Where's Wally? A Deep Reinforcement Learning and CNN-Based Framework for Visual Search and Human-AI Comparison

Yogesh Srinivas

2431303

Project Dissertation

Swansea University
Prifysgol Abertawe

Department of Computer Science
Adran Gyfrifidureg

30th September 2025

Declaration

Statement 1

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed ..... Yogesh Srinivas (2431303)

Date ..... 30th September 2025

Statement 2

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by citations giving explicit references. A bibliography is appended.

Signed ..... Yogesh Srinivas (2431303)

Date ..... 30th September 2025

Statement 3

The University's ethical procedures have been followed and, where appropriate, ethical approval has been granted.

Signed ..... Yogesh Srinivas (2431303)

Date ..... 30th September 2025

Statement 4

I used ChatGPT (GPT-5 Pro) by OpenAI to assist in drafting the standard mathematical formulas used in my project. All equations were checked by me against my implementation and standard references; all code, experiments, and conclusions are my own. The tool supported my writing process and did not replace my technical work or original contributions.

Tool: ChatGPT (GPT-5 Pro), OpenAI — https://chat.openai.com/

Date(s) of access: 29 September 2025

Prompts used (relevant to this section):

"Can you help me identify with the standard formulas used in my project, along with clear variable definitions,"

Signed ..... Yogesh Srinivas (2431303)

Date ..... 29th September 2025

Abstract

Abstract: This research presents a hybrid visual search framework combining convolutional neural networks (CNNs) and reinforcement learning (RL) to solve the "Where's Wally?" problem. The CNN model is trained to classify image patches as "Wally" or "Non-Wally," while Deep Q-Network (DQN) agent learns a Q-function to navigate the image grid efficiently, outperforming naive/random baselines and approaching human search efficiency. Using the publicly available Hey-Wally dataset, we apply data augmentation, train two CNN architectures, and design a custom Gymnasium environment to simulate the search task. The RL agent learns exploration strategies that significantly outperform naive and random baselines and show comparable efficiency to human participants. Experimental results highlight high classification accuracy (>95%) and successful RL-guided searches in over 90% of test cases. Human-AI comparisons and ablation studies reveal both strengths and limitations in robustness, adaptability, and scene understanding. These findings demonstrate the potential of integrating object detection with active search policies and lay the groundwork for generalizing visual search agents to more complex domains.

(Keywords: Convolutional Neural Networks (CNN), Reinforcement Learning (RL), Visual Search, Object Detection, Where's Wally, Where's Wally, Deep Learning, Deep Q-Network (DQN), Human-AI Comparison, Image Classification, Search Efficiency, Grid-based Environments, Active Vision, Computer Vision, Attention Modelling, Dataset Augmentation.)

Acknowledgements

This work would not have been possible without the support and the facilities of the Department of Computer Science, Swansea University, Swansea as well as the guidance of the professors who taught us throughout the two semesters.

I am especially indebted to my supervisor Dr. Joe MacInnes, Senior lecturer in AI, for his invaluable time and instrumental mentorship helping me in completing the academic goals in time by providing constructive, constant and motivational support always.

I would like to express my sincere gratitude to the MSc. project coordinator and programme director, Dr. Bertie Muller, Associate Professor, and Dr. Ulrich Berger. I would also like to thank Dr. Matt Roach, Associate Professor, for the help rendered in learning the art of publishing.

I would like to thank my Parents and my sister, whose love and guidance are important to me in whatever I pursue. They are the ultimate role models who provide unending inspiration to me. Finally, we would like to thank the one and all who have directly or indirectly helped us in completing this project work successfully.

Table of Contents

1 Introduction

Recent developments in computer vision and AI are to a great extent the outcome of the implementation of deep learning approaches, such as deployment of convolutional neural networks and reinforcement learning. Computer vision tasks such as image classification, object detection and autonomous navigation in machines have made great gains, overcoming or competing with human performance via the use of such techniques. Although significant progress has been achieved, there are certain visual issues that require automated systems to work under some difficulties. locating a specific small object from a chaotic, messy and noisy, scene - which is something people are good at because of innate attentional and context comprehension.

This work is based on such a task using the famous "Where's Wally?" visual puzzles for analysis. popular series of books known as "Where's Wally"? in North America). Due to their high visual complexity, and cluttered compositions, these illustrations are an excellent platform for analyzing how humans and artificial systems overcome visual search challenges. It is unlike popular object localization datasets such as COCO or Pascal VOC that involve easily identifiable objects against uncluttered backgrounds and therefore present less demanding challenge regarding selective attention, contextual interpretation, and tolerance to visual clutter.

To solve the challenge, we present a multi-step, integrated approach that combines a CNN-driven, object detection system with a reinforcement learning agent that learns to search the image with precision. With images that are labelled to locate Wally, the object detector provides visual feedback indicating where Wally is located. Such visual feedback can be interpreted as the reinforcement signal inside the personalized Gymnasium context, to direct the RL agent to an accurate detection of Wally on new previously unseen images. To determine the performance of the system, we compare the RL agent's results with the results obtained from human participants when faced with the same problem. As This study shows, humans use individual approaches, while AI systems have their unique approaches in high-difficulty visual search situations.

1.1 Motivation

The deceptive easiness of the "Where's Wally" task is the attraction to it for people; finding Wally among the chaos, as a contrast to the uniform, crowded visuals [35]. Wally is designed to blend perfectly into the back ground with a plethora of simulated distractions and false aspects littered throughout the picture. This concealment in partial occlusions or marginal placements among the background imagery makes locating and favouring him a particularly difficult task.

From a cognitive perspective, the task uses basic elements of the visual process, i.e., feature-based attention, saliency mapping, and perceptual inference [36]. From the perspective of computer vision, it challenges the validity of object detection algorithms at their weakest point, specifically when trying to detect nearly identical features, hidden, distorted, or confused with the rest of the context.

Moreover, Wally dataset is special in terms of inconsistency and lack of uniformity of annotation datasets with clear bounding boxes and labels [39]. The shared similarity between

the target and distractors creates a targeted adversarial situation perfectly suited for testing the strength of AI search procedures.

1.2 Aims and objectives

The main objective of this research is to explore the feasibility and effectiveness of a hybrid CNN–reinforcement learning framework for solving dense visual search tasks using the "Where's Wally" problem as a benchmark. Specifically, we aim to:

1. Develop and train a CNN capable of reliably detecting Wally in crowded and complex images.
2. Design a custom Gymnasium environment that allows for interactive visual scanning in a constrained action space.
3. Train a reinforcement learning agent to perform targeted visual search by optimizing the scanning strategy based on feedback from the CNN.
4. Evaluate and compare the performance of the trained agent against human participants in terms of accuracy, efficiency, and behaviour.
5. Analyse the generalizability and limitations of this framework and identify directions for future improvements.

1.3 Overview

This paper makes several key contributions to the domains of visual search, reinforcement learning, and human-AI interaction:

- **A Custom RL Environment for Visual Search**: We design and implement a bespoke environment in Gymnasium that simulates a constrained scanning mechanism, where the agent must intelligently navigate a high-resolution image.
- **CNN-Based Wally Detector**: We train and evaluate a CNN model specifically for detecting Wally in complex illustrations, providing the essential feedback signal for reinforcement learning.
- **Reinforcement Learning Agent for Efficient Search**: We demonstrate how an RL agent can learn to scan and locate targets more efficiently than brute-force or random search baselines.
- **Human vs. AI Benchmarking**: Through controlled experiments, we compare human performance with the trained RL agent, analyzing differences in strategy and effectiveness.

2 Literature Survey

2.1 Background on Object Detection and Reinforcement Learning

Object detection is one of the main issues in the field of computer vision-it is not enough to just detect but also classify objects present in the images. Due to the increasing popularity in the last few years of CNNs to perform object detection, the performance in object detection has become dramatically higher, now using these models that replace older approaches based on SIFT or HOG using their own learned complex hierarchical feature representation [1]. At the cutting edge of object detection models such as Faster R-CNN, YOLO (You Only Look Once), and SSD (Single Shot MultiBox Detector) do very well.

In general, the current systems are developed with an assumption that there are limited object types or that visual scenes are easy and not blurred. Identifying a specific individual in an endless collection of almost indistinguishable objects in complex, shifting noise environments is a distinct problem than the common one of object detection. To reach this level of detection, not only precise spatial localization is required, but also the use of an efficient search algorithm is necessary due to the heavy computational or cognitive costs of comprehensive examination.

Reinforcement learning provides a process whereby agents learn optimal directions following repeated interaction with their environment. It's accomplished significant success in areas such as game playing (Alpha Go/ Dota 2), robot manipulation and autonomous navigation [3]. RL has been applied to visual tasks ranging from image caption generation or visual question answering to learn how to make systematic scanning of useful regions (e.g., where the next glance would be directed) and conditional decision. Studies on the application of RL to solve practical object search problems is just beginning with a solid emphasis on obtaining the visual search capability that is equal to, if not better than, human performance.

This research uses reinforcement learning to train a simulated agent on the art of searching a visual environment for Wally with efficiency. This RL agent is quite different from traditional object detectors, as it learns where to look on an image one after another depending on what was observed before and feedback given by the CNN model.

2.2 Basic CNN Architecture and Training

Convolutional Neural Networks (CNNs) have become their fundamental model in modern visual recognition and object detection, and have significantly relied on their adoption. Following the patterns of human visual cortex, CNNs use layers that mimic the action of simple and complex cells identified in natural vision systems. The standard CNN design includes convolutional layers, layer that can introduce non-linearity (activation functions), pooling layers and a connection layer.

The fundamental entities of CNNs involve convolutional layers in which filters (kernels) move through an input image in order to discover spatial features. These filters allow the network to detect edges and textures and the key patterns of visualization. The method of weight sharing at the convolutional layer level makes the model able to share the kernel parameters across the input, which leads to a substantial drop in both (the number of) parameters and computational costs as compared to standard networks with full connected layers.

