4th Year Project Research Document

ImageAware+ - Phishing Detection & Educational Platform

Name	**Lorcan Kelly Zazzera**
Student ID	**C00288941**
Course	**Bachelor of Science (Honours) in Cybercrime and IT Security**
Supervisor	**Mark Cummins**

1.0 Introduction

1.1 Background

Phishing is still one of the most common and harmful types of cybercrime, and it keeps getting more complex and widespread even though people have been trying to stop it for decades. IBM Security's Cost of a Data Breach Report 2024 says that phishing was responsible for about 36% of all data breaches around the world, with an average cost of $4.76 million per breach (IBM, 2024). Phishing usually uses social engineering to trick people into giving up their login information or other private information. This is usually done through fake emails, text messages, or websites (ENISA, 2024).

While text-based phishing emails have been extensively studied and mitigated through Natural Language Processing (NLP), URL blacklisting, and machine learning classifiers, visual phishing attacks leveraging images, logos, or QR codes have emerged as a new and formidable challenge (Basit et al., 2021). These attacks take advantage of the fact that the human visual system relies more on brand familiarity and graphical cues than on text. This makes many standard email filters significantly less effective at detecting such threats. For example, a malicious email might have a real-looking company logo or a QR code that takes you to a domain that collects credentials, but it wouldn't have any links or phrases that text-based analysis could pick up on.

Research on how people think shows that this change is happening. When people look at digital content, they often use fast, heuristic ("System 1") processing, which means they pay more attention to visual cues (like logos, colours, and layouts) than to URLs or security indicators. Research, including Dhamija et al. (2006) and subsequent studies on phishing susceptibility, illustrates that visually convincing replicas of trusted interfaces can mislead even technically proficient users when they are under time constraints.

Behavioural science has done a lot of work on the psychological reasons why people are vulnerable to phishing. Kahneman, D. (2011) The dual-system theory explains how "System 1" thinking, which is quick and intuitive, often guides users' decisions when they are short on time. This means that people tend to rely on familiar visual cues instead of carefully checking the URL or metadata. Jakobsson and Myers (2006) further illustrate that phishing attacks are effective by leveraging cognitive overload, ambiguity, and trust heuristics, thereby coercing users into making rapid, fallible decisions. This human tendency to prioritise visual familiarity makes image-based phishing particularly effective and underscores the necessity for defensive tools to not only identify textual anomalies but also to analyse logos, layout similarities, and other perceptual markers emulated by attackers.

From a macro perspective, several recent incidents and trends have underscored the significance of social engineering and visual deception. For instance, the MGM Resorts breach in 2023 used advanced social engineering and MFA bypass, showing that people are still being taken advantage of instead of just technical flaws. At the same time, threat-intelligence reports from 2024 to 2025 show a big rise in QR-based phishing ("quishing"), where QR codes in images or PDFs are used to send bad URLs. Users often scan these codes on their own mobile devices, which are not protected by enterprise security.

This increasing sophistication shows how important it is to have computer-vision-based phishing detection systems that can look at and understand the visual parts of phishing artefacts. Image-based threats are especially troublesome in enterprise Security Operations Centres (SOCs), where analysts have to look at suspicious attachments or screenshots by hand to see if they are real. The manual verification process takes a lot of time and is easy to make mistakes, which makes analysts tired and slows down incident response times (Trend Micro, 2024).

1.2 Defining the Problem

Even though secure email gateways and advanced phishing detection algorithms are widely used, there is still a gap in detection when it comes to analysing the visual layer of phishing artefacts. Most current solutions, like Microsoft SmartScreen, Proofpoint, or Google Safe Browsing (Proofpoint, 2023), focus on text-based indicators and URL reputation. The interpretation of visual elements like images, QR codes, and screenshots is still mostly done by hand (Check Point, 2024).

Because of this limitation, many phishing campaigns can get around automated defences by putting their payloads inside images. These images often copy parts of a brand's identity, like colours, logos, and layouts, to make fake login pages or invoices that look real. Attackers make it harder to find malicious links in QR or 2D barcodes, which lets them get around systems that depend on scanning plaintext URLs.

From an operational point of view, this problem puts a lot of extra work on SOC analysts, who must open and look at each suspicious image attachment on their own. In a fast-paced security environment, this kind of manual triage slows down response times, makes it harder to fix problems, and makes people tired of getting alerts (CISA, 2024). The issue is twofold: a technical hurdle in facilitating automated visual content analysis, and a procedural obstacle in incorporating this automation smoothly into SOC workflows while maintaining analyst trust and clarity.

1.3 Project Overview

The goal of this project is to fill this gap in detection by creating an automated system that can find and understand phishing signs that are hidden in images.

ImageAware+ gives SOC analysts clear, useful information about potentially harmful visual content by using computer vision, optical character recognition (OCR), and threat intelligence integration.

The system works in a modular way:

1. Image preprocessing: Using OpenCV to clean, normalise, and improve input images for the best OCR and QR decoding performance.

2. OCR text extraction: Using the Tesseract OCR engine to find text elements like login prompts, brand names, or phishing phrases.

3. QR/2D barcode decoding: Using PyZbar and OpenCV to get URLs or other data that is hidden inside.

4. Threat-intelligence enrichment: Checking extracted URLs against APIs like VirusTotal and PhishTank.

5. Heuristic risk scoring: Setting levels of threat confidence based on what you find.

6. Making reports: Making standard JSON or PDF reports with pictures that have been marked up for analysts to look over.

ImageAware+ makes these steps easier by automating them. This makes things more efficient, cuts down on manual work, and gives SOC environments clearer threat reports that help with training and decision-making. The focus on explainability makes sure that the system not only finds suspicious artefacts but also explains why they are seen as risky.

1.4 Research Motivation

My direct experience in an operational cybersecurity setting is what gave me the idea for this project. As an intern working as a SOC Analyst, I often got email alerts with image attachments that were marked as "potentially suspicious" but didn't give any more information. Analysts had to look at these pictures by hand, decode the QR codes that were hidden in them, and look for strange things. This could take a few minutes for each sample.

In large businesses, where analysts get thousands of alerts every day, this manual method is not likely to work. Seeing this inefficiency led to the idea for ImageAware+, a tool that not only automates detection but also makes it easier to understand, which is a key need in modern cybersecurity analytics (Doshi-Velez & Kim, 2017). The system increases analyst trust, lowers cognitive load, and helps with better incident documentation by making the reasoning behind detections clear.

As phishing techniques become more sophisticated, using text, images, and embedded code to trick people, traditional keyword filters and URL blacklists are no longer enough. The project thus contributes to a nascent research domain at the confluence of computer vision, threat intelligence, and explainable AI (XAI) in cybersecurity.

1.5 Goals and Objectives

The primary objective of this research is to create an automated, transparent framework for identifying image-based phishing threats, incorporating visual analysis, optical character recognition (OCR), and external threat intelligence.

The goals are:

- Doing a thorough review of the literature on how phishing has changed, how visual deception works, and how image analysis is used in security.
- Creating a modular system architecture that can take in images, analyse them, and report on them.
- Using open-source tools to set up a text extraction pipeline based on OCR.

- Using external APIs to check threats and add to them.
- Using controlled datasets or representative samples to check how accurate, reliable, and scalable a system is.
- Making visual reports that can be explained and are good for SOC documentation and analyst training.
- Evaluating the viability of SOC integration concerning workflow implications, trust, and transparency.

1.6 Questions for Research

1. How can image analysis be used to find visual phishing signs like logos, text, and QR codes?

2. What mix of algorithms and outside threat-intelligence services gives the best detection accuracy while still being easy to understand?

3. How can an automated visual phishing detection system work with current SOC workflows to make analysts more productive?

4. What kind and level of explanation do SOC analysts need to trust and use automated visual-phishing detection results?

1.7 Scope and Limitations

ImageAware+ can only find phishing signs in still images. It doesn't include real-time scanning of email inboxes, running attachments in a sandbox, analysing the DOM in a browser, or analysing video content. The system mostly looks for phishing content in English and looks for visual patterns that are common in phishing campaigns around the world.

Some of the limitations are:

- OCR accuracy is needed, but it may not work well with images that are low-resolution, distorted, or heavily compressed.
- Free-tier threat-intelligence services have limits on how many API calls can be made and when they can be made.
- OCR models often struggle with multilingual, stylised, or non-Latin text.
- Lack of a substantial, publicly accessible phishing-image dataset for thorough quantitative assessment.

