is the power of the transmitted quantum signal, P_{noise} is the noise power in the channel, and Final Key Length is the length of the corrected key after error correction. By generating a wide range of QKD conditions the dataset enables the training of ML models that can generalise well across different quantum communication cases.

3.2.3 Model Selection

We examined and compared several neural network models, including MLP, LSTM, GRU, CNN and Autoencoders, to improve QKD error correction. The models were programmed to recognise patterns in QBER variations and boost the precision of crucial reconciliation. It was decided to use neural networks since they display a higher ability to deal with complex and non linear information in high-dimensional quantum key distribution data and are able to successfully generalise in a wide variety of test cases.

A Multi Layer Perceptron (MLP) is a feed forward Neural Network that consists of multiple layers of neurons applying an activation function to its weighted inputs before

passing the output to the next layer [46]. The transformation is defined as:

$$y = f(Wx + b) \tag{3.3}$$

where W is the weight matrix, x is the input vector, b is the bias term, and f is the activation function.

These networks work well on patterns of data in order, so they are useful for looking at QBER trends over several experiments [47]. Runs of QKD simulation involved at least 100 sequential trials, so the QBER results depended on previous observations. Therefore, models such as LSTM which are designed to work with time sequences, fitted the task perfectly.

$$f_t = \sigma(W_f \cdot [h_{t1}, x_t] + b_f) \tag{3.4}$$

$$i_t = \sigma(W_i \cdot [h_{t1}, x_t] + b_i) \tag{3.5}$$

$$o_t = \sigma(W_o \cdot [h_{t1}, x_t] + b_o) \tag{3.6}$$

where f_t, i_t, o_t are the forget, input, and output gate activations, σ represents the sigmoid activation function, h_{t1} is the hidden state from the previous time step, and x_t is the input at time step t [47].

GRUs were designed to fix the vanishing gradient problem that plagues standard recurrent neural networks and even though they are simpler than LSTMs, they still capture long-term features in data and use less computation than LSTMs [48].

GRUs make use of two gates, an *update gate* and a *reset gate*. The update gate chooses the amount of data from the previous timestep that will be used in the next step and the reset gate helps choose the amount that should be forgotten [48]. Take a look at how mathematical operations work for a GRU cell:

$$z_t = \sigma(W_z \cdot [h_{t1}, x_t]) \tag{3.7}$$

$$r_t = \sigma(W_r \cdot [h_{t1}, x_t]) \tag{3.8}$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t1}, x_t]) \tag{3.9}$$

$$h_t = (1z_t) * h_{t1} + z_t * \tilde{h}_t \tag{3.10}$$

Here, z_t and r_t are the update and reset gates respectively, h_t is the hidden state at time t, and x_t is the input at time t. The σ function denotes the sigmoid activation, and tanh is the hyperbolic tangent function.

Our investigation found that the GRU worked fairly well with handling time-oriented information, but did not perform as well as CNN and Autoencoder models. This can be explained by our dataset having limited complexity, so advanced temporal models were not required.

Spatial features are found in QKD data using CNN [49]. The features from the QKD input were restructured to fit the CNN, so it could learn about the connexions among noise, SNR and photon rate at different points in space. They philtre input data with convolution, making use of kernels.

$$y = \sum_{i=1}^{n} x_i w_i + b \tag{3.11}$$

where x_i represents the input values, w_i are the convolutional kernel weights, and b is the bias term.

Autocoders help reduce the size of QKD data so that corrections can be made more easily [50]. The reduced dimensions and cleaned high variance aspect in the encoder's latent space helped the decoder to provide important information for later classification and regression.

$$h = f(Wx + b) \tag{3.12}$$

and the reconstruction process is:

$$\hat{x} = g(W'h + b') \tag{3.13}$$

In this case, h is the latent representation, both f and g are activation functions and W, W', b, and b' are weight and bias matrices. A full assessment of the error correction abilities of the architectures was made by calculating MAE, RMSE, RMSLE and Final Key Accuracy, using them as key metrics.

3.2.4 Performance Evaluation

We evaluated all our models by looking at a set of performance metrics such as: The Mean Absolute Error measures on average the gap between the predicted value and the actual measure. The error in a time series prediction is often measured by it. It has the form:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i\hat{y}_i| \tag{3.14}$$

where y_i is the actual value and $\hat{y}_i$ is the predicted value. MAE provides a direct interpretation of the average magnitude of error, making it especially useful for understanding real world deviations.

RMS Error, abbreviated as RMSE, measures the daily variation between regression results and actual data. As RMSE responds to outliers, it performs well where important mistakes should not occur such as in safe quantum communication. When you notice a major change in the length of the secret code or its QBER, it might be sign of a weakness. Our results indicated that smaller MSE implied the model predictions were almost identical to the true figures, with few extreme outliers. It was evident that in general, the data was free of big common mistakes. It is shown as :

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i\hat{y}_i)^2} \tag{3.15}$$

Root Mean Squared Logarithmic Error (RMSLE) is an evaluation metric commonly used in regression problems, particularly when the target variable has a wide range of

values. It is given as:

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(1+y_i)\log(1+\hat{y}_i))^2} \tag{3.16}$$

RMSLE penalises relative differences by taking their logarithms, different to the harsher treatment of large differences seen in RMSE. As a result, when math is used for variables that can have either a tiny or enormous value, percentages are given a higher priority than absolute errors.