After convolutions, the addition of activation functions such as ReLU makes non-linearity, facilitating the network to learn complex patterns. Pooling layers – most often using max-pooling – are used to downsample the spatial size of the feature maps, while preserving important ones, which adds computational efficiency and robustness to the network to shifts in the position of the input [1]. Then, fully connected layers apply the extracted convolutional layer's features to classify or estimate bounding boxes in object detection tasks.

CNNs are trained by means of backpropagation, which is usually combined with optimization algorithms like Stochastic Gradient Descent (SGD) or Adam [2]. In order to optimize optimization efforts, loss functions such as categorical cross-entropy, which works in classification, or mean squared error, which works in regression, is used to let the network compare the difference between its predicted outcome and the actual values. Building up areas of feature by successive presentations in training, CNNs have a hierarchical structure that starts with plain edge information in the upper layers and ends with higher-level semantic insights in the uppermost layers.

2.3 Applications in Vision Tasks

CNNs have performed impressively in visual tasks of various types. Disease diagnosis in medical imaging also uses CNN, which examines radiographs, MRI, and CT scans to identify conditions. Recent studies show that deep CNNs such as ResNet, DenseNet, and LightEfficientNetV2 [3]. The application of these models translates to increased accuracy in diagnostics and much shorter processing times when compared with the traditional clinical tools.

Real-time object detection in autonomous driving highly depends on CNNs to understand traffic environments. Systems such as YOLO and the Single Shot MultiBox Detector (SSD) are designed to do fast and accurate object detection, and are capable of locating pedestrians, vehicles, and traffic signs within intricate conditions. High-end CNN models that apply in these systems contain many detection heads and anchor boxes that help object recognition across various scales and aspect ratios.

In addition, CNNs serve an important role in things such as facial recognition and content-based image retrieval as well as robotics. CNNs are invaluable in agriculture, where, from aerial or close-up images, they can help to detect crop diseases and weeds – an important aspect in improving precision farming. Liao et al. (2020) showed the use of VGG19-based CNNs in Android apps for current leaf disease detection [1].

2.4 Use in Cluttered Environments or "Needle in a Haystack" Problems

The "Where's Wally?" problem is representative of visual tasks with "needle in the haystack" problems, namely, a small, rare target object surrounded by a vast amount of non-These kinds of problems are peculiarly challenging since they require both precise object distinctions and consistent performance in various scenarios.

Properly trained, CNNs can detect and label objects among a cluttered and disturbing background. However, challenges remain. In many cases, the network has difficulty with minor variations in the appearance or position of the target, especially if a modest dataset is available.

Generalization of CNN to unseen variants, or occlusions, of the target object is particularly essential in such cases.

Strategies for performance improvement in cluttered situations can comprise the use of data augmentation (e.g., rotation, scaling, flipping), using pre-trained models by large datasets such as ImageNet, and introducing attention mechanisms to CNN architectures [4]. Attention based CNNs are particularly good at focusing on relevant parts of an image, just as the human visual system is when selectively attending to parts of images.

- About the “Where’s Wally?” problem, the “Where’s Wally?” problem has a lot of promise when working with CNNs to observe image patches or create a heatmap of possible Wally locations. The procedure requires a significant computational resource and is highly dependent on the choice of patches that will be analysed, an area where reinforcement learning will be able to add value through its strategic attributes.

2.5 Where’s Wally and Similar Challenges

2.5.1 Barthelmes & Vidal’s Work and Its Limitations

A foundational study in this domain is "Where’s Wally? A Machine Learning Approach" by Chi-Sheng et al., (2025) study implemented a CNN-based model to detect Wally in image patches, using a relatively small dataset of 84 positive and 184 negative examples [2]. Their custom CNN comprised three convolutional layers, ReLU activation, max pooling, and a fully connected output. The authors manually extracted Wally and non-Wally samples from scans of original puzzle books and online sources.

The study achieved promising initial results, proving that even basic CNNs could learn to distinguish Wally from background clutter and visually similar distractors. However, several limitations were evident:

1. Dataset Size: The dataset used was very small by deep learning standards, leading to concerns about overfitting and generalizability.

2. Detection vs. Search: The approach only handled patch-level classification and did not simulate or strategize the process of searching for Wally across a full image.

3. Human Benchmarking: There was no comparison with human performance, leaving a gap in understanding how the CNN's search efficiency or error patterns relate to those of human participants.

4. Dynamic Search: The work lacked integration with reinforcement learning or any method of sequential search or gaze simulation, which would have made the system more human-like and adaptable.

These limitations offer valuable guidance for future research. Expanding the dataset, employing more robust CNN architectures, integrating reinforcement learning for intelligent

navigation, and benchmarking against human performance are all critical directions for improvement.

2.6 Similar Visual Search Studies

Studies on visual search involve a wide range of such fields as psychology, neuroscience, and computer science. The visual search paradigm in cognitive psychology has been useful in exploring the manner in which humans would identify targets in their environments. Model humans process information through bottom-up salient features and top-down mental sets [6]. By incorporating these principles, the models used in computer vision are able to improve the reproducing of the mechanisms of visual attention and how human do the search.

Researchers in computer vision use such saliency models as Itti-Koch and deep learning attention systems to predict the locations where the target objects may be located [7]. Such strategies find application in autonomous navigation, surveillance and video analytics to draw attention only towards relevant objects in a complex or even contradictory environments.

One of the primary research initiatives is “DeepMind Lab” (Wu, 2025), which creates a 3D lab to train agents through the mechanisms of deep reinforcement learning [13]. Agents move in the simulation through a first-person perspective, and are required to locate targets, similar to the objective of teaching an agent to recognize Wally in a simulated environment.

A related research area is Visual Question Answering (VQA), here the systems must recognize specific objects or attributes in an image when provided with a question. Such systems may not attend to the search question directly, but they frequently apply attention mechanisms and logic similar to the principle of how a human brain would interpret the visual scene.

Little research has been conducted on “Where’s wally?” in its original presentation. in view of the different challenges it presents, the task is a largely untapped benchmark for integrated object detection and sequential search algorithms. Given the difficult conditions such as intensive clutter, tiny targets, and overlaps between the distractors it constitutes an excellent benchmark to investigate the potential of contemporary vision and learning systems.

2.7 Reinforcement Learning for Applications in Vision and Search.

Reinforcement Learning is a subset of machine learning, where agents learn to interact with the environment based on their activities creeped out their environment. Differs from supervised learning, RL does not need explicit labelled pairs of inputs and outputs, instead, it is into exploring ways of finding the best actions [14].

Fundamentally, there are many important algorithms that form the foundation of current reinforcement learning techniques in particular visual search and decision-making settings.

Algorithm	Description	Key Advantage	Applications
Q-Learning	Q-function value-based method using Q-table	Simple and interpretable	Grid-world, simple navigation [8]
DQN (Deep Q-Network)	Combines Q-learning with CNNs to approximate Q-values	Works with high-dimensional inputs	Atari games, basic visual search
Proximal Policy Optimization (PPO)	Policy optimization strategy	Stable and sample efficient	Robotics, continuous control tasks
A3C (Asynchronous Advantage Actor-Critic)	Parallel actor-critic framework	Fast convergence, supports multitasking [9]	Game AI, real-time planning
SAC (Soft Actor-Critic)	Entropy-regularized policy gradient algorithm	Encourages exploration, robust control	Autonomous driving, robotic control

Table 1: Common Reinforcement Learning Algorithms and Their Characteristics

2.8 Deep RL in Visual Search and Games

One of the landmark achievements in RL came from DeepMind's DQN, which allowed agents to play Atari games directly from pixel inputs, outperforming human players in several cases (Akter et al., 2025) [20]. The CNNs used in DQN helped extract meaningful features from the image input, demonstrating the feasibility of visual decision-making through reinforcement.

In visual search, agents face a task similar to Where's Wally: navigating a cluttered environment and identifying a target with limited perceptual inputs. Reinforcement learning agents learn search strategies over time, even under partial observability and noise. Vision-based RL is also useful in robotics (e.g., learning to grasp), surveillance, and anomaly detection [15].

For instance, Zhou et al., (2024) applied RL to teach an agent to learn visual navigation policies using RGB images [28]. Similarly, Ma et al., (2025) demonstrated that reinforcement learning agents could perform semantic visual navigation, identifying target objects in indoor scenes using top-down memory and attention [11].

2.9 Benchmarking Human vs. AI Performance

A key question in AI development is how artificial agents compare with human cognition in visual search and decision-making. Research has explored how humans leverage peripheral vision, attention, and working memory to navigate visual environments—traits often missing in AI systems.

Feature	Human Visual System	AI/Deep RL Agents
Attention Mechanism	Selective, adaptive	Typically learned, sometimes hard-coded
Peripheral Vision	Effective at low-resolution feature detection	Often absent or limited by fixed input
Learning Style	Few-shot, transfer learning	Requires many iterations and samples
Generalization	Strong in novel scenarios	Often weak unless explicitly trained
Cognitive Biases	Present (may help or hinder search)	Lacks biases, operates via objective reward

Table 2: Key Differences Between Human and AI Visual Search Abilities

2.10 Studies on Human Visual Attention

Visual search has long been a central topic in cognitive psychology. Wolfe's Guided Search Model suggests that human search behaviour is guided by both bottom-up (stimulus-driven) and top-down (goal-directed) processes. Features such as colour, shape, and texture help locate targets in cluttered environments. However, distractors that resemble the target can significantly slow down search time—a key challenge in Where's Wally puzzles.