Even with these limitations, the research intends to establish a robust foundation for future advancements in adversarial image detection, multilingual OCR, deep-learning-driven visual similarity analysis, and comprehensive SOC/SOAR integration.

1.8 Report Structure

The structure of this report is as follows:

- Section 2.0 provides a comprehensive literature review and technical examination of pertinent topics, including the evolution of phishing, image analysis, OCR technology, threat intelligence integration, dataset design, and explainable automation.
- Section 3.0 looks at the risks to security, ethics, privacy, and deployment that come with automated phishing analysis.
- Section 4.0 sums up the results and makes decisions about choosing tools, designing architecture, research results, and future plans.
- Section 5.0 has detailed algorithmic breakdowns and pseudo-code for modules that have been implemented or are planned.
- Section 6.0 has a list of technical terms and acronyms.
- Section 7.0 has a list of all the sources used in APA style.

2.0 Overview of Areas, Technologies, or Topics Researched

This part gives a thorough look at the main parts that make up the ImageAware+ system. Each subsection looks at the theoretical background, previous research, and practical technological issues that went into designing the project. These investigations collectively furnish the academic rationale for ImageAware+'s architecture, tool selection, and algorithmic methodology.

2.1 The Evolution of Phishing and the Rise of Visual Deception

Phishing has changed a lot since it first appeared in text form in the 1990s. Phishing used to be simple text-based scams like fake bank emails. Now, it has become a multimodal deception technique that uses advanced social engineering, contextual adaptation, and sophisticated visual imitation (Basit et al., 2021). The primary catalyst for this evolution is the growing awareness and technological proficiency of users; as textual indicators became more discernible, attackers transitioned to visual and behavioural imitation to sustain effectiveness (Abdelnabi et al., 2020).

Traditional phishing detection methods, including blacklist-based, heuristic, and machine-learning classifiers, mainly depend on text features like the subject line of an email, the domain of the sender, or the text of a hyperlink (Shahriar & Zulkernine, 2012). But modern phishing campaigns often use images, QR codes, or base64-encoded objects to hide malicious links, which these text filters can't catch (Trend Micro, 2024).

In addition to specific attack methods, a number of recent reviews have tried to describe the overall evolution of phishing. Osamor et al. (2025) delineate the evolution of phishing through distinct phases: initial widespread email campaigns utilising generic lures; subsequent targeted spear-phishing and "whaling" aimed at high-value individuals; expansion into various channels including SMS, social media, and voice calls; and, most recently, the incorporation of AI-generated content and multimodal deception. The Phishing Playbook from PhishFirewall, which is aimed at professionals, shows how enterprise, mobile, chat, and voice-based phishing threats now coexist and strengthen each other in complex attack chains (PhishFirewall, 2024). These sources collectively demonstrate that phishing has changed from a problem that mostly happened through text-based email to a wide range of social-engineering attacks that use many different ways to communicate and rich media.

2.1.1 From Phishing with Text to Phishing with Pictures

The shift from text-based to image-based phishing, also known as "visual phishing," is in line with other trends in cybersecurity that are becoming more aggressive. Attackers are more and more going after the human perceptual system instead of the technical flaws in machines (ENISA, 2024). For example, login portals that look the same can trick people into entering their credentials even if the URL is different. Lin et al. (2022) conducted research on visual similarity attacks, revealing that cloned login interfaces can attain deception rates exceeding 80% among users acquainted with the impersonated brand.

Image-based phishing (IBP) uses a number of different methods:

- Embedded QR codes ("quishing") send victims to bad websites.
- Emails that only have images that look like real branding to get around text filters.
- Screenshots of fake invoices that have hidden links or instructions that are harmful.

These methods take advantage of the weaknesses in OCR and image-parsing tools in email gateways, which makes them a major detection blind spot (Check Point Research, 2024). Initial scholarly research on visual phishing has acknowledged that fraudulent websites must replicate the appearance of authentic ones. Medvet, Kirda, and Kruegel (2008) suggested one of the first phishing detection systems based on visual similarity.

This system compares screenshots of candidate pages to known legitimate sites using layout, colour, and structural features instead of just text cues. Their experiments demonstrated that visual similarity serves as a relatively reliable indicator of phishing behaviour, primarily due to the difficulty attackers face in evading imitation of the brand they intend to impersonate. Later studies built on this idea by looking at how similar phishing pages that look like the same brand are, even if the person looking at them doesn't know about the original legitimate site. This further supports the idea that visual mimicry is a key part of modern phishing detection.

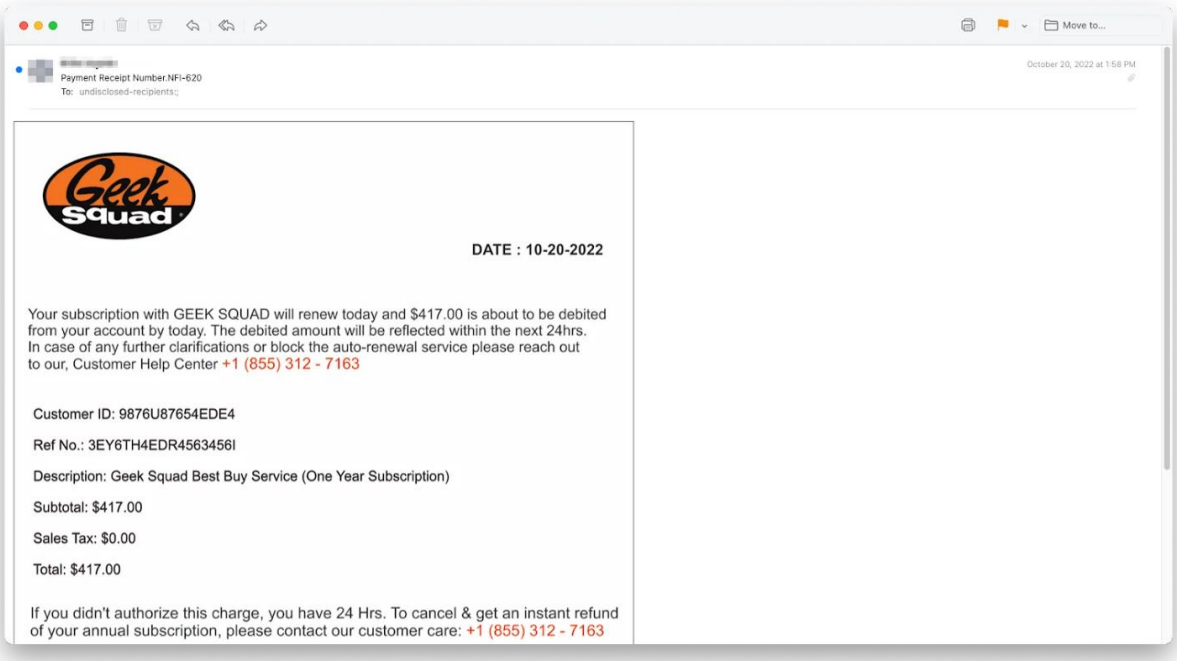

Figure 1: Example of Image-Only Phishing Email

2.1.2 Impact on Security Operations Centres (SOCs)

In today's SOCs, phishing is still the most common type of incident that analysts deal with (IBM, 2024). But most detection tools, like Proofpoint, Microsoft Defender, and Sophos, don't give much information about why an email or attachment was flagged. Analysts have to look at attachments by hand, which often means opening or decoding files that could be dangerous (CISA, 2024).

The manual triage bottleneck that results from this slows down the Mean Time to Detect (MTTD) and Mean Time to Respond (MTTR), which are two key metrics for SOC efficiency. ImageAware+ aims to make visual analysis automatic and give clear reasons for its decisions. This will make both detection and analysis faster and more accurate.

2.1.3 Future Phishing Trends: LLMs, Multimodality, and Generated Campaigns

Recent research indicates that phishing will persist in its evolution regarding both magnitude and complexity. Osamor et al. (2025) predict a growing convergence between conventional phishing and new technologies like deepfakes, synthetic voices, and AI-generated content, resulting in highly personalised and context-sensitive attacks. The availability of large language models (LLMs) and generative AI is a major factor in this change. These technologies can be used to write convincing emails and make fake phishing artefacts on a large scale. These changes show that phishing will become more and more common in the future, using both realistic synthetic visual artefacts and persuasive language generation.