When used in QKD key prediction, RMSLE shows how precisely the model tracks changes in key length as they become either small or vary over various ranges. Our results suggest that the accuracy of all models was high and sustained over different sizes of sample data.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100 \tag{3.17}$$

In the way we used it, even though accuracy is usually for classification, we used it to see how many times the model predicted the exact Final Key Length within a specified acceptable margin. A high accuracy of 99.897% in our CNN model proves that it can often make accurate predictions and is suited for real world QKD activities.

Chapter 4

EXPERIMENTAL SETUP

4.1 NC4K

COD researchers consider the Natural Camouflaged 4K dataset, also called NC4K dataset, a valuable resource. A large variety of 4,121 high-resolution images taken from the internet make up this new dataset. Many of these pictures present several natural locales like forests, open pasture, the sea, mountains and deserts, each with numerous well-hidden animals, sea animals, insects and man-made objects. A simple example appears in Fig. 4.1. This is shown in Fig. 4.2.

In order to support further research and development on COD, the images in this dataset are grouped into three categories: camouflaged objects, backgrounds and non-camouflaged objects. The camouflage objects are clearly labelled in each image on the website. The annotation set also provides segmented masks for all images.

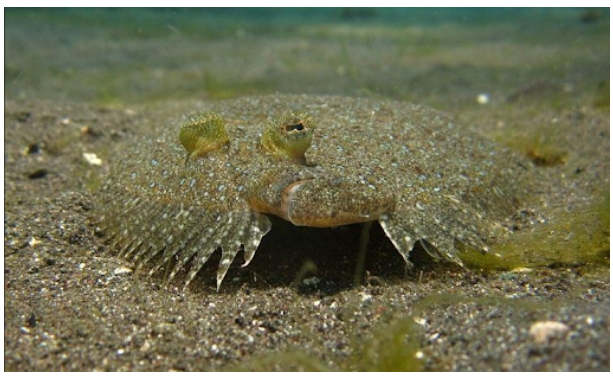

Figure 4.1: A picture of a fish from the NC4K dataset

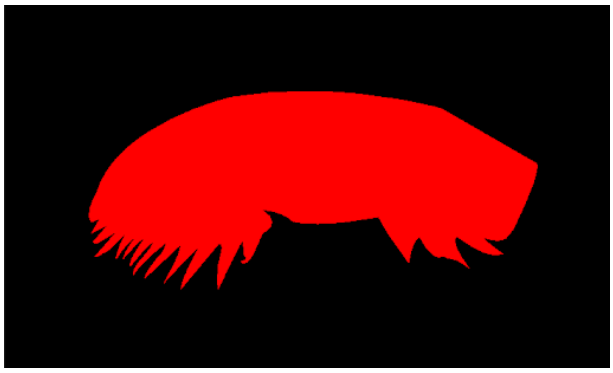

Figure 4.2: Image of an instance of a fish from the NC4K dataset

4.2 EfficientDet

EfficientDet is a model developed by Google Research for object detection. It is one of the most efficient models, as it prioritizes a balance between achieving high accuracy in object detection and maintaining computational efficiency. Hence, this model is suitable for deployment on platforms that offer limited resources.

EfficientDet is separate from its original version, EfficientNet [51]. To make CNNs more effective at larger scales, EfficientNet proposed compound scaling that scales the resolution, depth and width together. EfficientDet is mainly recognised for two key aspects, compound scaling and BiFPN. All components of the model's architecture such as resolution, depth and width, are increased step by step in proportion to the dataset as the model is trained. This helps every model part impact the model's end performance in an effective way. BiFPN which stands for Bi-Directional Feature Pyramid Network, is a new part of the model that assists with better feature extraction. It supports the exchange of messages in both from top to bottom and from bottom to top within the network. As a result, EfficientDet can recognise a larger range of features that matter for accurate object detection. Multi-scale features are traditionally gathered by the FPN in a process starting from top and moving to the bottom.

$$P_7^{out} = Conv(P_7^{in}) \tag{4.1}$$

$$P_6^{out} = Conv(P_6^{in} + Resize(P_7^{out}))... \tag{4.2}$$

$$P_3^{out} = Conv(P_3^{in} + Resize(P_4^{out})) \tag{4.3}$$

The equation (1), (2), and (3) were presented by Tan et al. in their paper [**?**].

4.3 SAM

Segment Anything Model or SAM is designed to find and group objects in pictures. Meta AI brought it out in 2023. The usual approach to segmentation required lots of data for every object, but SAM does not need any data for what it is segmenting. SAM is most valuable due to this feature. In this way, SAM recognises objects in an image without first needing to learn about them. This means that tasks dealing with hidden or low occurrence categories are much easier. See Fig. 4.3 to see it.