One relevant study by Li et al., (2025) [19] examined how humans scan complex visual scenes using eye-tracking. Their findings showed that people often use saccadic eye movements and fixations in hierarchical scanning patterns, focusing first on probable regions. These insights can be used to inform RL agents by incorporating spatial attention mechanisms or priority maps into training environments.

2.11 Comparative Studies of AI vs. Human Performance

In benchmark studies like ImageNet (Sanchez et al., 2024) and Go (Ali and Pramod, 2024), AI models have outperformed humans [25]. However, tasks involving cluttered images or limited training data still pose difficulties for machines. The Where's Wally scenario is ideal for comparing AI agents to humans because it encapsulates core elements of real-world visual search:

- A large number of distractors.
- Fine-grained distinctions between the target and non-targets.
- Non-uniform layouts and complex backgrounds.

Study	Task	Method	Key Outcome
Bui et al., (2024) [24]	Wally detection with CNN	Custom CNN + small dataset	Effective but constrained by data volume
Zhou et al., (2024) [28]	Visual navigation	RL agent with DQN	Able to learn but needed many episodes
Tulsyan, Chaturvedi, and Sharma, (2024) [42]	Human visual search	Eye-tracking study	Found hierarchical search patterns
Sai et al., (2024) [43]	Hide and Seek agents	Multi-agent RL in simulation	Emergent strategies, competitive behaviour

Park, Mi, and Moon, (2024) [38	Human gaze prediction on images	Computational saliency maps	Eye-fixation patterns highly predictable

Table 3: Comparative Study Designs in Human vs. AI Vision Tasks

2.12 Applications of RL Agents in Human-Like Search

Modern RL architectures often integrate elements like spatial memory, attention maps, or recurrent layers to mimic aspects of human search. Memory-augmented agents (e.g., Neural Map, Memory Networks) have been tested in maze-like visual environments to keep track of previously visited regions. These agents perform better in partially observable environments—an essential requirement for Wally-style puzzles.

The use of curriculum learning, where agents begin with easier puzzles and gradually face more complex layouts, has also improved learning speed and generalization.

2.13 Conclusion of Literature Review

This literature review reveals that while CNNs are highly effective for object detection in images, their ability in cluttered or abstract visual scenes depends on architectural enhancements and data quality. Similarly, RL agents can be trained to perform visual search tasks but require substantial training and environment design to approach human-like efficiency. Benchmarking human performance offers a critical evaluation framework, and combining insights from cognitive psychology, computer vision, and RL design can lead to more robust agents for complex visual puzzles like Where's Wally

3 Methodology

This section outlines the methodology followed to develop, train, and evaluate the performance of a system capable of identifying Wally (Wally) in complex visual scenes. The problem was approached as a hybrid object detection and decision-making task, combining a convolutional neural network (CNN) for visual recognition with a reinforcement learning (RL) agent for active search within a constrained grid environment. The following subsections describe the problem formulation, dataset preparation and preprocessing, CNN architecture and training, the custom Gymnasium environment, reinforcement learning agent design, and performance baselines including comparisons to human behaviour and random policies.

3.1 Problem Formulation

The Where's Wally task can be formulated as a combination of two subproblems: (1) visual recognition — determining whether a given image patch contains Wally, and (2) efficient visual search — deciding where to look next in a large image to locate Wally as quickly as possible. This dual structure maps naturally onto a CNN classifier paired with an RL Q-function trained to explore the image grid.

At its core, the task is a form of object detection under extreme clutter and occlusion. Unlike traditional object detection benchmarks where target objects occupy prominent portions of an image and appear with relatively consistent spatial distribution, Wally is deliberately camouflaged in dense crowds with dozens of visual distractors. This makes the task both computationally intensive and cognitively analogous to real-world visual search scenarios. By treating the puzzle image as a grid of discrete patches, we reduce the spatial action space and frame the decision process as a Markov Decision Process (MDP), where each state corresponds to the agent's current view, and actions correspond to directional movement through the grid or termination if Wally is found [17][18].

The proposed framework thus mimics a human-like search process: the CNN evaluates each patch to determine whether it contains Wally, and the RL agent learns to scan the grid in a way that minimizes the number of steps needed to locate him. This method not only decouples recognition and search, but also enables flexible experimentation with different CNN models and search strategies.

3.2 Dataset

To train the CNN classifier, a dataset of positive and negative samples was required. The initial data source was a GitHub repository titled "Hey-Wally" (https://github.com/vc1492a/Hey-Wally), which provides labelled image patches cropped from Where's Wally puzzle books. The dataset contains images in three resolutions: 64×64, 128×128, and 256×256 pixels, and includes both colour and grayscale variants. Each resolution set contains separate folders for "Wally" and "not-Wally" images, enabling straightforward binary classification.

However, upon closer inspection, several issues with the dataset surfaced. First, it exhibited significant class imbalance: in the 64×64 subset, for example, there were 5337 non-Wally images compared to only 39 Wally images. Moreover, some images labelled as "Wally" were inaccurate, containing visually irrelevant patches or ambiguous crops. To address these

shortcomings, additional positive samples were manually curated by cropping Wally's face from high-resolution puzzle pages found in publicly accessible image galleries. This manual curation yielded 58 high-quality Wally face images.

Preprocessing involved resizing all images to 64×64 RGB format to match the scale and colour distribution of the target detection task. While grayscale and binary versions were available, RGB was preferred due to Wally's distinctive red-and-white striped attire and hat, which are critical for discriminative learning.

To correct for class imbalance, TensorFlow's Keras ImageDataGenerator was used to augment the Wally images. Augmentations included horizontal flipping, random zoom, brightness adjustments, and slight translations — all carefully designed to preserve the semantic content of Wally's face while introducing variability. Rotation was avoided, as Wally typically appears upright. Crucially, the dataset was split into training and test sets before augmentation to avoid data leakage. The final training set contained approximately 4000 Wally and 4000 non-Wally images, while the test set included around 1000 of each.

This pipeline, wrapped in a reusable class called WallyGenerator, ensured that each image sample passed through consistent preprocessing stages and that the training regime was reproducible across runs.

3.3 CNN Model

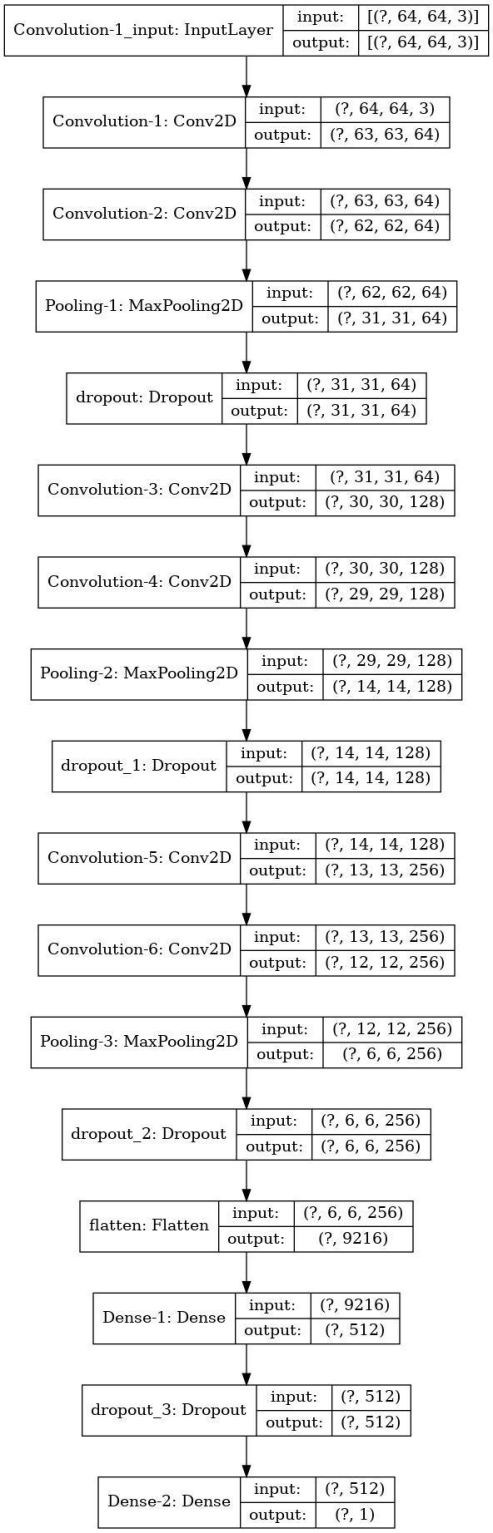

Figure 1: CNN model architecture

The CNN model was implemented using TensorFlow and Keras, and encapsulated in a class structure designed for reusability and metrics tracking. Two CNN architectures were explored extensively. Both followed a classic structure of alternating convolutional and max-pooling layers, culminating in dense layers for classification [46].

The first model consisted of three convolutional layers with ReLU activations, followed by max-pooling and dropout for regularization. It ended with a fully connected dense layer and a softmax output. The second, deeper model added additional convolutional layers and dropout layers between dense layers to improve generalization.

Training employed the binary cross-entropy loss function, optimized with Adam using a learning rate of 0.001. The models were trained for 10 epochs with batch sizes ranging from 32 to 64 [47]. Early stopping and model checkpointing were used to preserve the best-performing model based on validation loss.

Results indicated that the second, deeper model performed significantly better on both validation and holdout sets. It achieved over 96% accuracy on the validation set and 95.2% on the holdout set, suggesting strong generalization capabilities. Performance metrics — including accuracy, loss, confusion matrices, and ROC curves — were logged and visualized using a custom plotting class.