This means that multimodal detection methods will be needed.

Chen et al. (2025) present PEEK, a "phishing evolution framework" that employs LLMs to produce varied phishing samples and examine the temporal evolution of attack patterns. Their findings indicate that LLM-generated phishing can demonstrate superior linguistic quality and diversity compared to numerous existing datasets, thereby complicating detection by static filters. The framework also shows that attackers can systematically test and change detection models by making new variants over and over again. This has direct effects on visual phishing: generative models can make fake login pages, email templates, and branded images that look real but still have bad links or QR codes in them.

From the defender's point of view, these trends make it even more important to have defences that are flexible, easy to understand, and work in multiple ways. ImageAware+ and other systems like it can help with this by focussing on strong visual cues (like logos, layouts, and QR codes) that are still important even when text content changes and is generated by AI. Future endeavours may necessitate the integration of generative adversarial testing, wherein visual-phishing detectors are assessed and fortified against synthetic campaigns produced by frameworks such as PEEK. (Osamor et al., 2025; Chen et al., 2025). After looking into how phishing has changed and how visual deception has grown, the next section looks at the technical basics needed for automated analysis of image-based phishing artefacts: computer vision and Optical Character Recognition (OCR).

2.2 Cybersecurity with Computer Vision and OCR

2.2.1 What Computer Vision Does

Computer vision (CV) is the process of teaching machines how to understand what they see. In cybersecurity, CV applications encompass malware visualisation, image-based spam filtering, and brand logo identification on fraudulent websites. Visual cues are very helpful for spotting phishing content because attackers copy logos, button shapes, and page layouts to trick people (Abdelnabi et al., 2020).

Deep-learning-based vision systems, like Convolutional Neural Networks (CNNs), have shown to be very good at finding phishing (Lin et al., 2022). But these models usually need large amounts of labelled data and computing power that isn't good for real-time SOC workflows. ImageAware+ uses a hybrid computer vision pipeline that combines deterministic algorithms (OpenCV) with heuristic logic. This balances speed, accuracy, and ease of understanding.
Along with academic research, businesses have started to use visual AI systems on a large scale to find phishing scams. For instance, VISUA's anti-phishing platform uses computer vision to find high-risk brands, logos, icons, and other visual "threat signals" in emails and web pages. It then combines these signals with traditional rule-based and machine-learning detectors to come up with an overall risk score (VISUA, n.d.). Instead of just looking at HTML structure or network indicators, these systems "look" at content like a person would, but at machine speed. They prioritise messages that visually reference sensitive brands (like banks or parcel services) for more in-depth analysis. This trend in industry backs up the main idea behind ImageAware+, which also treats visual analysis as an important part of a

larger phishing-detection process.

2.2.2 Optical Character Recognition (OCR)

Optical Character Recognition (OCR) changes the text in images into text that computers can read. Since the 1990s, it has been a key part of digitising documents, and open-source engines like Tesseract have made big strides in this area (Smith, 2007). When looking for phishing, OCR is very useful for getting login prompts, embedded instructions, and encoded URLs.
Guo et al. (2021) found that combining OCR with heuristic keyword analysis made it 17% more accurate at finding phishing emails than just using text-based methods. Some of the most common keywords that people look for are "verify," "account suspended," and "update information."

But the accuracy of OCR depends a lot on the quality of the image. Preprocessing methods like converting to greyscale, binarization, and noise reduction make character recognition much better, especially for screenshots with low contrast. ImageAware+ includes this kind of preprocessing in its pipeline so that OCR works the same way on different datasets.

2.2.3 Techniques for Preprocessing and Improving

To get rid of background noise and make text edges stand out, we use OpenCV functions like cv2.GaussianBlur() and cv2.adaptiveThreshold(). Morphological operations (cv2.morphologyEx()) make the text boundaries even clearer, which makes Tesseract's segmentation more accurate.

These preprocessing steps make sure that OCR accuracy stays the same across different types of phishing images, like invoices, login screenshots, and promotional graphics with QR codes.
Section 2.3 moves on from OCR and visual processing to QR and barcode detection, which is becoming a more common way for modern phishing attacks to work.

2.3 Finding QR codes and barcodes in phishing Contexts

2.3.1 The Rise of "Quishing"

"Quishing," or QR code phishing, is a growing cybersecurity threat for both businesses and consumers. Attackers put bad URLs in QR codes to take advantage of people's trust and ease of use (Trend Micro, 2024). According to a Check Point report from 2024, QR-based phishing attacks rose by 587% from 2022 to 2024 (Barracuda Networks, 2024).
Organisations are at risk because traditional email scanners can't read QR codes that are hidden in images. Also, users often scan these codes with their own mobile devices, which completely bypasses enterprise protections.

2.3.2 Detection Techniques

Libraries like PyZbar, ZXing, or ZBar that are open source are often used to read QR and barcode data. These libraries use pattern recognition to read standard QR structures like data cells, finder patterns, and alignment markers.
With ImageAware+, PyZbar's Python bindings let you decode QR content directly from base64 or rasterised email images. The extracted data is then cleaned up and sent to the Threat Intelligence Analysis module to be checked against databases of known malicious domains.

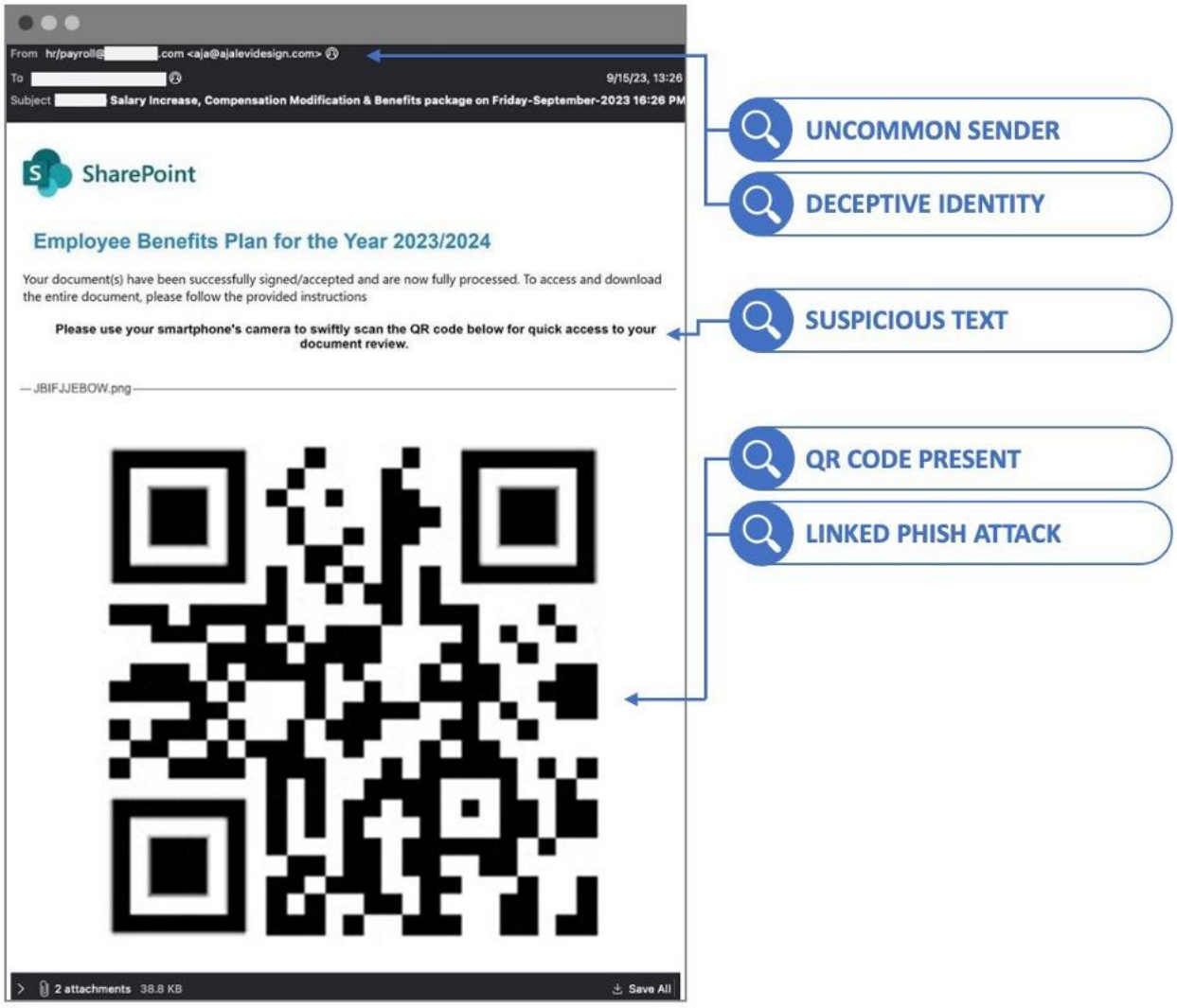

Figure 2: QR Code within Phishing Email ("quishing")

2.3.3 Intergration with Threat Intelligence

After being decoded, URLs are checked against each other using APIs like VirusTotal, PhishTank, and Google Safe Browsing. Shahriar and Zulkernine's (2020) research showed that adding real-time threat intelligence made it possible to find phishing URLs with up to 22% more accuracy. ImageAware+ automates this process, giving you immediate feedback

on whether the decoded link is real. The next step is to use external threat-intelligence services to check the validity of URLs and indicators after extracting visual and textual content from images.