Thanks to SAM, we have the option to train segmentation using our own prompts and language. As a result, this feature enables users to adjust the segmentation process more manually.

Those are the main components found in SAM: the image encoder, prompt encoder and mask decoder. The image encoder explores the input image and obtains different features. It uses a pre-programmed Vision Transformer (ViT) to accomplish this. ViTs can recognise and represent the complex connexions present in images. Prompt Encoder processes different prompt types including text descriptions, bounding boxes and segmentation masks to learn what segmentation is desired. Based on information gathered from the image features and prompts, Mask Decoder produces a segmentation mask for each item shown in the picture.

4.4 Qisbit

QNu Labs built Qisbit, a quantum key distribution simulator, so that users can use it easily to simulate QKD protocols. Since users can change noisy levels, approaches to matching the basics, error correction and privacy amplification, the simulator gives a good platform for learning about real-world problems in QKD. This paper benefited from

Figure 4.3: SAM being used for masking an image

Qisbit-generated data that simulated what QKD operations would be like in practise. Key elements of the datasets included raw input keys, how many errors were found and secure key length which allowed for training and validation of deep learning in QKD error correction.

Using Qisbit's tools, we were able to simulate the kinds of situations often seen in real QKD systems. Because of this, the model had access to a broad variety of error samples and encryption results which enhanced its adaptability in various conditions. This work benefited greatly from Qisbit, as it made it practical to move between lab theories and real engineered QKD systems.

4.5 Google Colab

Both the Camouflaged Object Detection (COD) and Quantum Key Distribution (QKD) error prediction research use Google Colab as their main tool for experimenting. Since we integrated GPU and TPU, training and inference of deep learning models have become much faster and easier. Thanks to Colab's shared environment and cloud storage, we could make quick changes, follow different versions of models and test multiple kinds of models.

Researchers used Colab to handle the data generated with Qisbit, train the neural networks and deep learning models and cheque how well the system worked. Just like in YCB-video, Colab delivered the computation required to adjust the EfficientDet and

SAM model, run tests on NC4K and view the results. Because it works with Python libraries such as TensorFlow, PyTorch, OpenCV and scikit learn, developers found it easy to finish end to end work for both projects.

Chapter 5

RESULTS AND DISCUSSION

In this chapter we analyse the experimental outcomes obtained from the two main part of our studies that are Camouflaged Object Detection (COD) and Quantum Key Distribution (QKD) error correction. We compared the performance of the proposed models using standard metrics and discuss the importance of the results in relation to existing approaches. For each component both quantitative and qualitative analyses are included to provide a better view of the model behavior, strengths, and limitations.

We have analysed the results using accuracy, error distribution, inference efficiency and how adaptable the models were. Next, we examined what limits there are and how further work might progress. The purpose of this study is to explain the model selection and suggest areas where results could be improved in practise.

5.1 Overall Outcome for NC4K Dataset

The approach we use reached promising performance when detecting camouflaged items in the NC4K data. By using the dice coefficient to compare the predicted masks to the actual ones, we measured an average of 87.87% similarity between them. The strong result shows that our method separates camouflaged objects from the background successfully. On the other hand, IoU which cheques how much predicted and ground truth masks have in common relative to their whole, provided an average score of 81.18%. Even though this shows effective segmentation, it also points out a chance to improve detecting the

edges of hidden objects. The findings are shown in Fig. 5.1. An elevated Dice coefficient reflects the model's successful manner of segmenting different objects. The low IoU shows that working on the model will help it better determine the delicate edges of camouflaged things.

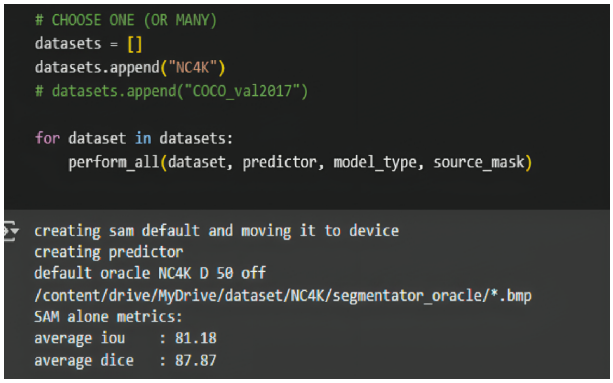

Figure 5.1: Results of EfficientDet with SAM on NC4K

5.2 Overall Outcome for Quantum Key Prediction

5.2.1 Different Measurement Metrics

Five different deep-learning models were used to cheque which one provides the most effective error correction in QKD: Autoencoders, CNN, GRU, LSTM and MLP. The results for each model on the test dataset are shown in Table 3.1.