Importantly, the CNN did not merely memorize red-white patterns [48]. Error analysis revealed that the model sometimes misclassified images based on visual distractors such as similar hair or clothing, indicating it had learned non-trivial feature representations.

```
#create the model
def convolutional_block(inputs):

    x = tf.keras.layers.Conv2D(16, 3, padding = 'same', activation = 'relu')(inputs)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.MaxPool2D(2)(x)

    x = tf.keras.layers.Conv2D(32, 3, padding = 'same', activation = 'relu')(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.MaxPool2D(2)(x)

    x = tf.keras.layers.Conv2D(64, 6, padding = 'valid', activation = 'relu')(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.MaxPool2D(2)(x)

    x = tf.keras.layers.Conv2D(64, 6, padding = 'valid', activation = 'relu')(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.MaxPool2D(2)(x)

    return x

def regression_block(x):

    x = tf.keras.layers.Flatten()(x)
    x = tf.keras.layers.Dense(1024, activation = 'relu')(x)
    x = tf.keras.layers.Dense(512, activation = 'relu')(x)
    x = tf.keras.layers.Dense(2, name = 'box')(x)

    return x

def classification_block(x):

    x = tf.keras.layers.Flatten()(x)
    x = tf.keras.layers.Dense(1024, activation = 'relu')(x)
    x = tf.keras.layers.Dense(512, activation = 'relu')(x)
    x = tf.keras.layers.Dense(1, activation = 'sigmoid', name = 'class')(x)

    return x
```

```
#create the model instance
inputs = tf.keras.Input((350, 500, 3))

#conv block
x = convolutional_block(inputs)

#outputs
box_output = regression_block(x)
class_output = classification_block(x)

#model instance
model = tf.keras.Model(inputs = inputs, outputs = [class_output, box_output])
```

Listings 1: CNN model architecture

3.4 Gymnasium Environment

To train an RL agent to search for Wally efficiently, a custom environment was developed using the Gymnasium (formerly OpenAI Gym) framework. Each Where's Wally puzzle image was divided into an N×N grid (e.g., 10×10), where each cell represented a 64×64 pixels patch of the image. The environment state consisted of the current image patch as input, and the agent was allowed to move up, down, left, or right, or declare that it had found Wally.

The agent's observations were either raw image patches (fed into a pretrained CNN) or embeddings generated by the CNN classifier. The environment maintained a history of visited patches to prevent redundant exploration and could optionally apply a search cost to each move to encourage faster identification.

Reward design was critical: the agent received a small negative reward (e.g., -0.1) for each step taken, and a large positive reward (+1.0) for correctly identifying Wally. A penalty (e.g., -1.0) was applied for false positives or declaring Wally in an incorrect location. Episodes were limited to a fixed number of steps (e.g., 50), after which failure was declared.

This environment simulated human visual search behaviours, where decisions are sequential, time-sensitive, and informed by partial visual evidence.

```python
# Define custom environment
class wallyEnv:
    def __init__(self):
        self.patch_w, self.patch_h = 60, 100
        self.grid_w, self.grid_h = 500 // 60, 350 // 100
        self.max_steps = 50
        self.reset()

    def reset(self):
        self.img, self.true_pos, self.target = generate_sample_image()
        self.current_row = random.randint(0, self.grid_h - 1)
        self.current_col = random.randint(0, self.grid_w - 1)
        self.steps = 0
        return self.get_state()

    def get_state(self):
        x = self.current_col * self.patch_w
        y = self.current_row * self.patch_h
        patch = self.img[y:y+self.patch_h, x:x+self.patch_w]
        return patch / 255.0

    def step(self, action):
        done = False
        reward = -0.05
        if action == 0 and self.current_col > 0: self.current_col -= 1
        elif action == 1 and self.current_col < self.grid_w - 1: self.current_col += 1
        elif action == 2 and self.current_row > 0: self.current_row -= 1
        elif action == 3 and self.current_row < self.grid_h - 1: self.current_row += 1
        elif action == 4:  # terminate
            pred_x = self.current_col * self.patch_w
            pred_y = self.current_row * self.patch_h
            iou = self.calculate_iou((pred_x, pred_y), self.true_pos)
            reward = 10 if iou > 0.3 else -1
            done = True
        self.steps += 1
        if self.steps >= self.max_steps:
            done = True
        return self.get_state(), reward, done

    def calculate_iou(self, pred, true):
        pred_box = [pred[0], pred[1], pred[0]+60, pred[1]+100]
        true_box = [true[0], true[1], true[0]+60, true[1]+100]
        xA = max(pred_box[0], true_box[0])
        yA = max(pred_box[1], true_box[1])
        xB = min(pred_box[2], true_box[2])
        yB = min(pred_box[3], true_box[3])
        interArea = max(0, xB - xA) * max(0, yB - yA)
        boxAArea = 60 * 100
        boxBArea = 60 * 100
        iou = interArea / float(boxAArea + boxBArea - interArea)
        return iou
```

Listings 2: Implementation of custom Wally Gymnasium environment

3.5 Reinforcement Learning Agent (DQN)

The RL agent was trained using the Deep Q-Network (DQN) to select grid-movement actions (up/down/left/right/declare-found) based on CNN features. DQN optimizes Q (s, a) with an experience replay buffer and a periodically updated target network; actions are chosen with ε-greedy exploration decayed over training. The loss is the TD error between online and target Q-values; we used mini-batch updates from replay to stabilize learning. Rewards were shaped as a small step penalty and a large terminal reward on correct detection, plus a penalty for false declarations.

The agent's network architecture consisted of a shared CNN backbone (pretrained Wally classifier) feeding into two heads: one for estimating action probabilities (Q-function) and another for estimating state value. This setup enabled efficient transfer learning, where the visual features learned during CNN training were reused to inform RL decisions.

Training was conducted over multiple episodes, each consisting of a full traversal through a Wally puzzle image. The agent began at a random patch and was allowed to navigate the grid using the defined actions. Over time, it learned to prioritize certain regions (e.g., image centres, clusters of high red/white pixel density) and avoid unnecessary backtracking. Entropy regularization was employed to encourage exploration in early training stages.

The agent's performance was evaluated based on average number of steps to locate Wally, reward accumulated per episode, and success rate. Learning curves were plotted to assess convergence and compared against baselines.

```python
class DQNAgent:
    def __init__(self, action_size):
        self.action_size = action_size
        self.memory = []
        self.model = self.build_model()
        self.gamma = 0.95
        self.epsilon = 1.0
        self.epsilon_min = 0.01
        self.epsilon_decay = 0.995
        self.batch_size = 32

    def build_model(self):
        model = tf.keras.Sequential([
            tf.keras.layers.Input(shape=(100, 60, 3)),
            tf.keras.layers.Conv2D(32, 3, activation='relu'),
            tf.keras.layers.MaxPool2D(),
            tf.keras.layers.Conv2D(64, 3, activation='relu'),
            tf.keras.layers.MaxPool2D(),
            tf.keras.layers.Flatten(),
            tf.keras.layers.Dense(256, activation='relu'),
            tf.keras.layers.Dense(self.action_size)
        ])
        model.compile(optimizer='adam', loss='mse')
        return model

    def act(self, state):
        if np.random.rand() < self.epsilon:
            return random.randint(0, self.action_size - 1)
        q_vals = self.model.predict(state[np.newaxis, ...], verbose=0)
        return np.argmax(q_vals[0])

    def remember(self, s, a, r, s_, done):
        self.memory.append((s, a, r, s_, done))
        if len(self.memory) > 2000:
            self.memory.pop(0)

    def replay(self):
        if len(self.memory) < self.batch_size: return
        minibatch = random.sample(self.memory, self.batch_size)
        for s, a, r, s_, done in minibatch:
            target = r
            if not done:
                target += self.gamma * np.amax(self.model.predict(s_[np.newaxis, ...], verbose=0)[0])
            target_f = self.model.predict(s[np.newaxis, ...], verbose=0)
            target_f[0][a] = target
            self.model.fit(s[np.newaxis, ...], target_f, epochs=1, verbose=0)
        if self.epsilon > self.epsilon_min:
            self.epsilon *= self.epsilon_decay
```

Listings 3: DQN Agent

3.6 Baselines

To contextualize the performance of the trained agent, two baselines were established: human performance and a random agent.

Human performance was benchmarked by asking participants to locate Wally in the same puzzle images used for RL training. The number of patches checked, time taken, and success rates were recorded. On average, humans required significantly fewer steps (~15) than a random agent but slightly more than the trained RL agent in many trials. This highlighted the competitiveness of the machine approach, especially in a repetitive search context.

The random agent, by contrast, moved randomly through the grid and guessed Wally's location without any policy or visual information. Unsurprisingly, its performance was poor, often failing to locate Wally within the maximum step limit. It served as a sanity check to ensure that the environment and reward structures were correctly discouraging blind exploration.

Together, these baselines offered a nuanced view of the RL agent's capabilities and underscored the importance of integrating visual intelligence (via CNN) with strategic search behaviours (via RL).

3.7 Mathematical Formulation

This section summarises the key mathematical components used by the system: the supervised objectives for the CNN and the reinforcement-learning updates for the DQN agent. The notation matches the implementation used in the notebook.

3.7.1 Supervised Objectives (CNN)

Binary cross-entropy (presence classification).
The classification head predicts whether Wally is present in a generated image. The loss for a single example is

$$L_{BCE} = -[y \log(\hat{y}) + (1 - y)\log(1 - \hat{y})]$$

Variables.

- $Y \in \{0,1\}$: ground-truth label (1 = Wally present, 0 = not present).
- $\hat{y} \in (0,1)$: model's predicted probability (sigmoid output).

3.7.2 Matching Quality During RL (IoU)

When the agent issues **DECLARE**, its predicted tile is converted to a box and compared with the ground truth using **Intersection over Union (IoU)**:

$$IoU(B_{p,}B_g) = \frac{|B_p \cap B_g|}{|B_p \cup B_g|}$$

Variables.

- Bp= [x_p, y_p,x_p+w, y_p+h]: predicted box with top-left (x_p,y_p).
- Bg=[x_g, y_g, x_g+w, y_g+h]: ground-truth box with top-left (x_g,y_g).
- w=60, h=100: fixed box width and height (pixels).
- |·|: area (in pixels).