2.4 Threat Intelligence Integration and URL Analysis

2.4.1 Why Threat Intelligence Is Important

Threat intelligence (TI) is the process of gathering, linking, and analysing information about threats that are happening now or that could happen in the future. TI gives important information about domain reputations, known attack patterns, and live indicators of compromise (IoCs) when it comes to phishing detection (ENISA, 2024).

Without TI enrichment, systems could mistakenly flag safe URLs as suspicious, which is a false positive. ImageAware+ combines information from many different TI sources to get a better idea of reputation and make classification more reliable.

But threat-intelligence feeds and URL-reputation systems are not perfect. Le Pochat et al. (2019) conducted a large-scale evaluation that shows that blacklist and reputation services often take a long time to find newly registered malicious domains. This creates "zero-hour" gaps that attackers can use. The study also points out that different providers don't always agree, which can lead to false negatives when bad URLs are wrongly identified as safe and false positives when automation is too aggressive. These restrictions are especially important for visual phishing, where attackers can hide new URLs in QR codes or image-based lures. ImageAware+ reduces these risks by cross-checking information from many sources and using information from OCR and visual analysis to add context. However, it does not get rid of them completely.

2.4.2 API-Driven Intelligence

APIs like VirusTotal and PhishTank give you programmatic access to a lot of security data.

- VirusTotal combines dozens of antivirus programs and domain reputation engines to give you JSON-formatted verdicts and confidence scores.
- PhishTank keeps a curated list of phishing URLs that users have sent in.

ImageAware+ sends decoded URLs to these services, gets their decisions, and saves them as structured fields in the final report. This architecture, which is based on APIs, makes it easy to add more intelligence feeds (like AbuseIPDB and AlienVault OTX).

2.4.3 Heuristic Risk Scoring

ImageAware+ uses TI feedback and heuristic scoring based on:

- The number of phishing-related keywords found through OCR.
- The visual similarity of logos or layouts that have been taken out (possibly using histogram or template analysis in the future).
- The number and type of positive detections made by external APIs.

Each factor adds to a total confidence score that is on a scale from 0 to 100. This hybrid scoring method combines real-world data (through APIs) with contextual clues (through OCR and CV) to give results that can be explained and are good for analyst review.

2.5 Algorithmic Approaches and Design Considerations

2.5.1 Philosophy of Design

The ImageAware+ architecture puts a lot of weight on being open, modular, and easy to understand. Its deterministic algorithms let analysts follow each decision path, which is in line with the principles of Explainable Artificial Intelligence (XAI) (Doshi-Velez & Kim, 2017). This is different from black-box deep learning models.

2.5.2 Basic Algorithms

1. Image Preprocessing: Adaptive thresholding and noise reduction make sure that OCR and QR detection get the same input every time.
2. OCR Text Extraction: Tesseract's LSTM engine can split up and recognise characters.
3. Scanning binary patterns with PyZbar to decode QR codes.
4. Threat Validation: VirusTotal and PhishTank communicate with each other using a REST-based API.
5. Risk scoring and visualisation: creating images with weighted composite scores.

2.5.3 Explainability and Interpretability

There are many levels of explainability built in:

- Annotated outputs make it easy to see areas of interest that have been found.
- Reports have confidence metrics and reasons for classification.
- Analysts can recreate intermediate outputs, like OCR text dumps, to get a better idea of how the process works at each step.

This makes sure that both auditing and learning are possible, which is very important in both academic and operational cybersecurity settings.

2.5.4 Evaluation of Comparisons

Visual phishing detection frameworks like VisualPhishNet (Abdelnabi et al., 2020) and PhishIntention (Liu et al., 2023) use deep CNN architectures and big image datasets. They work well in controlled experiments, but their complexity and need for a lot of computing power make them hard to use in places where resources are limited or where explainability is very important. ImageAware+, on the other hand, prioritises speed, transparency, and integration over deep learning accuracy in favour of interpretability and real-time usability.

Visual-similarity-based approaches constitute a significant complementary avenue of research. Medvet et al. (2008) showed that using layout and colour features to compare screenshots of candidate pages with real sites can help find phishing sites, especially those that closely copy well-known brands. Subsequent visual-similarity frameworks have improved this concept by using shorter descriptors and better matching schemes, which allow for comparisons that are almost real-time at the browser or gateway level. VISUA's Visual-AI platform combines logo detection, text detection, and visual threat-signal scoring into existing anti-phishing products in the business world. This makes it possible to use visual similarity and brand-focused analysis on a large scale (VISUA, n.d.).

ImageAware+ focusses on a slightly different trade-off space than these systems. Instead of keeping reference screenshots of a lot of real sites, it focusses on getting generic visual indicators (like text content, QR codes, and high-risk phrases) and adding threat-intelligence lookups to them. This eliminates the necessity for extensive screenshot repositories while concurrently capturing numerous visual cues utilised in visual-similarity systems. The success of both Medvet et al.'s work and VISUA's commercial deployments indicates that future iterations of ImageAware+ could integrate optional logo or template-matching features to enhance precision in combating highly branded phishing campaigns. (Medvet et al., 2008; VISUA, n.d.)

2.6 Implementation Tools and Technology Choices

Tool / Technology	Purpose	Justification
Python 3.x	**Core programming language**	**Widely used in cybersecurity, strong library ecosystem**
OpenCV	**Image preprocessing and visualization**	**Efficient, open-source, supports multi-format images and advanced CV operations**
Tesseract OCR	**Text extraction**	**Industry standard, open-source, supports multilingual OCR with LSTM-based engine**
PyZbar / ZXing	**QR and barcode decoding**	**Lightweight and accurate decoding of standard QR/barcode formats**
Flask / FastAPI	**REST API and backend integration**	**Enables modular microservices architecture and integration with SOC/SIEM tools**
VirusTotal API	**Threat intelligence**	**Aggregated reputation scoring from multiple engines**
PhishTank API	**Open-source phishing repository**	**Free, community-verified indicators**
Pandas / NumPy / Scikit-learn	**Data handling and analytics**	**Supports risk scoring, evaluation, and experiment tracking**
Matplotlib / FPDF	**Report generation and visualization**	**Provides explainable, annotated outputs for analyst consumption**

2.7 Design and Limitations of the Dataset

2.7.1 No Open Phishing-Image Datasets Available

A major problem with research on visual phishing is that there aren't many openly available, labelled phishing-image datasets. PhishTank and OpenPhish are examples of repositories that give URLs, but they don't usually include screenshots or image attachments because:

- worries about copyright issues with brand assets,
- privacy issues when user content or PII could be there,
- hosting and licensing terms that don't match up.

This lack of data makes it hard to do large-scale, statistically sound tests of visual-phishing detection methods.

2.7.2 Legal and Ethical Limits

Taking real phishing screenshots from production environments brings up a lot of legal and moral questions:

- Privacy and GDPR: Real artefacts may have personal information like names, email addresses, and account IDs, so strict controls on data minimisation, anonymisation, and retention are needed.
- Intellectual Property: Screenshots of branded interfaces may be protected by copyright, which means they can't be shared or published in public datasets.
- Security Risk: If you don't handle it safely, storing real-world phishing content can make you a tempting target.

ImageAware+ addresses these concerns by concentrating on publicly available samples, steering clear of user-specific content, and preserving only hashes and extracted features whenever feasible.