Table 5.1: Performance comparison of models over different evaluation metrics

Model	MAE	RMSE	RMSLE	Accuracy
MLP	23.236	28.594	0.014	29.812%
LSTM	5.735	7.580	0.012	85.137%
GRU	28.661	38.454	0.019	26.616%
CNN	**1.571**	**2.111**	**0.002**	**99.897%**
Autoencoder	3.309	4.066	0.002	99.334%

The CNN model done best, with an accuracy of 99.897%. It could be because CNN has the skill to notice local variations in the data and their connexions across space, just like the arrangement of errors in QKD devices. The model also performed very well, reaching 99.334% accuracy and only having a small low value for MAE (3.309) and RSME

(4.066). The model compresses input data and then reconstructs the original data from its compressed version. This ability allows the system to remove irrelevant noise from useful signals. It works well when a QKD system is surrounded by noise. The LSTM model was able to predict data accurately. It showed accuracy of 85.137%. It further shows that LSTM is able to learn how events are connected over different time frames. These systems let us use their main advantage, as errors will keep changing over time during their operations. The combination of RNN models did not do as well as expected; MLP came in second with 29.812 and the worst result came from GRU at 29.616, showing that these models were challenged by the complexity of QKD errors.

Such stability means they can be used in QKD systems, as consistent behaviour is essential in applications that use them promptly. LSTM training showed that the validation loss sometimes fluctuated around a normal figure which suggests it was only mildly affected by hyperparameters and learned steadily most of the time. Convergence was an issue for both MLP and GRU because their high validation loss hampered them.

5.2.2 Error Distribution

We have generated graphs that visualise the error in key length for all the working models. The Error Distribution for MPL is given in Fig. 5.2, while Fig. 5.3 displays it for LSTM. When measured against other models, GRU showed the least success and its Error Distribution is seen in Fig. 5.4. In this case, CNN and Autoencoders worked well and the distribution of their errors is displayed in Fig. 5.5 and Fig. 5.6. The errors of CNN and Autoencoder are mainly around the middle or zero which means both models are accurate. It seems that performance of the LSTM model varies slightly due to the wider error spread. Having so few erroneous keys makes the model's behaviour consistent which is significant for the security of QKD processing pipelines. Because both models have a high number of large and scattered errors, MAE and RMSE are noticeably higher for them. We found that our plot of predicted and actual key length revealed the same outcomes.

To decide if these models can be used in real-time QKD systems, we looked at how fast