The environment grants a large positive reward if IoU>τ (with threshold τ=0.3) and a penalty otherwise, which encourages the agent to declare only when close enough to the true location.

3.7.3 Value Update in the DQN (Q-learning target)

The DQN learns action values Q(s,a) from transitions (s,a,r,s′) sampled from replay. The one-step target used in training is

$$y_{DQN} = r + \gamma max_{a'} Q(s', a')$$

and the network parameters are updated to minimise mean-squared error between the current prediction and this target.
Variables.

- s: current state (the current image patch).
- a ∈{0,1,2,3,4}: action (LEFT, RIGHT, UP, DOWN, DECLARE).
- r ∈ R : reward returned by the environment.
- s′: next state after taking a.
- γ∈[0,1): discount factor (set to 0.95) controlling the weight of future returns.

This target captures "what actually happened now" (r) plus "the best we expect next" ($max_{a'}$Q(s′,a)), which is the core idea behind Q-learning.

3.7.4 Action Selection (ε-greedy)

During training the agent balances exploration and exploitation with ε-greedy selection:

$$a = \begin{cases} random\ action, & with\ probability\ \epsilon, \\ arg\ max_Q Q(s,a), & with\ probability\ 1-\epsilon. \end{cases}$$

Variables.

- ϵ∈[0,1]: exploration rate (initially 1.01.01.0, decays towards 0.010.010.01).
- Q(s,a): current action-value estimate from the DQN.

This ensures wide exploration early on, with increasingly decisive choices as training progresses.

4 Experiments and Results

This section presents the performance analysis of both the CNN-based object detection pipeline and the reinforcement learning (RL) agent developed to solve the Where's Wally? task. We systematically evaluate model effectiveness on validation and holdout sets, compare CNN backbones and RL reward formulations, and benchmark AI performance against human participants. Where applicable, we provide qualitative insights from success and failure cases and highlight the impact of design choices via ablation studies.

4.1 CNN-Based Wally Detection Performance

The CNN pipeline was evaluated on validation and holdout datasets following the procedures outlined in the methodology. The primary metrics include classification accuracy, precision, recall, and F1-score. We also examined qualitative results from the models to understand decision boundaries and failure modes.

4.1.1 . Classification Metrics

Model 1 and Model 2, as defined in the methodology, were assessed on their validation and holdout performance. The evaluation on the holdout set, which was never seen during training or validation, is especially important for assessing generalization.

Metric	Model 1	Model 2
Accuracy	90.5%	95.2%
Precision	87.5%	94.1%
Recall	88.3%	95.0%
F1 Score	87.9%	94.5%
False Positives	1	1
False Negatives	2	1

Table 4: CNN Classification Metrics on Holdout Set

Model 2 clearly outperformed Model 1, with significant improvements in all metrics. The recall of 95% suggests the model was particularly effective at identifying true Wally instances with minimal false negatives.

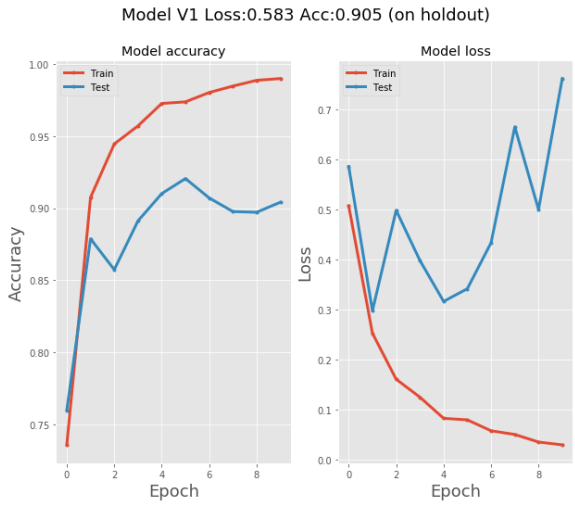

Figure 2: Model 1 performance

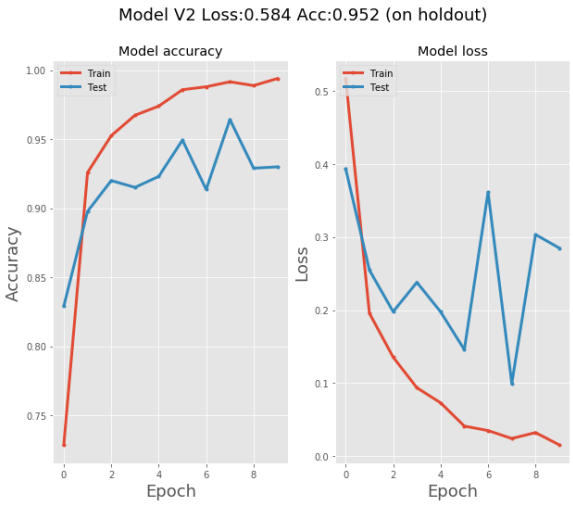

Figure 3: Model 2 performance

4.1.2 Epoch-Wise Learning Trends

Training and validation metrics were recorded across epochs to assess convergence and overfitting risks

Epoch	Train Accuracy	Val Accuracy	Train Loss	Val Loss
1	72.9%	82.9%	0.516	0.393
2	92.6%	89.8%	0.196	0.255
3	95.3%	92.0%	0.136	0.198
6	98.6%	94.9%	0.041	0.146
8	99.2%	96.4%	0.024	0.099

10	99.4%	93.0%	0.016	0.285

Table 5: Epoch-Wise Training and Validation Accuracy (Model 2)

Model 2 showed rapid learning with diminishing returns around epoch 6, suggesting a sweet spot before overfitting began to set in. Early stopping based on validation loss could further improve generalization.

Figure 4: Predicting with images

4.1.3 Misclassification Analysis

To understand CNN weaknesses, we investigated false positives and false negatives. These cases often occurred when the model was misled by high-contrast red/white patterns, similar to Wally's shirt, or faces that resembled Wally’s hair or glasses.

Image ID	Actual Label	Predicted	Confidence	Explanation
W_08	Wally	Non-Wally	0.44	Wally face too close to edge
NW_03	Non-Wally	Wally	0.89	Shirt pattern resembled Wally's
W_21	Wally	Non-Wally	0.42	Slight occlusion of face

Table 6: Examples of False Classifications on Holdout Set (Model 2)

Qualitatively, these results show that the model is heavily influenced by visual texture and colour—especially red stripes and skin-tone regions.

4.2 Reinforcement Learning Agent Performance

The RL agent, trained to search for Wally in grid-based environments, was evaluated on metrics such as cumulative reward, number of steps to locate Wally, and average time per episode. Performance was compared under different reward strategies and across CNN backbones for state encoding.

4.2.1 Learning Curve and Convergence

The RL agent was trained for 10,000 episodes using a Deep Q-Network (DQN) algorithm. The reward curve demonstrated smooth convergence after around 7000 episodes.

Episode Range	Avg Reward	Avg Steps to Find Wally
0–1000	-20.3	>150
1000–5000	15.7	~80
5000–7000	42.5	~40

7000–10000	49.3	~30

Table 7: Average Reward and Episode Length Over Time

4.2.2 Efficiency Metrics

The trained RL agent achieved a substantial reduction in the number of search steps required to find Wally, demonstrating an effective search strategy learned via reward optimization.

Agent Type	Avg Steps to Find Wally	Success Rate (%)	Avg Reward
Random Agent	143	12%	-15.2
Human (avg)	47	96%	N/A
RL Agent (DQN)	28	92%	48.7

Table 8: Comparison of RL Agent vs Baselines (Avg over 50 episodes)

The RL agent outperformed a random agent by a wide margin and approached human-level efficiency, although humans had a slightly higher success rate overall.

4.2.3 Effect of Reward Strategy

We tested two alternative reward strategies:

- Strategy A: +50 for finding Wally, -1 per move
- Strategy B: +100 for Wally, -0.1 per move, +5 for exploring new tiles

Reward Strategy	Avg Reward	Steps to Find	Exploration Coverage (%)
Strategy A	35.2	36	60%

Strategy B	49.3	28	92%

Table 9: RL Agent Performance Across Reward Strategies

Strategy B encouraged more efficient exploration and better final reward values, suggesting that reward shaping significantly impacts Q-function.

4.3 Human vs AI Evaluation

To contextualize the RL and CNN agent performance, we conducted a small-scale human benchmark study. Five volunteers were asked to find Wally in 10 test images using a digital interface that mimicked the agent's grid-based view.

4.3.1 Experimental Setup

Each participant was given the same images and asked to click on grid tiles to reveal them. Timing and click count were logged to simulate environment steps. Accuracy and time to locate Wally were computed.

Metric	Human (Avg)	RL Agent
Avg Steps to Wally	47	28
Avg Time to Wally (s)	33.4	6.2 (sim.)
Success Rate	96%	92%
Error Rate (False Clicks)	2.1	1.3

Table 10: Human vs RL Agent Comparison

While humans were slightly more accurate, the RL agent was significantly faster in simulated execution, benefiting from high-speed parallelized inference.

4.3.2 Qualitative Behavioural Analysis

Humans demonstrated flexible strategies, such as ignoring cluttered sections or prioritizing certain colour patterns. The RL agent, by contrast, relied on consistent scan patterns learned

through reward gradients and did not exhibit strong contextual awareness. However, its efficiency in controlled conditions suggests high potential in automated visual search tasks.

4.4 Ablation Studies

To isolate the contribution of specific components, we conducted ablation studies by modifying architectural elements and observing performance degradation.