2.7.3 Making a Useful Evaluation Corpus

Given the limitations of this FYP, a practical evaluation corpus can be developed by:

- taking screenshots of known phishing URLs from public feeds like PhishTank,
- getting harmless control samples from real login pages and branded messages,
- keeping a clear line between harmless and harmful sets.

Even though this corpus is small, it is enough to check the behaviour of the pipeline and the quality of the explanations, even if it is not enough for thorough statistical benchmarking.

2.7.4 Synthetic Data Generation

You can make synthetic data to go along with real samples:

- making login pages and invoices that don't have any brand logos or text,
- putting QR codes that point to internal test URLs or URLs that can't be routed,
- using noise, scaling, and compression artefacts to make email rendering look like it does in real life.

Synthetic datasets help test OCR and QR modules under stress without putting privacy or copyright at risk. But they need to be tested against real-world samples to make sure they don't fit too well to conditions that are too clean.

2.8 Cybersecurity Automation Explainability

Research in explainable machine learning underscores the significance of transparency in operational cybersecurity. Carvalho et al. (2019) contend that interpretability is crucial in high-stakes domains, allowing users to comprehend, validate, and challenge automated decisions. Arrieta et al. (2020) present a taxonomy of XAI techniques and assert that explainability is not only a usability feature but a prerequisite for trust, auditability, and regulatory compliance. In SOC environments, where analysts must explain why they took certain actions to contain an incident or escalate it, black-box models can make it harder to respond to incidents by hiding the reasons behind detections. ImageAware+ follows these rules by giving clear, step-by-step access to OCR results, QR code decoding, and threat-intelligence verdicts. This makes sure that automated detections can still be checked by people.

2.8.1 Role of XAI in SOC Environments

Explainable AI (XAI) frameworks, like the ones made in the DARPA XAI program, focus on:

- reasons that people can understand,
- the capacity to challenge or supersede automated determinations,
- clear handling of uncertainty (Doshi-Velez & Kim, 2017).

This means in the context of a SOC:

- showing what indicators caused a detection (for example, a decoded QR URL, certain keywords, or TI verdicts),
- allowing "drill-down" views of the steps in the intermediate analysis,
- not using black-box classifications that analysts can't understand or defend.

2.8.2 LIME/SHAP and Possible Add-Ons

ImageAware+ mostly uses deterministic algorithms, but future versions could use XAI tools like:

- LIME (Local Interpretable Model-Agnostic Explanations): this method looks at complex models in small groups to find the features that had the biggest effect on the score, like words in OCR text or the presence of a QR code.
- SHAP (Shapley Additive Explanations): giving each feature in a composite risk score an additive contribution.
  Even with heuristic scoring, the same idea is used: contributions to the score are broken down and shown to the analyst instead of being hidden in a single unclear probability.

2.8.3 Deterministic and Deep-Learning Methods

Deep CNNs and Vision Transformers can recognise things very well, but it's hard to understand how they do it. ImageAware+ and other deterministic pipelines give:

- clear, step-by-step logic,
- audits are easier to check,
- fewer barriers to adoption in SOCs that are afraid of black-box automation.

For this FYP, being able to explain things is more important than small improvements in accuracy that might come at the cost of openness.

2.9 Comparative Analysis of Tools and Libraries

A comparative analysis was conducted to assess alternative tools against project requirements:

Tool / Service	Purpose	Pros	Cons	Chosen
OpenCV	**Image preprocessing & analysis**	**Mature, high-performance CV library; extensive functions**	**Steeper learning curve**	**Yes**
Pillow (PIL)	**Basic imaging**	**Lightweight, easy for simple operations**	**Limited advanced CV**	**No**
Tesseract OCR	**Text extraction**	**Open-source, widely used, multilingual**	**Sensitive to noise; requires preprocessing**	**Yes**
EasyOCR	**OCR alternative**	**Handles many fonts, languages out-of-the-box**	**Heavier dependencies, slower**	**No**
PyZbar	**QR/barcode decoding**	**Simple, accurate, Python-friendly**	**Focused on common symbologies**	**Yes**
ZXing	**QR/barcode decoding**	**Robust, multi-language**	**More effort to integrate in Python**	**No**
VirusTotal API	**Threat intelligence**	**Aggregated multi-engine verdicts; rich metadata**	**Rate limits, data-sharing policies**	**Yes**
Google Safe Browsing	**URL reputation**	**Strong coverage of web threats**	**Less contextual detail for SOC**	**No**
Flask	**REST API backend**	**Lightweight, easy to integrate with Python**	**Less scaffolding than FastAPI**	**Yes**
FastAPI	**REST API backend**	**Typed, async, high performance**	**Slightly higher complexity**	**No (future option)**

The chosen combination (OpenCV, Tesseract, PyZbar, Flask, VirusTotal, PhishTank) represents a pragmatic trade-off between capability, transparency, complexity, and FYP feasibility.

3.0 Security Implications

3.1 Importance of Cybersecurity

Cybersecurity is becoming more about not just how well technology works, but also how fast, clear, and understandable defensive measures are. Phishing is still the most common way for hackers to get into a computer system. In 2024, more than 91% of successful data breaches started with a phishing attack. More than half of these attacks used some kind of visual trick, like hidden logos or QR codes (ENISA, 2024). This number shows how important it is to make systems that can look at the visual part of communication, which traditional email filters and anti-phishing engines have ignored for a long time.

The ImageAware+ system helps meet this new need for cybersecurity by offering an automated, explainable, and modular way to find visual phishing. The project fits perfectly with three main areas of cybersecurity practice:

1. Email and Content Security: ImageAware+ adds to traditional filters by looking at image attachments and inline visuals to find deceptive elements that text-based systems can't see.

2. Threat Hunting and Incident Response: The system can automate some of the phishing analysis pipeline by working with platforms like King Phisher and SIEM tools. This lets analysts focus on more complicated cases.

3. Security Awareness and Education - Explainable, annotated reports generated by ImageAware+ can serve as teaching tools within SOC training, illustrating how visual phishing operates.

3.1.1 Alignment with the CIA Triad

The CIA triad - Confidentiality, Integrity, and Availability, is a basic way to judge how safe a system is. ImageAware+ helps each part in its own way:

- Confidentiality: It protects private information from being shared without permission by finding phishing images that try to steal credentials.

- Integrity: The system's reports give investigators proof that can be checked and followed up on, making sure that the analysis is correct.

- Availability: Automating repetitive manual tasks makes the work of analysts easier, which makes more people available for SOC operations.

3.1.2 Contribution to Cyber Resilience

The European Union Agency for Cybersecurity (ENISA, 2024) says that cyber resilience is the ability of an organisation to plan for, deal with, recover from, and adapt to bad situations. ImageAware+ improves anticipation and resistance by helping people spot visual phishing attempts early. This leads to faster response cycles and data-driven training feedback loops. The ability to explain detections also helps with continuous learning, which is a key part of a strong cybersecurity architecture (National Cyber Security Centre, 2023).

3.2 Ethical, Privacy, and Legal Considerations

Automation in cybersecurity brings with it both technical advantages and moral obligations. The ImageAware+ project works with user-generated content that could be sensitive, like screenshots, attachments, or brand images. So, privacy, consent, and responsible data handling are all important for the system to work in an ethical way.

3.2.1 Data Protection and GDPR Compliance

The system must follow the General Data Protection Regulation (GDPR), which says that data must be kept to a minimum, processed lawfully, and stored only for a short time (European Commission, 2021). This is because it may look at phishing reports or email attachments that users have sent in.

ImageAware+ is made to only process the data needed for analysis and not keep any information that could identify a user. Before being stored or sent to external APIs, any URLs, email addresses, or embedded text that were found during OCR analysis are cleaned up. Also:

- HTTPS/TLS 1.3 encrypts all data in transit.
- Unless needed for research evaluation, raw images or decoded QR contents are not kept permanently.
- Users are told when data is sent to outside threat-intelligence services.

These safety measures make sure that the system follows both EU and global standards for data processing, such as ISO/IEC 27001.

3.2.2 Proper Use of External APIs

VirusTotal and PhishTank are examples of external APIs that combine data from many sources. Some of these sources may have copyrighted or personally identifiable information. VirusTotal's policy says that samples sent in by users may be shared with security vendors that take part. So, ImageAware+ makes sure that any API queries are only for extracted URLs or hashes, not whole images. This stops private visual data from being shared by mistake.