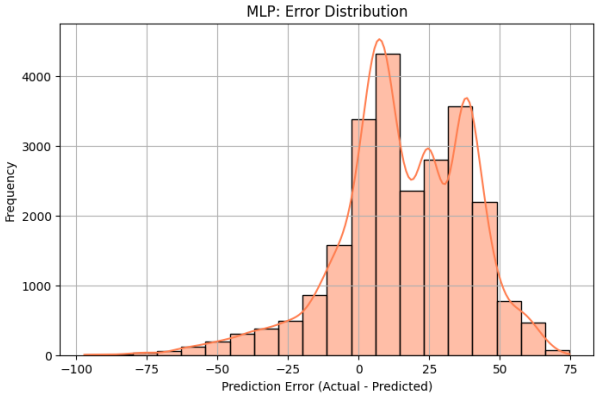

Figure 5.2: Error Distribution chart of MLP

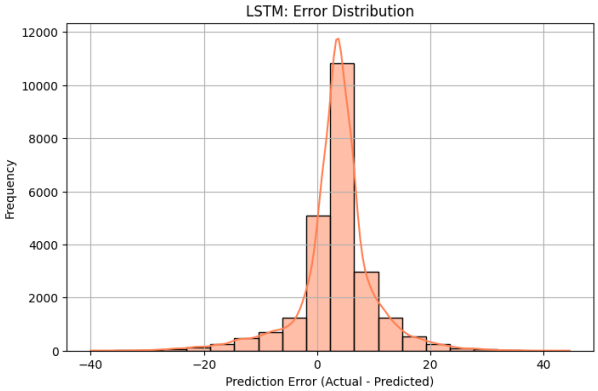

Figure 5.3: Error Distribution chart of LSTM

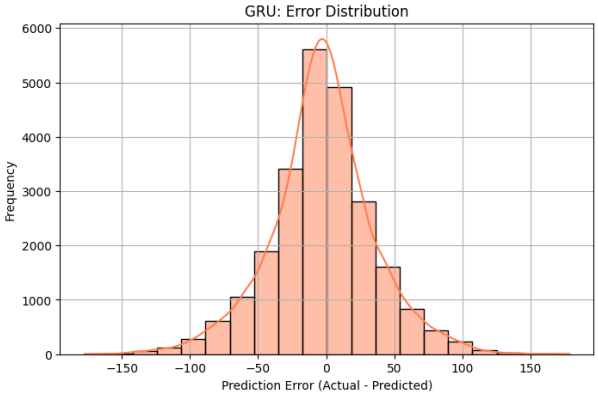

Figure 5.4: Error Distribution chart of GRU

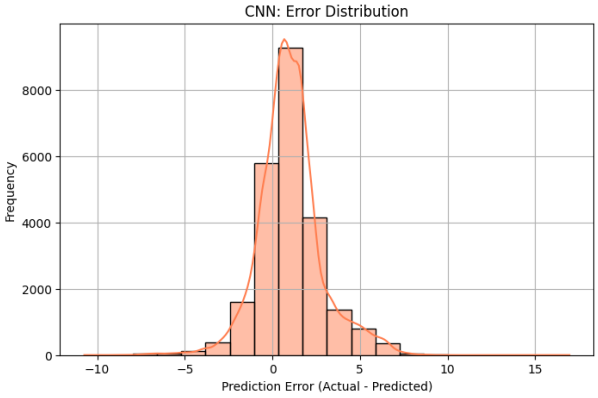

Figure 5.5: Error Distribution chart of CNN

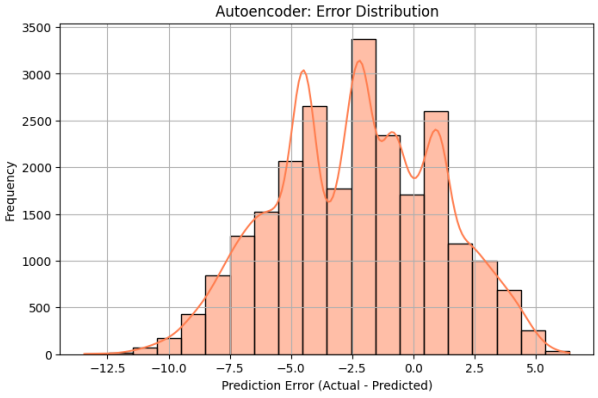

Figure 5.6: Error Distribution chart of Autoencoders

they trained and how quickly they made calculations. Due to how fast inference happens in CNN and Autoencoder, they are an appealing choice for quantum communication using limited resources. The computational efficiency for every architecture is listed in Table 5.2.

Table 5.2: Computational performance of models

Model	**Training Time (min)**	**Inference Speed (ms/sample)**
MLP	5.4	0.09
LSTM	12.7	0.15
GRU	10.5	0.12
CNN	8.1	0.08
Autoencoder	9.3	0.11

This model ran the fastest and was the most accurate out of all the other models tested. This approach showed excellent performance, succeeding in attaining both accuracy and a reasonable amount of computation. Because of their larger architecture, LSTM models usually take the longest time to train among the models examined.

5.3 Limitation

Both studies have their separate limitations and we have discussed them in this section. This section shows the limitations of the research setup and knowledge in the presented domain.

The approach we suggested produced good outcomes, yet there are still challenges that should be addressed. Originally, the biggest concern was the huge size of the NC4K dataset. Even if the NC4K dataset is large, there are other, even larger datasets available. There are other datasets such as COD10K, that exceed the size of the NC4K dataset. Furthermore, the performance of the model may be connected to how diverse the objects in the NC4K data are and how high the image resolutions are. We should point out that the findings from this study do not fully apply to using deep learning on videos or live material. If these issues are overcome in future research, COD systems will become both more powerful and better able to adapt.

Even though the results are very good, we must recognise several limitations. Instead

of using QKD hardware to obtain the data, we generated our dataset using Qiskit simulations. Still, simulations act as a first phase for prototyping, helping you practise and test models before putting them into operation. This type of dataset gives an optimised space for testing simulations, but it still leaves out details like poor hardware, outside disturbances or attacks from ill-wishers. In optical QKD, problems such as phase drift and misalignment of polarisation can have a strong effect on how fast keys are produced. In the future, the models should be tested on actual QKD systems to cheque how robust they are in this type of experiment. Even so, it is not clear how well these models adapt in real time to new and changing levels of quantum noise in the circuit. In future, experiments might use online learning methods or adjustable philtres to account for changing levels of noise in the environment.

Due to the nature of training LSTMs and GRUs, our experiments have become more time-consuming which also involves higher computational costs than using CNNs and Autoencoders. Despite this, if precision of the predictions is more important than how fast results are delivered such models could work well. We can make further improvements using pruning or quantization techniques to ensure our quantum model runs well on devices with limited resources and lowers the time needed for inferences. Furthermore, although forecasting Final Key Length and QBER were the main research points, key research topics such as channel stability, identifying eavesdroppers and managing the process of key reconciliation were not considered. If multi-objective learning is used, models can maximise performance in security and communication metrics together. Reinforcement learning does this by setting proactive protocol parameters quickly, whereas adversarial training helps models spot or resist when an attack happens.

Chapter 6

CONCLUSION AND FUTURE SCOPE

This chapter wraps up by describing the main points and findings developed in the two main research paths discussed in this work: studying Camouflaged Object Detection (COD) with deep learning and QKD error correction through machine learning. They have different uses, but they share the same purpose of overcoming big, important issues with AI methods such as spotting objects or hackproofing quantum communication. Using specialised datasets, matching models and strict evaluation tools, this research has shown that deep learning is useful for handling real-world problems.