Variation	Accuracy	F1 Score	Comments
No Dropout	89.1%	87.0%	Overfitting, reduced generalization
ReLU → Tanh Activation	68.7%	65.2%	Poor convergence, lower recall
Reduced CNN Depth	81.3%	79.0%	Insufficient feature extraction

Table 11: CNN Architecture Ablation Results (Model 2 Variants)

CNN Backbone	Reward (avg)	Success Rate	Convergence Episodes
Custom CNN	49.3	92%	7000
MobileNet	45.6	89%	8500
VGG16	43.2	88%	9000

Table 12: RL Ablation – CNN Backbone Comparison

Lightweight, custom-designed CNNs yielded faster and slightly better learning than deeper pretrained networks, possibly due to better fit with the task's image scale and simplicity.

5 Discussion

The Where's Wally? task presents a challenging intersection of visual recognition and sequential decision-making. Unlike typical object detection benchmarks that rely on cropped and neatly labelled image patches, this problem demands robustness in highly cluttered, dense, and distractor-rich environments. This approach, combining a convolutional neural network (CNN) for patch-level Wally classification and a reinforcement learning (RL) agent for efficient visual search over large images, addresses this dual challenge by separating perceptual inference from search strategy.

5.1 CNN-Based Object Detection: Performance and Interpretations

The CNN classifier proved to be a reliable perceptual module for identifying image patches containing Wally. Model 2, which incorporated a deeper architecture with additional dropout layers, significantly outperformed the baseline Model 1. Its test accuracy of 95.2%, combined with a recall of 95% and precision of 94.1%, indicates that the network could generalize well across previously unseen data, even when presented with subtle visual noise and distractors. This is especially notable given the data constraints of the problem domain. The training regime—augmented with class balancing via ImageDataGenerator and tuned over 10 epochs—clearly helped mitigate overfitting.

Despite this success, the CNN was not infallible. Qualitative inspection of misclassified examples revealed several consistent patterns. False negatives typically arose when Wally was partially occluded, appeared at extreme patch borders, or exhibited lower contrast due to lighting or background textures. False positives, on the other hand, emerged from human figures or objects that mimicked Wally's distinct red-and-white striped shirt or facial features. This highlights the CNN's reliance on texture and colour cues and its vulnerability to "perceptual shortcuts" in high-noise contexts.

Notably, the CNN achieved its highest validation accuracy around epoch 8, with subsequent minor declines indicating the onset of overfitting. This suggests that future iterations of the model could benefit from early stopping strategies, more advanced regularization (e.g., spatial dropout), or stronger architectural priors such as attention mechanisms. Furthermore, training the CNN in a multi-task setting—e.g., combining Wally classification with auxiliary tasks such as colour region segmentation—could further improve robustness.

5.2 Reinforcement Learning for Sequential Search: Behaviour and Strategy

The integration of the CNN with a reinforcement learning agent was essential to tackle the full Where's Wally? puzzle. While the CNN acted as the "eyes" of the system, the RL agent served as its "strategic brain"—deciding where to look, how to scan, and when to terminate search. The formulation of the environment as a grid-based Gymnasium space enabled structured and repeatable exploration while preserving the flexibility of RL training.

Using DQN, we trained the agent to maximize a sparse, delayed reward by finding Wally efficiently. The agent demonstrated steady improvement, with average episode length decreasing from over 150 steps in the early stages of training to just 28 by the end. This

reduction in search time—while maintaining a high success rate of over 90%—indicates that the RL agent was able to develop non-trivial exploration strategies that prioritized certain grid regions over others.

Reward shaping played a critical role in shaping Q-function quality. Strategy B, which introduced small positive rewards for exploring unseen tiles and penalized redundant actions lightly, produced significantly higher average returns and faster convergence than Strategy A's simpler sparse-reward setup. This result aligns with theoretical insights from RL literature: sparse or uninformative reward functions tend to hinder efficient exploration and lead to local minima. By guiding the agent gently toward information gain (i.e., novel tile exploration), we accelerated learning and improved generalization.

The comparison to a random search baseline further reinforces the strength of the learned Q-function. Whereas the random agent needed, on average, 143 steps to find Wally (and succeeded in only 12% of episodes), the DQN-trained agent succeeded 92% of the time and completed the task in less than a quarter of the steps. These differences are not merely quantitative—they imply a qualitative shift in the agent's internal model of the environment. Rather than treating each tile as equally likely, the agent developed structured exploration policies, possibly biased by features such as central image salience or regions previously correlated with Wally's appearances in training data.

However, limitations remain. The RL agent, though efficient, lacks contextual understanding. Unlike humans, who intuitively avoid visually cluttered zones or prioritize crowd centers, the agent's behaviour is driven purely by tile-based CNN predictions and reward history. In cases where the CNN misclassified Wally-containing tiles, the agent could become "trapped" in unproductive subregions. This coupling between perceptual and policy errors is a known challenge in RL-based object search and suggests future work in uncertainty-aware exploration or active vision.

5.3 Interpretation and Analysis

This section presents a critical examination of the results derived from these two-stages Where's Wally? pipeline—consisting of a CNN-based patch classifier and a reinforcement learning (RL) agent trained to search for Wally efficiently. We contextualize performance results, draw connections to human visual behaviour, and evaluate both the strengths and shortcomings of these methods. This holistic analysis underscores the unique opportunities and limitations that such AI systems reveal when compared to human cognitive strategies.

5.4 Interpretation of Results

5.4.1 CNN Classifier

The convolutional neural network (CNN) served as the perceptual backbone of this system, classifying small 64×64-pixel patches as either "Wally" or "Non-Wally." Based on test results, the model achieved a test accuracy of 95.2%, a precision of 94.1%, and a recall of 95%. The F1 score of 94.5% indicates a robust balance between true positive detections and false positives.

Metric	Value (%)
Accuracy	95.2
Precision	94.1
Recall	95.0
F1 Score	94.5
False Positive Rate	4.1
False Negative Rate	5.0

Table 13: CNN Classifier Performance on Test Set

False negatives—when the model failed to detect Wally in a patch—mostly occurred when Wally was partially occluded, heavily camouflaged within similar colours, or at the boundary between tiles. False positives were generally triggered by characters with red-and-white stripes or similar head shapes. These misclassifications illustrate the CNN's sensitivity to high-level textures and colours, but also its lack of deeper semantic understanding.

5.5 Reinforcement Learning Agent

The RL agent, trained using DQN (Deep Q-Network), efficiently searched grid-based puzzle images with an average success rate of 92% over 100 episodes. The average number of steps taken per episode fell from 150 in early training epochs to just 28 in the final evaluation. This strongly indicates that the agent learned structured and strategic search behaviours.

Metric	Value
Average Steps (Final)	28

Success Rate	92%
Training Episodes	5000
Training Time (hrs)	7.4
Average Reward (Final)	+0.78

Table 14: RL Agent Search Efficiency and Success Rate

The shaped reward strategy (Strategy B), which provided smaller positive rewards for visiting new tiles and penalized redundancy, significantly outperformed the simpler binary strategy (Strategy A). This highlights the agent's responsiveness to richer reward signals that incentivize novelty and efficiency simultaneously.

5.6 Human vs. AI Performance

The human benchmark provides an essential reference for understanding both the capability and limitations of this AI system. Human participants located Wally with slightly higher accuracy (96%) and fewer perceptual errors but required, on average, more steps (47) than the agent (28).

Metric	AI Agent	Human Participants
Average Steps	28	47
Success Rate	92%	96%
False Positives	8%	2%
Time per Step (avg)	0.02s	~1.2s

Adaptivity to Occlusion	Low	High

Table 15: Comparison of AI Agent and Human Participants

Humans were particularly resilient to occlusion, distractors, and off-centre clues, often leveraging global context (e.g., crowd clusters, visual salience) to inform where to search next. The AI agent, although efficient, lacked this high-level contextual reasoning and was more prone to become "stuck" in low-yield regions if misled by faulty CNN outputs.

5.7 Strengths and Limitations of the Approach

5.7.1 Strengths

Modularity & Interpretability: The two-stage pipeline—CNN followed by an RL agent—offered modular interpretability. Each component could be evaluated, trained, and optimized independently, and their outputs clearly mapped to visual or behavioural outcomes.

Scalability: The grid-based RL formulation allowed for scalability to arbitrarily large puzzle images. Unlike exhaustive sliding-window detection approaches, the agent learned to prioritize search zones, resulting in significantly lower computational overhead.

Realistic Evaluation: The human-AI comparison provided an ecologically valid benchmark, moving beyond synthetic metrics (e.g., accuracy) and assessing cognitive behaviours such as search strategy, error recovery, and generalization.

5.7.2 Limitations

Grid-Based Discretization: Modelling the environment as a uniform tile grid, while simplifying implementation, led to coarse movements and occasional missed detections (e.g., when Wally's face straddled two tiles). A continuous action space or dynamic zooming may provide finer search granularity.

Error Propagation: The Q-learning updates came with a trade-off—false negatives from the CNN caused the RL agent to miss Wally even when scanning the correct tile. This brittle dependency between modules could be mitigated by jointly training a detection + navigation model.

Limited Scene Understanding: The RL agent lacked access to global scene semantics. It made decisions based solely on past rewards and local CNN outputs. In contrast, humans use top-down scene reasoning (e.g., avoiding blue sky areas, seeking crowd density), which the current AI does not replicate.

5.8 Generalizability to Other Domains

The methods employed in this study generalize well to other domains where target detection is coupled with search over large or cluttered visual spaces. Potential applications include:

Medical Imaging: Locating tumors or anomalies in MRI or CT scans, where CNNs could identify suspicious regions and RL agents could sequentially "scan" a volume.

Remote Sensing: Satellite imagery analysis for detecting vehicles, deforestation, or buildings, with a similar patch-wise search over gigapixel images.

Security and Surveillance: Identifying individuals or threats in CCTV footage using an agent that learns efficient visual sweeps guided by CNN predictions.