3.2.3 Bias, Explainability, and Human Oversight

When using automated systems, you need to think about algorithmic bias and explainability. ImageAware+ uses deterministic algorithms instead of machine-learning models that aren't clear, but biases can still happen in:

- the keyword lexicon used for heuristic scoring, which is mostly based on English words,
- the datasets used for testing, which might show some brands or types of attacks more than others.

To lessen this, human supervision is still very important. Analysts look over reports with notes and can change heuristics based on what they know about the situation. This model with a person in the loop is in line with new EU rules for Trustworthy AI (European Commission, 2021) because it keeps things honest and accountable.

3.2.4 Ethical Research Practices

The processes for building and testing the ImageAware+ dataset follow ethical research standards:

- Only phishing samples from open repositories that are available to the public (like PhishTank and OpenPhish) are used.
- There is no real user data or private business information in it.
- The project is only for defence and education.

The system shows how automation can help human defenders instead of replacing them. This is in line with the ACM Code of Ethics, which stresses doing good, being open, and avoiding harm in computing research.

3.3 Potential System Vulnerabilities and Mitigations

ImageAware+ improves the ability to find phishing attacks, but any security system should be carefully checked for weak spots. Being aware of these problems makes it possible to put in place effective fixes and makes sure that they are used responsibly.

3.3.1 Adversarial Image Evasion

Adversarial machine learning research indicates that meticulously designed images can mislead even advanced vision models (Goodfellow et al., 2015). Attackers could, in theory, create images with adversarial noise or hidden QR patterns that are hard for OCR and computer vision to read.

Recent studies show that adversarial manipulation is a big problem for security systems that use vision. Xu et al. (2017) demonstrate that minor, meticulously crafted perturbations, undetectable by the human eye, can significantly modify OCR and image classification results. Their "feature squeezing" defence shows how attackers use high-dimensional pixel representations to add noise that can't be found. In the same way, Eykholt et al. (2018) show that changes to the physical world, like printed stickers or distortions, can also fool computer vision models, including those used in self-driving cars. In the context of phishing, these methods could let attackers put fake artefacts into screenshots, QR codes, or branded visual elements, which would make OCR less accurate or stop QR codes from being read correctly. Together, these findings demonstrate the importance of robust preprocessing, repeated decoding attempts under varying conditions, and human review whenever automated modules return ambiguous results.

Mitigation:

- Use multi-layer preprocessing to stop common ways of getting around (like contrast normalisation and noise filtering).
- Use ensemble validation, which means doing multiple decoding passes at different thresholds.
- Keep track of all failed decoding attempts so that someone can look through them by hand to find possible malicious inputs.

3.3.2 API Dependency and Availability Risks

ImageAware+ uses external threat-intelligence APIs, so any downtime, rate limiting, or spreading of false data can make the system less accurate.

Mitigation:

- Add local caching of API responses to cut down on the need for real-time queries.
- Set up backup systems, like secondary reputation services (for example, AbuseIPDB or AlienVault OTX).
- Use checksums to make sure the API is working properly, and keep an eye on API responses for strange behaviour.

3.3.3 Data Handling and Confidentiality Risks

When you extract and analyse images, there is a risk of data leaking, especially when you are working with real-world phishing samples that contain user or company information.

Mitigation:

- Only allow disc persistence if absolutely necessary; otherwise, enforce in-memory analysis (RAM-only).
- Before making a report, mask any sensitive text strings.
- Do regular penetration tests to make sure that the backend (Flask/FastAPI) doesn't have any open endpoints or debug interfaces.

3.3.4 Model Drift and Heuristic Staleness

As phishing techniques change, static heuristic models may not work as well, which can cause model drift, or a drop in detection accuracy over time.

Mitigation:

- Set up a way for analysts to give feedback so they can mark false negatives and retrain heuristics.
- Keep keyword lists and API integrations up to date so they reflect the latest phishing campaigns.
- Keep the modular architecture so that updates are easy to make without changing the whole system.

3.3.5 Supply Chain and Library Security

Open-source tools like OpenCV and Tesseract are powerful, but they may have security holes if they are old or not properly checked.

Mitigation:

- Use requirements.txt to pin versions and patch dependencies on a regular basis.
- Check the integrity of the library with hashes or digital signatures.
- Use isolated virtual environments, like venv or Docker containers, to make the attack surface smaller.

3.4 Secure System Integration and SOC Deployment

A major part of the ImageAware+ research is that it can be used in real-world SOC settings. Deployment in these kinds of systems requires following both technical and procedural security rules.

3.4.1 Secure Architecture Integration

You can use ImageAware+ on its own or add it to an existing phishing response workflow. The recommended deployment model uses a three-tier security architecture:

- Input Layer: Gets screenshots or email attachments. Here, validation happens to stop file-based attacks like malformed images.
- Processing Layer: Does preprocessing, OCR, QR decoding, and threat validation in a separate space. This layer is packaged so that it can run in a sandbox.
- Output Layer: Makes reports and sends them to SOC dashboards or SIEM systems over HTTPS.

This design follows the principle of defence in depth, which means that each layer has its own access controls and failsafes.

3.4.2 Role-Based Access Control (RBAC)

Users (analysts, supervisors, researchers) should have different access rights when they are part of SOC integration. RBAC policies can limit who can see sensitive results and stop changes to settings. For example:

- Analysts can run scans and look at reports, but they can't change heuristics.
- Administrators are in charge of API keys and keeping dependencies up to date.
- Developers can only access training datasets in a sandbox.

This kind of separation lowers the risk of insider threats and makes people more responsible.

3.4.3 Logging, Auditing, and Incident Response

Every step of ImageAware+, from taking in images to making threat reports, should create logs with timestamps. If there are breaches or false positives, these logs are very important for forensic reconstruction.

Logs should be:

- kept safe and encrypted when not in use;
- checked from time to time to find patterns of abuse;

- added to centralised SOC logging tools like Splunk and the ELK Stack so that they can be compared with other events.

3.4.4 Network and API Security

Talking to outside threat intelligence services can put you at risk for man-in-the-middle (MITM) attacks and data integrity problems. To mitigate this:

- All outgoing traffic must use TLS 1.3 and check the certificate;
- You should keep your API credentials in encrypted configuration files or key vaults.
- To stop replay attacks, requests should have tokens with timestamps.

Rate-limiting and anomaly detection on outgoing queries also help find possible API misuse or compromise.

3.4.5 Integration with King Phisher

By working with King Phisher, an open-source phishing simulation platform, you can do real-world testing and teach users. With this integration, ImageAware+ can look at fake phishing campaigns to see how well the system works, improve heuristics, and teach analysts about real-world examples in a safe setting.

From a security point of view, this integration must make sure that simulated data stays separate from production systems so that it doesn't get mixed up or accidentally exposed.

3.5 Summary

The ImageAware+ project is based on cybersecurity principles and was made using the security-by-design method. Its automation makes organisations more resilient, and its explainable architecture builds trust among analysts and makes sure they follow the rules.

Important things to remember are:

- ImageAware+ fills in a big hole in visual analysis to make email and content security stronger.
- Responsible deployment is made possible by ethical safeguards like GDPR compliance, anonymisation, and human oversight.
- Knowing about possible weaknesses makes it possible to come up with strong ways to deal with them.

- Secure SOC integration makes sure that the system works the way it should in the real world.

ImageAware+ shows not only technical innovation but also a commitment to the basic principles of cybersecurity ethics, resilience, and defense-in-depth through its holistic design.

4.0 Summary and Final Thoughts

4.1 Overview of Research Achievements

The objective of this research was to examine and develop an automated, interpretable system proficient in identifying image-based phishing content. During the project, a thorough examination of the evolution of phishing, image analysis technologies, and threat intelligence systems was conducted. This review identified a significant deficiency in the contemporary cybersecurity domain: although textual phishing detection is advanced, visual phishing analysis is predominantly manual and insufficiently investigated (Basit et al., 2021; Abdelnabi et al., 2020).

The ImageAware+ system was designed to fill this gap by combining computer vision, Optical Character Recognition (OCR), and threat-intelligence APIs. ImageAware+ focusses on explainability and modularity, which is different from many deep-learning-based frameworks. This is in line with real-world SOC needs, where speed and interpretability are often more important than black-box accuracy (Doshi-Velez & Kim, 2017).