By collaborating across disciplines, we have looked into how EfficientDet, SAM, CNNs and Autoencoders can be adjusted and used to perform well in particular cases. We have found that, when combined with specific requirements, AI can truly drive important changes. In the following portions, we describe essential findings from both COD and QKD optimisation works and suggest valuable ideas for upcoming AI projects in practical settings.

6.1 Overall Conclusion

For our COD method, we explained why EfficientDet and SAM perform well and together outperform other models. By using this technique, the team obtained notable outcomes

on the NC4K dataset and proved that it detects camouflaged objects effectively. The Dice metric proved segmentation accuracy; however, IoU suggests the model could improve with boundaries of objects. Work on the model will next address fine margins and investigate additional approaches like better data augmentation and new type of model structures. Therefore, we plan to push the limits of camouflaged object detection by looking into the suggested methods and comparing ourselves to current best practises.

Our study in the second paper shows that deep learning greatly reduces the error rate of QKD by precisely estimating the Final Key Length and increasing the QBER. As a result, errors are managed more efficiently and the secure key is preserved which is crucial for stable quantum communication. The model achieved almost perfect accuracy and had few prediction errors. Because it is both computationally inexpensive and fast to use, this approach is suitable for live use in QKD protocols. The Autoencoder architecture showed excellent results, so it became a good candidate for optimising QKD errors. Thanks to its unsupervised learning, it is able to learn complex QKD error behaviours, even in fast-changing environments without needing labelled inputs. Of these models, LSTM performed the best and was not far from average accuracy. LSTM offer an average level of accuracy, but we can apply them when the data or channel depends on time or sequence.

MLP and GRU models are less compatible for this case as they were not able to generalize QKD error patterns. There is a scope of improving these models by doing architectural tuning or hybrid models. These results shows the potential of AI driven solution in the field of QKD efficiency, reducing computational overhead, and improving real life scalability. For future work we could check reinforcement learning or hybrid quantum classical models to further improve adaptability and performance in evolving QKD systems.

As a result, we can use part of this research in fields such as computer vision and quantum cryptography to obtain results and guide our future work. The COD task indicated that designing the architecture to cover various visual features improved detecting ambiguous scenes. According to the QKD work, neural networks are good at identifying patterns found in encryption errors and using them to create secure keys. Both approaches

highlight that data-driven methods are flexible and successful in handling difficult and precise challenges. Investigating AI in these ways increases research on AI as well as helps promote interdisciplinary approaches to current technological problems.

6.2 Future Scope

We have achieved excellent results with the method we propose in the first paper, but there is still more that can be improved and worked on in the future. It's important to improve how well the IoU evaluates results. As a result, we could enhance the model structure or set the best hyperparameter values to boost the quality of detecting boundaries. A range of experiments and specially created architectures may help improve performance further. We aim to promote progress in COD and update detection tools applied in practical situations by studying these future topics.

Our schedule now includes validating the trained models on real QKD hardware to see their performance in real-time uses. Partnering with places that have QKD testbeds such as quantum labs or institutions, could help reduce the gap between what is simulated and what is actually performed. Extra optimization of CNN and Autoencoder architectures is needed to make our work suitable for use in real-time quantum communication systems. The integration of reinforcement learning can allow QKD protocols to modify their settings on the fly to suit changes in the environment's noise level which improves how QKD protocols adapt. Tweaking the basis reconciliation threshold or privacy amplification ratio in real time could greatly improve the amount of secure keys generated.

In addition, I will examine aspects of security by introducing attack aware ML models to consider adversarial attacks and eavesdropping detection. Adversarial training and techniques using Generative Adversarial Networks (GANs) are part of this effort. In order to enhance AI driven QKD optimisation techniques, we can increase the size of our dataset to cover different conditions found in quantum networks such as QKD with satellites and fibre optics. The use of these setups created new obstructive factors such as high delays and issues caused by nearby weather. This makes it necessary for ML models to figure out ways to cope with these problems.

For future, an approach that can draw insights across both studies could open new research directions. For instance, techniques which were used to improve boundaries precision in COD such as attention mechanisms or transformer based models may enhance QKD model understandability or robustness under noisy conditions. The success of lightweights models and unsupervised architectures in QKD optimisation could inspire more efficient COD pipelines. There is also scope for exploring hybrid applications where quantum-secured communication supports AI based surveillance or remote detection tasks. Linking such cross domain applications represents a promising perimeters for future related research.

References

[1] H. A. Al-Mohammed and E. Yaacoub, "On the use of quantum communications for securing iot devices in the 6g era," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6.

[2] R. Wolf and R. Wolf, "Recent developments in practical qkd," *Quantum Key Distribution: An Introduction with Exercises*, pp. 183–217, 2021.

[3] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, "Quantum entanglement," *Reviews of modern physics*, vol. 81, no. 2, pp. 865–942, 2009.