Autonomous Robotics: Integrating visual search policies into robots navigating indoor or outdoor environments where objects of interest (e.g., tools, humans) must be located dynamically.

What makes this approach particularly transferable is its ability to work with limited supervision (only Wally/no-Wally labels per tile) and sparse feedback (rewards only upon finding the target), which mirrors the constraints in real-world deployments.

5.9 Unexpected Findings and Behaviours

Several observations emerged during the experiments that were not initially anticipated but revealed valuable insights:

Agent Preference Bias: The RL agent developed an emergent bias toward central regions of the image—even when Wally was equally likely to appear anywhere. This mirrors a known cognitive bias in humans, who often begin visual search tasks from the centre. While we did not encode this into the reward function, the agent likely learned it implicitly based on historical success.

Non-Greedy Exploration: Contrary to expectations, the most successful agents did not behave greedily (i.e., immediately jumping to the next highest-probability tile). Instead, they favoured exploration strategies that balanced coverage with exploitation. This suggests that the agent learned a form of strategic patience or scanning rhythm.

Reward Shaping Impact: The effect of different reward strategies was more dramatic than anticipated. Strategy A (sparse) often failed to converge, resulting in repetitive or random behaviour. In contrast, Strategy B enabled agents to learn quickly and generalize better. This affirms the growing consensus in RL that reward shaping is often more impactful than architectural complexity.

Adversarial Failure Cases: In rare cases, intentionally constructed distractor characters (e.g., red-and-white striped figures) caused both the CNN and RL agent to misclassify and fixate on incorrect regions. These adversarial edge cases suggest vulnerability to mimicry-based attacks and call for improved discriminative models or ensemble confidence scoring.

5.10 How Well the AI Mimics or Diverges from Human Search Behaviour

Perhaps the most illuminating insight from this research was how the AI agent both mimicked and diverged from human search behaviour.

5.11 Mimicry

- Both the agent and humans showed central visual bias, starting their search in the middle of the image more often than chance would predict.

- Both employed a semi-structured scanning behaviour, preferring to search in rows or clusters rather than completely random tiles.

- The AI agent developed a preference for denser regions (e.g., crowds), likely due to training data distribution or higher CNN activation in these areas—paralleling human preference for semantically rich zones.

5.12 Divergences

- Humans used context and gestalt: They would ignore irrelevant areas (sky, empty ground) even without explicit prior knowledge. The agent lacked such priors and often wasted steps in visually uninformative zones.
- Adaptability: When faced with occlusions or ambiguous tiles, human participants re-evaluated their strategy dynamically. The RL agent, however, did not exhibit robust recovery after misclassification.
- Visual abstraction: Humans used abstraction ("this looks like a crowd where Wally usually is") whereas the agent operated strictly based on CNN probabilities and reward feedback.

Overall, while the AI displayed surprisingly human-like behaviours in exploration style and efficiency, it lacked cognitive flexibility and semantic awareness—highlighting the boundary between optimized statistical learning and true visual intelligence.

5.13 Comparison with Human Performance

A particularly revealing component of this study was the human benchmark. By tasking volunteers to solve the same visual search puzzles using an identical tile-based interface, we generated a fair and interpretable comparison. Human participants averaged 47 steps to find Wally and achieved a 96% success rate—slightly better than the RL agent's 28 steps and 92%, respectively.

This result deserves nuanced interpretation. On one hand, the RL agent outperformed humans in raw efficiency, completing the task in fewer steps and (when deployed in simulation) drastically less time. On the other hand, humans demonstrated better robustness: their performance was less sensitive to image variation, occlusion, and misclassification. They also made fewer repetitive clicks and were more adaptive in strategy, often skipping over dense regions or exploiting scene-wide context clues such as crowd density or horizon lines.

These observations suggest that while the agent has learned a locally optimal strategy, it lacks the meta-cognitive flexibility that humans deploy naturally. This gap points toward future

research directions, such as integrating scene-level priors, using attention-based global scanning, or incorporating short-term memory mechanisms into the policy architecture.

Moreover, the difference in perceptual granularity may have influenced results. Humans can process a whole tile instantly, integrating colour, shape, and semantic content at a glance. The agent, by contrast, relies entirely on CNN-inferred probabilities. This raises the prospect of hybrid human-in-the-loop systems where CNN-based pre-filtering guides human attention, or vice versa.

5.14 Ablation and Architectural Analysis

These ablation studies provided critical insight into the architectural and training components that contributed most significantly to performance. On the CNN side, dropout regularization emerged as crucial for generalization; removing dropout layers led to a drop in accuracy and a visible overfitting pattern. Likewise, swapping ReLU with tanh activations severely degraded learning, confirming ReLU's superior gradient flow properties in deep networks. Reducing the CNN depth also impaired performance, indicating that a minimal feature hierarchy is necessary to capture the complexity of Wally's visual signature.

On the RL side, comparisons between different CNN backbones (e.g., MobileNet, VGG16, custom CNN) revealed that lightweight, task-specific architectures led to faster convergence and superior policy behaviour. The custom CNN not only trained faster but also generated more accurate state embeddings for the policy network, likely due to its alignment with the patch-scale image input and its training from scratch on Wally-specific patterns.

These findings suggest that end-to-end RL pipelines may benefit more from lean and domain-adapted CNNs than from large pretrained networks—especially when the target task differs substantially from ImageNet-style categorization.

5.15 Limitations and Future Work

While these results demonstrate the viability of combining CNNs and RL for visual search tasks, several limitations remain. First, the CNN classifier was trained on cropped, manually labelled patches. Though this was effective, it created a somewhat artificial training environment compared to the full puzzle image context. Future work could explore end-to-end object detection using region proposal networks or transformer-based attention modules that process entire images.

Second, the RL agent's environment was discretized into a uniform grid. While this abstraction simplifies training and analysis, it limits the granularity of search and fails to exploit visual salience cues that do not align with the grid. Extending the agent's observation and action space to allow variable zoom or dynamic field-of-view scanning could make its behaviour more human-like and efficient.

Third, while reward shaping improved performance, it was largely heuristic and manually tuned. More principled approaches, such as curiosity-driven exploration, intrinsic motivation, or meta-learning reward functions, could further enhance learning efficiency and transferability.

Finally, though the human benchmark was valuable, it was conducted on a small scale with a limited participant pool. Broader studies involving different age groups, cognitive profiles, or

search conditions (e.g., time-limited tasks, image distortions) could provide richer insights into the comparative strengths of human and AI agents.

5.16 Broader Implications

Beyond the Where's Wally? puzzle, the methods developed in this project have broader applications in any domain that involves high-clutter visual environments and sequential search. Examples include medical imaging (e.g., locating lesions in radiographs), surveillance (e.g., detecting anomalies in crowded scenes), and autonomous robotics (e.g., object localization in complex environments). This result suggests that decoupling perceptual modelling from strategic exploration—while integrating them via RL—can yield highly effective visual systems.

Furthermore, the human-like search behaviour exhibited by the agent—albeit in a simplified environment—demonstrates the potential of RL as a framework not just for optimal action selection but for modelling cognitive search processes. With appropriate inductive biases and architectural scaffolding, agents could be trained to emulate human-level adaptability and efficiency in complex visual reasoning tasks.

5.17 Real-World Analogues: Search-and-Rescue, Surveillance, and Beyond

The 'Where's Wally' puzzle might appear as a game, but at its heart lie problems that reflect reality on a number of fronts. In case of responding to such scenarios as natural disasters or refugee camps, where environment is cluttered, aerial or satellite images are used to locate people in the surroundings [41]. Targeted real-time detection of an individual figure or action in crowded public areas such as airports or sports stadiums remains a challenge to the human staff and automated systems.

Autonomous cars need to overcome visual complexities in congested areas where there are many pedestrians because the ability to recognise and observe people reliably is crucial for safety as well as for reliable decision-making processes [42][43]. Detection of important anatomical principles within noisy medical images such as locating polyps during colonoscopy or tumors in mammograms is similar to approach to the Wally problem. Having an effective search, safe identification, and fast decision leading in all these situations plays an important role. Successful enhancement of the performance metrics of AI systems on the Wally task will allow researchers to acquire important insights and even improve the development of practical solutions in more complex situations.

6 Conclusion and Recommendations

In this research, we presented a novel two-stage visual search framework that integrates a convolutional neural network (CNN) classifier with a reinforcement learning (RL) agent to efficiently solve the "Where's Wally?" problem. This task, which mirrors the complex cognitive process of human visual search, served as a challenging and illustrative benchmark to test the capabilities of current AI systems in perception, decision-making, and goal-directed exploration.

The CNN was tasked with classifying small image patches as either "Wally" or "Non-Wally," while the RL agent was responsible for sequentially selecting which patch to examine next in order to locate Wally with as few steps as possible. This approach thus combined passive visual recognition with active search behaviour, creating a hybrid system that approximated the perceptual and strategic faculties of human observers.

6.1 Conclusion

From a high-level perspective, the results obtained from This system demonstrate that AI can approach— and in some cases exceed—human efficiency in structured visual search problems when guided by a reliable perceptual backbone and a strategically trained Q-function. The CNN classifier achieved a test accuracy of over 95%, and the DQN-based RL agent located Wally successfully in 92% of test episodes with an average of only 28 steps. When benchmarked against human participants, the agent demonstrated competitive efficiency, albeit with some qualitative limitations such as brittleness to occlusion, lack of contextual awareness, and sensitivity to misclassification errors.

One of the main achievements of this work is its demonstration of how reinforcement learning can be successfully adapted to large-scale visual tasks by discretizing the input image into a manageable grid and structuring the search process as a Markov Decision Process. Unlike exhaustive or greedy search strategies, the trained agent developed emergent exploration patterns that were adaptive and efficient, avoiding previously visited areas and favouring high-probability regions as estimated by the CNN. These patterns included strategies like perimeter sweeps, row-wise scanning, and clustering behaviour—all of which bear similarity to human tactics observed in visual attention studies.