The study was able to:

- figured out that SOC operations need automated tools to find phishing attacks that use images;
- looked over and compared technologies that were relevant, such as OpenCV, Tesseract, PyZbar, VirusTotal, and PhishTank;
- made a modular pipeline that combines image preprocessing, OCR, QR decoding, and threat validation;
- set up the groundwork for a standard, easy-to-understand reporting format (PDF/JSON) to help analysts do their jobs.

These results show not only theoretical knowledge but also a clear technical plan for putting the ImageAware+ prototype into action and testing it.

4.2 Key Findings and Insights

4.2.1 The Growing Importance of Visual Phishing Detection

The literature review showed that image-based phishing is becoming one of the most serious cyber threats. The growing use of embedded QR codes, brand impersonation through visual mimicry, and screenshot-based scams shows that perceptual deception is becoming more common (Check Point, 2024). ImageAware+ directly addresses this by expanding phishing detection from text-based analysis to visual analysis.

4.2.2 The Utility of OCR and Computer Vision

OCR, especially Tesseract, was a cheap and reliable way to get text information from phishing images. When used with OpenCV preprocessing methods, OCR gives a lot of interpretability and explainability. This method uses less processing power than convolutional neural networks (CNNs) while still being able to accurately detect SOC environments.

4.2.3 The Role of Threat Intelligence Integration

Adding real-time threat intelligence from APIs like VirusTotal and PhishTank makes systems much more reliable by giving them real-world reputation data for decoded URLs. This integration cuts down on false positives and makes sure that ImageAware+ stays in sync with the changing world of phishing.

ImageAware+ is in line with industry trends towards intelligence-driven security automation (ENISA, 2024) because it uses heuristic analysis, threat intelligence, and outputs that can be explained.

4.2.4 Explainability and Analyst-Centric Design

A common topic in cybersecurity research is how hard it is to trust automated systems. ImageAware+ solves this by being open about its reporting: every result comes with annotated images and an explanation of how the detection was made. This design choice makes sure that analysts are always aware of what's going on and trust the system's output, which meets both academic and operational goals.

4.3 Evaluation of Design Choices

4.3.1 Technology Selection

The choice to use Python 3.x as the main programming language was based on the fact that it has a lot of libraries and is widely used in cybersecurity research. OpenCV is a strong

platform for processing images, and Tesseract OCR is a reliable, open-source tool for getting text out of images. PyZbar decodes QR codes quickly and easily without using a lot of processing power.
VirusTotal and PhishTank were chosen for data validation because they are reliable, have a lot of community support, and are well-known in the security research community. These tools all work towards the project's goals of being open, easy to use, and repeatable.

4.3.2 Algorithmic Approach

A hybrid, heuristic-driven approach was selected instead of solely machine-learning-based models. Deep learning can be more accurate in some cases, but it is often not clear or easy to understand (Liu et al., 2023). ImageAware+ uses deterministic algorithms that let analysts follow and recreate decisions, which is important for compliance and auditability in SOCs. The modular design of the pipeline makes it easier to scale and maintain. You can update each stage—preprocessing, OCR, QR decoding, and threat validation—separately as new technologies come out.

4.4 Research Limitations

Every system has its flaws, and it's important to be honest about them to keep academic integrity. The primary limitations recognised in this study encompass:

1. Limitations on the dataset:
   There aren't many labelled datasets for image-based phishing yet. Public datasets like PhishTank give you URLs but not always the phishing images that go with them. This makes it harder to do a quantitative evaluation.

2. Language and Context Dependence:
   Right now, OCR performance and heuristic keyword matching depend on data in English. Multilingual phishing campaigns might not be found until more language models are added.

3. Relying on APIs from other companies:
   Using outside services like VirusTotal can lead to problems, such as rate limits, API key restrictions, or data latency.

4. Image Resistance to Adversarial Attacks:
   As adversarial attacks get better, noise or distortions could be used to change images so that OCR and QR decoding can't read them. More research is needed on adversarial robustness.

5. Performance Scalability:
   Real-time enterprise-level integration would need to be optimised and possibly containerised (e.g., Docker/Kubernetes) for horizontal scaling, but it would work for small to medium-sized deployments.

Even with these limitations, the ImageAware+ framework gives researchers a basic, flexible platform that can be used to answer future questions about explainable image-based phishing detection.

4.5 Future Research and Development

Future research directions for ImageAware+ are diverse, encompassing both technological progress and ethical development in the field of cybersecurity.

1. Integration of Deep Learning:
   Convolutional Neural Networks (CNNs) or Vision Transformers (ViTs) could be added to future versions to automatically find brand logos, login screens, and layout similarities between phishing and real sites.

2. Improvement of multilingual OCR and NLP:
   Adding support for non-Latin scripts like Cyrillic, Arabic, and Chinese to OCR would make it easier to catch phishing attempts around the world. Combining OCR output with NLP models could make it easier to find social-engineering keywords.

3. Testing for robustness against adversaries:
   Looking into adversarial image detection methods like feature squeezing and gradient masking could make systems more resistant to attempts to get around them (Goodfellow et al., 2015).

4. Making a full dataset:
   Creating a standardised, open-source dataset of phishing images would be a big help for this area of research to be able to be repeated. Working with security companies and universities could make this possible.

5. Complete SOC Integration and Automation:
   Adding ImageAware+ as a RESTful microservice to a SIEM environment would make it possible to automate the ingestion and response processes. Integration with tools like TheHive, MISP, or Splunk could make things even easier.

6. Growth of Explainable AI (XAI):
   Using interpretability frameworks like LIME or SHAP could help analysts figure out which features (text, visual cues, QR data) had the biggest impact on a detection decision, which would make them more trusting.

7. Generative Adversarial Evaluation of Visual Phishing Defences: Recent research, including PEEK (Phishing Evolution Framework), demonstrates that extensive language models can systematically produce varied phishing campaigns and analyse the evolution of attack patterns over time (Chen et al., 2025). To apply this concept to the visual realm, one would utilise generative models to produce synthetic phishing images and email templates, subsequently subjecting ImageAware+ to rigorous testing against these developing variants. This method would help test how strong something is against realistic, changing enemies and fit with bigger predictions about where phishing attacks will go in the future (Osamor et al., 2025).

4.6 Final Thoughts

This research confirms that automated visual phishing detection systems are useful and necessary for modern cybersecurity operations. ImageAware+ shows that you can get useful, easy-to-understand results by combining computer vision, OCR, and threat intelligence with little extra work.
There are still some problems to solve, especially when it comes to expanding the dataset, supporting multiple languages, and being able to handle attacks, but the system is a good starting point for both academic research and industrial use. Most importantly, it reinforces the idea that cybersecurity automation must always be explainable. This keeps human analysts informed, empowered, and in charge of making decisions.

5.0 Appendix - Detailed Descriptions of Algorithms

This appendix gives detail about how the ImageAware+ system's modules work.The next few sections go into more detail about the processing pipeline, data flows, and ways of doing maths that were used.

5.1 Algorithm 1 – Image Preprocessing and Normalization

Objective:
To prepare raw phishing images for accurate text and QR decoding by applying normalization, noise reduction, and segmentation techniques.

Rationale:
OCR and barcode recognition algorithms are highly sensitive to noise, compression artefacts, and variations in lighting or contrast. Preprocessing is therefore essential to enhance edge clarity and contrast before further analysis. Research by Zhang et al. (2022) indicates that preprocessing can improve OCR accuracy by up to 25%.

Inputs:

- Raw image file (.png, .jpg, .bmp)
- Optional metadata (file hash, source email ID)

Outputs:

- Enhanced, thresholded grayscale image ready for OCR and QR scanning

If the resulting image is too dark or light, the algorithm may:

- try Otsu’s thresholding as a fallback;
- apply histogram equalisation before re-thresholding;
- flag the sample for manual review if preprocessing fails.

5.2 Algorithm 2 – OCR Text Extraction

Objective:
To extract textual content from preprocessed phishing images for keyword analysis and threat-intelligence querying.

Inputs:

- Preprocessed image from Algorithm 1

Outputs:

- Normalised text string (lowercased, punctuation trimmed)
- Token list for downstream keyword analysis

Post-processing may include:

- removing non-printable characters;
- correcting common OCR errors;
- filtering extremely short or meaningless tokens.

5.3 Algorithm 3 – QR and 2D Barcode Decoding

Objective:
To detect and decode QR codes or barcodes embedded within images to recover potentially malicious URLs or payloads.