[4] K. Bartkiewicz, K. Lemr, A. Černoch, J. Soubusta, and A. Miranowicz, "Experimental eavesdropping based on optimal quantum cloning," *Physical Review Letters*, vol. 110, no. 17, p. 173601, 2013.

[5] D. Tupkary and N. L¨utkenhaus, "Using cascade in quantum key distribution," *Physical Review Applied*, vol. 20, no. 6, p. 064040, 2023.

[6] T. Chapuran, P. Toliver, N. Peters, J. Jackel, M. Goodman, R. Runser, S. McNown, N. Dallmann, R. Hughes, K. McCabe *et al.*, "Optical networking for quantum key distribution and quantum communications," *New Journal of Physics*, vol. 11, no. 10, p. 105001, 2009.

[7] L.-J. Wang, K.-H. Zou, W. Sun, Y. Mao, Y.-X. Zhu, H.-L. Yin, Q. Chen, Y. Zhao, F. Zhang, T.-Y. Chen *et al.*, "Long-distance copropagation of quantum key distribution and terabit classical optical data channels," *Physical Review A*, vol. 95, no. 1, p. 012301, 2017.

[8] Q. Zhu, X. Yu, Y. Zhao, A. Nag, and J. Zhang, "Resource allocation in quantum-key-distribution-secured datacenter networks with cloud–edge collaboration," *IEEE Internet of Things Journal*, vol. 10, no. 12, pp. 10 916–10 932, 2023.

[9] H. A. Al-Mohammed, S. Al-Kuwari, H. Kuniyil, and A. Farouk, "Towards scalable quantum key distribution: A machine learning-based cascade protocol approach," *arXiv preprint arXiv:2409.08038*, 2024.

[10] H.-K. Mao, Q. Li, P.-L. Hao, B. Abd-El-Atty, and A. M. Iliyasu, "High performance reconciliation for practical quantum key distribution systems," *Optical and Quantum Electronics*, vol. 54, no. 3, p. 163, 2022.

[11] D.-P. Fan, G.-P. Ji, T. Zhou, G. Chen, H. Fu, J. Shen, and L. Shao, "Pranet: Parallel reverse attention network for polyp segmentation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2020, pp. 263–273.

[12] W. Liu, X. Shen, C.-M. Pun, and X. Cun, "Explicit visual prompting for universal foreground segmentations," *arXiv preprint arXiv:2305.18476*, 2023.

[13] Y. Lv, J. Zhang, Y. Dai, A. Li, B. Liu, N. Barnes, and D.-P. Fan, "Simultaneously localize, segment, and rank the camouflaged objects," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 591–11 601.

[14] Y. Pang, X. Zhao, T.-Z. Xiang, L. Zhang, and H. Lu, "Zoomnext: A unified collaborative pyramid network for camouflaged object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[15] A. Walton, A. Ghesquière, and B. Varcoe, "Experimental thermal state qkd dataset," 2024.

[16] R. J. Collins, R. Amiri, M. Fujiwara, T. Honjo, K. Shimizu, K. Tamaki, M. Takeoka, M. Sasaki, E. Andersson, and G. S. Buller, "Experimental demonstration of quantum digital signatures over 43 db channel loss using differential phase shift quantum key distribution," *Scientific reports*, vol. 7, no. 1, p. 3235, 2017.

[17] M. Polnik, L. Mazzarella, M. Di Carlo, D. K. Oi, A. Riccardi, and A. Arulselvan, "Scheduling of space to ground quantum key distribution," *EPJ Quantum Technology*, vol. 7, no. 1, p. 3, 2020.

[18] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross *et al.*, "Quantum computing with qiskit," *arXiv preprint arXiv:2405.08810*, 2024.

[19] T.-N. Le, T. V. Nguyen, Z. Nie, M.-T. Tran, and A. Sugimoto, "Anabranch network for camouflaged object segmentation," *Computer vision and image understanding*, vol. 184, pp. 45–56, 2019.

[20] D.-P. Fan, G.-P. Ji, M.-M. Cheng, and L. Shao, "Concealed object detection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 44, no. 10, pp. 6024–6042, 2021.

[21] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "Basnet: Boundary-aware salient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7479–7489.

[22] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 781–10 790.

[23] D.-P. Fan, G.-P. Ji, M.-M. Cheng, and L. Shao, "Concealed object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6024–6042, 2022.

[24] T.-N. Le, Y. Cao, T.-C. Nguyen, M.-Q. Le, K.-D. Nguyen, T.-T. Do, M.-T. Tran, and T. V. Nguyen, "Camouflaged instance segmentation in-the-wild: Dataset, method, and benchmark suite," *IEEE Transactions on Image Processing*, vol. 31, pp. 287–300, 2021.

[25] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[26] T.-N. Le, Y. Cao, T.-C. Nguyen, M.-Q. Le, K.-D. Nguyen, T.-T. Do, M.-T. Tran, and T. V. Nguyen, "Camouflaged instance segmentation in-the-wild: Dataset, method, and benchmark suite," *IEEE Transactions on Image Processing*, vol. 31, pp. 287–300, 2022.