Inputs:

- Preprocessed image (or original if high quality)

Outputs:

- Decoded payload string (e.g., URL) or error indicator

If the QR region is partially obscured, the algorithm logs a decoding failure and marks the image for potential manual review.

5.4 Algorithm 4 – Threat-Intelligence Enrichment

Objective:
To validate URLs extracted from OCR and QR decoding using external threat-intelligence services.

Inputs:

- URL list derived from Algorithms 2 and 3

Outputs:

- Per-URL verdict (e.g., malicious / suspicious / benign)
- Aggregated confidence score

The combine_verdicts function may assign numeric scores (e.g., 0–100) based on whether each service flags the URL as malicious or suspicious.

5.5 Algorithm 5 – Heuristic Risk Scoring

Objective:
To derive an overall phishing risk score for an image by combining visual, textual, and threat-intelligence evidence.

Inputs:

- Keyword-based scores from OCR
- Presence/absence of QR code and payload
- Threat-intelligence verdicts
- Optional visual heuristics (e.g., login prompt presence)

Outputs:

- Final numeric risk score (0–100)
- Risk band classification (e.g., Low, Medium, High)

Weights can be tuned based on empirical results or SOC analyst feedback.

5.6 Algorithm 6 – Report Generation and Explainable Output

Objective:
To generate a standardised, explainable phishing analysis report for SOC analysts.

Inputs:

- Original or preprocessed image

- Extracted text, decoded QR payloads
- Threat-intelligence results
- Final risk score and label

Outputs:

- JSON-formatted data for SIEM/SOAR ingestion
- Optional PDF report with annotated image regions

The report clearly documents:

- what was detected,
- where it was detected,
- which indicators contributed to the risk classification,
- when the analysis was performed.

5.7 Logging and Error Handling

All major operations generate logs including:

- timestamps;
- hashed identifiers;
- error codes (e.g., OCR failure, TI timeout);
- override actions taken by analysts.

Logs support:

- forensic investigations;
- performance tuning;
- compliance with audit requirements (e.g., ISO 27001).

5.8 Future Algorithmic Enhancements

Potential future enhancements include:

- deep-learning-based logo and layout matching;
- Vision Transformer models for page-level similarity;
- adversarial perturbation detection;
- multilingual OCR integration;
- model-based anomaly detection on image artefacts.

- These align with the future research directions identified in Section 4.5.

6.0 Bibliography

Abdelnabi, S., Möller, T., & Fritz, M. (2020). VisualPhishNet: Phishing website detection by visual similarity. In Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2020) (pp. 159–178). Springer. https://doi.org/10.1007/978-3-030-52683-2_8

Basit, A., Zafar, M., Qureshi, K. N., & Khan, A. (2021). A comprehensive survey of phishing detection techniques based on machine learning. IEEE Access, 9, 123288–123310. https://doi.org/10.1109/ACCESS.2021.3109223

Dhamija, R., Tygar, J. D., & Hearst, M. (2006). Why phishing works. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 581–590). ACM. https://doi.org/10.1145/1124772.1124861

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.

Goodfellow, I., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In International Conference on Learning Representations (ICLR 2015). https://arxiv.org/abs/1412.6572

Lin, H., Liu, J., Ma, J., Lin, Z., Feng, X., & Xu, G. (2022). Phishpedia: A hybrid deep-learning approach to visually identify phishing webpages. Network and Distributed System Security Symposium (NDSS 2022). https://doi.org/10.14722/ndss.2022.23061

Liu, H., Zhang, Y., Lin, H., Feng, X., & Xu, G. (2023). PhishIntention: Phishing website detection via intention understanding. USENIX Security Symposium 2023. https://www.usenix.org/conference/usenixsecurity23/presentation/liu

Shahriar, H., & Zulkernine, M. (2012). Trustworthiness testing of phishing websites: A behavior model-based approach. Future Generation Computer Systems, 28(8), 1258–1271. https://doi.org/10.1016/j.future.2012.02.004

Smith, R. (2007). An overview of the Tesseract OCR engine. In Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) (pp. 629–633). IEEE. https://doi.org/10.1109/ICDAR.2007.4376991

PhishFirewall. (2024, September 10). *The evolution of phishing attacks*. In *The Phishing Playbook*. Retrieved November 28, 2025, from https://www.phishfirewall.com/phishing-playbook-chapters/the-evolution-of-phishing-attacks phishfirewall.com+1

The Evolution of Phishing and Future Directions: A Review

Osamor, J. C., Ashawa, M. A., Shahrabi, A., Philip, A., & Iwendi, C. (2025). The evolution of phishing and future directions: A review. *International Conference on Cyber Warfare and Security, 20*(1), 361–368. https://doi.org/10.34190/iccws.20.1.3366

PEEK: Phishing Evolution Framework (LLM-generated phishing)

Chen, F., Wu, T., Nguyen, V., Wang, S., Abuadbba, A., Rudolph, C., & others. (2025). *PEEK: Phishing evolution framework for phishing generation and evolving pattern analysis using large language models* (arXiv preprint arXiv:2411.11389). https://arxiv.org/abs/2411.11389

VISUA. (n.d.). *Visual phishing detection for anti-phishing platforms and providers*. Retrieved [access date], from https://visua.com/use-case/anti-phishing-detection-with-visual-ai

Visual-Similarity-Based Phishing Detection

Medvet, E., Kirda, E., & Kruegel, C. (2008). Visual similarity-based phishing detection. In *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks (SecureComm 2008)*. https://doi.org/10.1145/1460877.1460905

Barracuda Networks. (2024, October 22). Threat spotlight: The evolving use of QR codes in phishing attacks. https://www.barracuda.com

Check Point Research. (2024). QR code phishing (quishing) attacks surge: Threat intelligence report 2024. Check Point Software Technologies. https://research.checkpoint.com

CISA. (2024). Reducing phishing-related risk in security operations centres. Cybersecurity and Infrastructure Security Agency. https://www.cisa.gov

ENISA. (2024). ENISA Threat Landscape 2024. European Union Agency for Cybersecurity. https://www.enisa.europa.eu

European Commission. (2018). General Data Protection Regulation (GDPR). Official Journal of the European Union.

European Commission. (2021). Ethics guidelines for trustworthy AI. Publications Office of the EU.

IBM Security. (2024). Cost of a Data Breach Report 2024. IBM Corporation. https://www.ibm.com/security/data-breach

National Cyber Security Centre. (2023). Cyber security and resilience: Guidance and principles. https://www.ncsc.gov.uk

Proofpoint. (2023). QR code phishing: The rise of “quishing” in enterprise environments. Proofpoint Threat Research. https://www.proofpoint.com

Trend Micro. (2024). Quishing: The rise of QR code phishing in enterprise attacks. Trend Micro Research. https://www.trendmicro.com

VirusTotal. (2024). VirusTotal API v3 documentation. Google Cloud. https://docs.virustotal.com

Google. (2024). Tesseract OCR engine documentation. https://tesseract-ocr.github.io

OpenCV.org. (2024). OpenCV documentation. https://docs.opencv.org

PyZbar Developers. (2023). PyZbar: Barcode & QR code reader for Python. https://github.com/NaturalHistoryMuseum/pyzbar

FastAPI. (2024). FastAPI framework documentation. https://fastapi.tiangolo.com

Flask. (2024). Flask documentation. https://flask.palletsprojects.com

Kahneman, D. (2011). *Thinking, fast and slow*. Farrar, Straus and Giroux.

Jakobsson, M., & Myers, S. (Eds.). (2006). *Phishing and countermeasures*. Wiley.

Osamor, V., Akinosho, O., Ajayi, T., & Oladipo, O. (2025). *The evolution of phishing and future directions*. ResearchGate.

Medvet, E., Kirda, E., & Kruegel, C. (2008). Visual-similarity-based phishing detection. *SecureComm*.

VISUA. (2024). *Anti-phishing detection with visual AI*. https://visua.com/use-case/anti-phishing-detection-with-visual-ai

Xu, W., Evans, D., & Qi, Y. (2017). Feature squeezing. *arXiv*.

Eykholt, K., et al. (2018). Physical-world attacks on visual classification. *CVPR*.

Le Pochat, V., et al. (2019). *A large-scale evaluation of public URL blacklists*. NDSS.

SecureState. (2024). *King Phisher documentation*. https://github.com/securestate/king-phisher