[27] P. Zheng, D. Gao, D.-P. Fan, L. Liu, J. Laaksonen, W. Ouyang, and N. Sebe, "Bilateral reference for high-resolution dichotomous image segmentation," *arXiv preprint arXiv:2401.03407*, 2024.

[28] A. K. Ekert, "Quantum cryptography based on bell's theorem," *Physical review letters*, vol. 67, no. 6, p. 661, 1991.

[29] R. Chatterjee, K. Joarder, S. Chatterjee, B. C. Sanders, and U. Sinha, "qkdsim: An experimenter's simulation toolkit for qkd with imperfections, and its performance analysis with a demonstration of the b92 protocol using heralded photon," *arXiv preprint arXiv:1912.10061*, 2019.

[30] ——, "qkdsim, a simulation toolkit for quantum key distribution including imperfections: performance analysis and demonstration of the b92 protocol using heralded photons," *Physical Review Applied*, vol. 14, no. 2, p. 024036, 2020.

[31] K. W. Hong, O.-M. Foong, and T. J. Low, "Challenges in quantum key distribution: A review," in *Proceedings of the 4th International Conference on Information and Network Security*, 2016, pp. 29–33.

[32] R. Gallager, "Low-density parity-check codes," *IRE Transactions on information theory*, vol. 8, no. 1, pp. 21–28, 1962.

[33] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," in *Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1993, pp. 410–423.

[34] H.-J. Ding, J.-Y. Liu, C.-M. Zhang, and Q. Wang, "Predicting optimal parameters with random forest for quantum key distribution," *Quantum Information Processing*, vol. 19, pp. 1–8, 2020.

[35] F.-Y. Lu, Z.-Q. Yin, C. Wang, C.-H. Cui, J. Teng, S. Wang, W. Chen, W. Huang, B.-J. Xu, G.-C. Guo *et al.*, "Parameter optimization and real-time calibration of a measurement-device-independent quantum key distribution network based on a back propagation artificial neural network," *JOSA B*, vol. 36, no. 3, pp. B92–B98, 2019.

[36] D. Huang, P. Huang, D. Lin, and G. Zeng, "Long-distance continuous-variable quantum key distribution by controlling excess noise," *Scientific reports*, vol. 6, no. 1, p. 19201, 2016.

[37] W. Wang and H.-K. Lo, "Machine learning for optimal parameter prediction in quantum key distribution," *Physical Review A*, vol. 100, no. 6, p. 062334, 2019.

[38] V. Dunjko and H. J. Briegel, "Machine learning & artificial intelligence in the quantum domain: a review of recent progress," *Reports on Progress in Physics*, vol. 81, no. 7, p. 074001, 2018.

[39] Z.-P. Liu, M.-G. Zhou, W.-B. Liu, C.-L. Li, J. Gu, H.-L. Yin, and Z.-B. Chen, "Automated machine learning for secure key rate in discrete-modulated continuous-variable quantum key distribution," *Optics Express*, vol. 30, no. 9, pp. 15 024–15 036, 2022.

[40] Y. Mao, W. Huang, H. Zhong, Y. Wang, H. Qin, Y. Guo, and D. Huang, "Detecting quantum attacks: A machine learning based defense strategy for practical continuous-variable quantum key distribution," *New Journal of Physics*, vol. 22, no. 8, p. 083073, 2020.

[41] J. Li, Y. Guo, X. Wang, C. Xie, L. Zhang, and D. Huang, "Discrete-modulated continuous-variable quantum key distribution with a machine-learning-based detector," *Optical Engineering*, vol. 57, no. 6, pp. 066 109–066 109, 2018.

[42] S. Iqbal, A. N. Qureshi, J. Li, and T. Mahmood, "On the analyses of medical images using traditional machine learning techniques and convolutional neural networks," *Archives of Computational Methods in Engineering*, vol. 30, no. 5, pp. 3173–3233, 2023.

[43] S. Niu, X. Zhou, D. Zhou, Z. Yang, H. Liang, and H. Su, "Fault detection in power distribution networks based on comprehensive-yolov5," *Sensors*, vol. 23, no. 14, p. 6410, 2023.

[44] F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan, "Secure quantum key distribution with realistic devices," *Reviews of modern physics*, vol. 92, no. 2, p. 025002, 2020.

[45] A. Choudhary and A. Wasan, "Cracking the curious case of the cascade protocol," *IEEE Access*, 2023.

[46] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Transactions on Circuits and Systems*, vol. 8, no. 7, pp. 579–588, 2009.

[47] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[48] K. Cho, B. Van Merri¨enboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[49] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[50] D. E. Rumelhart, J. L. McClelland, P. R. Group *et al.*, *Parallel distributed processing, volume 1: Explorations in the microstructure of cognition: Foundations*. The MIT press, 1986.

[51] D.-P. Fan, G.-P. Ji, M.-M. Cheng, and L. Shao, "Concealed object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.