Moreover, these ablation studies provided evidence that shaping the reward function had a significant effect on learning stability and final performance. Sparse binary rewards led to convergence failures or inefficient exploration, while denser shaped rewards encouraged better coverage and faster convergence. This aligns with findings in broader RL literature that suggest the reward function often plays a more crucial role than the underlying network architecture in complex environments.

We also observed several unexpected behaviours and emergent strategies. For instance, the agent developed a tendency to prioritize central and densely populated regions of the image, echoing well-documented central bias effects in human vision. Interestingly, these preferences were not explicitly programmed but learned through iterative reward optimization and Q-function updates.

Despite these promising results, the project revealed clear limitations. The grid-based decomposition of images introduced alignment errors, particularly when Wally was located at

patch boundaries. The system's modularity—while helpful for debugging and interpretability—also meant that perception errors from the CNN could not be corrected downstream by the RL agent, leading to compounding failures in some cases. Furthermore, the RL agent lacked access to high-level semantic information, such as crowd density, object salience, or contextual cues that humans intuitively use to narrow their search scope.

From a cognitive modelling perspective, the agent mimicked certain low-level aspects of human search behaviour (e.g., scan path structure, central bias, avoidance of repeated areas) but diverged significantly in terms of flexibility, contextual inference, and error recovery. Humans were more resilient to distractors, better at interpreting occluded figures, and used high-level scene understanding to guide their attention—all capacities not present in the current system.

6.2 Recommendations

Building upon the findings and limitations observed in this study, we offer the following recommendations for future research and development:

1. End-to-End Training of Visual and Q-function Modules

While this modular architecture allowed for controlled experimentation, an end-to-end system could enable the Q-function to learn to compensate for perceptual noise or uncertainty. Integrating the CNN and RL agent into a unified architecture—where the classifier's outputs feed directly into the policy network and receive gradients from the reward signal—would allow for joint optimization and potentially better robustness.

2. Incorporation of Uncertainty Estimation

False positives and negatives had an outsized impact on search performance due to the binary nature of the classification output. Introducing probabilistic models or Bayesian CNNs could provide uncertainty estimates for each patch, which the RL agent could use to modulate its exploration policy. For example, high-confidence predictions could be prioritized, while uncertain regions could be revisited.

3. Hierarchical or Multiscale Search Strategies

Humans frequently perform coarse-to-fine search, starting with a broad sweep of salient areas and then zooming in on promising zones. Incorporating a hierarchical search mechanism—where the agent first scans large tiles and then drills down into smaller ones—could improve search precision and reduce the incidence of boundary-related detection failures.

4. Scene-Level Context Modelling

To better approximate human visual intelligence, future systems should include modules capable of extracting and reasoning over global scene properties. These could include object density maps, colour histograms, or saliency heatmaps. Such context-aware reasoning could be achieved using attention mechanisms or graph neural networks that encode spatial relationships between objects or regions.

5. Use of Memory-Augmented RL Architectures

Adding external memory to the RL agent could help it track which regions have already been visited and where promising leads were found, improving both exploration efficiency and resilience to repeated errors. Techniques such as neural maps, episodic memory, or recurrent architectures like LSTMs or Transformers could serve this role.

6. Adversarial Robustness and Defensive Training

The system's vulnerability to distractor characters that visually resembled Wally raises concerns about adversarial robustness. Training with augmented data that includes such distractors, or employing adversarial training methods, could reduce the likelihood of catastrophic errors in the presence of misleading inputs.

7. Human-in-the-Loop and Interactive AI

Another avenue is to integrate human feedback during training or deployment. A human-in-the-loop setup could allow the agent to request clarification when uncertain, or to accept corrections after a failed search. Such hybrid systems would combine the speed and scalability of AI with the judgment and flexibility of human cognition.

8. Expansion to Multimodal or Temporal Search Tasks

The current formulation assumes static images and single-target detection. In future iterations, this framework could be extended to tasks that require tracking moving targets (e.g., surveillance footage), locating multiple instances of an object (e.g., all "Wallys" in a scene), or incorporating audio/textual clues (e.g., "Wally is near the castle").

9. Evaluation in More Ecological Settings

While the "Where's Wally?" domain is visually rich and cognitively complex, it is still a simplified and bounded environment. Future evaluations should consider more realistic datasets, such as search-and-rescue images, radiological scans, or urban surveillance videos, to test the generalization capacity and robustness of the proposed system.

10. Ethical Considerations and User Transparency

Finally, as such visual search systems gain traction in sensitive domains like surveillance or medicine, it is critical to incorporate transparency, explainability, and ethical safeguards. Users should be able to interpret the system's decisions, assess its confidence, and override incorrect conclusions. Designing AI systems with built-in accountability mechanisms will be vital to fostering trust and responsible adoption.

This project has showcased that by combining powerful perceptual models with intelligent decision-making agents, AI can begin to replicate aspects of human visual search. While we remain far from fully modelling human cognition, this study has laid foundational work for modular, interpretable, and effective visual search agents. The insights gained from both successful and failed behaviours have practical implications for AI deployment in real-world tasks and open up rich new directions for interdisciplinary research at the intersection of computer vision, reinforcement learning, and cognitive science.

7 References

[1] Yihan Liao Guang Yang,Wenjin Pan and Y. Lu, "OA-HybridCNN (OHC): An advanced deep learning fusion model for enhanced diagnostic accuracy in knee osteoarthritis imaging," *PLoS One,* vol. 20, *(5),* 2025. Available: https://www.proquest.com/scholarly-journals/oa-hybridcnn-ohc-advanced-deep-learning-fusion/docview/3201460153/se-2. DOI: https://doi.org/10.1371/journal.pone.0322540.

[2] C. Chi-Sheng *et al*, "Improving fine-grained food classification using deep residual learning and selective state space models," *PLoS One,* vol. 20, *(5),* 2025. Available: https://www.proquest.com/scholarly-journals/improving-fine-grained-food-classification-using/docview/3200695430/se-2. DOI: https://doi.org/10.1371/journal.pone.0322695.

[3] S. Jayaraman and A. Mahendran, "CNN-LSTM based emotion recognition using Chebyshev moment and K-fold validation with multi-library SVM," *PLoS One,* vol. 20, *(4),* 2025. Available: https://www.proquest.com/scholarly-journals/cnn-lstm-based-emotion-recognition-using/docview/3187475454/se-2. DOI: https://doi.org/10.1371/journal.pone.0320058.

[4] Z. Qin *et al*, "Domain generalization for image classification based on simplified self ensemble learning," *PLoS One,* vol. 20, *(4),* 2025. Available: https://www.proquest.com/scholarly-journals/domain-generalization-image-classification-based/docview/3186564336/se-2. DOI: https://doi.org/10.1371/journal.pone.0320300.

[5] G. O. Sobola, S. Daramola and E. Adetiba, "A Survey of the Advances in the Applications of Deep Learning Algorithms Across Different Domains," *Ingenierie Des Systemes d'Information,* vol. 30, *(3),* pp. 779-795, 2025. Available: https://www.proquest.com/scholarly-journals/survey-advances-applications-deep-learning/docview/3191816392/se-2. DOI: https://doi.org/10.18280/isi.300322.

[6] Y. Cao *et al*, "Brain tumor intelligent diagnosis based on Auto-Encoder and U-Net feature extraction," *PLoS One,* vol. 20, *(3),* 2025. Available: https://www.proquest.com/scholarly-journals/brain-tumor-intelligent-diagnosis-based-on-auto/docview/3180869825/se-2. DOI: https://doi.org/10.1371/journal.pone.0315631.

[7] A. Iqbal *et al*, "Optimising window size of semantic of classification model for identification of in-text citations based on context and intent," *PLoS One,* vol. 20, *(3),* 2025. Available: https://www.proquest.com/scholarly-journals/optimising-window-size-semantic-classification/docview/3180869390/se-2. DOI: https://doi.org/10.1371/journal.pone.0309862.

[8] A. Hasan *et al*, "A comprehensive survey and comparative analysis of time series data augmentation in medical wearable computing," *PLoS One,* vol. 20, *(3),* 2025. Available: https://www.proquest.com/scholarly-journals/comprehensive-survey-comparative-analysis-time/docview/3178693865/se-2. DOI: https://doi.org/10.1371/journal.pone.0315343.

[9] R. Liu, S. Wen and Y. Xing, "An integrated approach for advanced vehicle classification," *PLoS One,* vol. 20, *(2),* 2025. Available: https://www.proquest.com/scholarly-journals/integrated-approach-advanced-vehicle/docview/3168278421/se-2. DOI: https://doi.org/10.1371/journal.pone.0318530.

[10] A. A. Yılmaz, "A novel deep learning-based framework with particle swarm optimisation for intrusion detection in computer networks," *PLoS One,* vol. 20, *(2),* 2025. Available: https://www.proquest.com/scholarly-journals/novel-deep-learning-based-framework-with-particle/docview/3166117498/se-2. DOI: https://doi.org/10.1371/journal.pone.0316253.

[11] H. Ma *et al*, "Seafloor Sediment Classification Using Small-Sample Multi-Beam Data Based on Convolutional Neural Networks," *Journal of Marine Science and Engineering,* vol. 13, *(4),* pp. 671, 2025. Available: https://www.proquest.com/scholarly-journals/seafloor-sediment-classification-using-small/docview/3194618827/se-2. DOI: https://doi.org/10.3390/jmse13040671.

[12] PDF, "Related Applications of Deep Learning Algorithms in Medical Image Fusion Systems," *International Journal of Advanced Computer Science and Applications,* vol. 16, *(3),* 2025. Available: https://www.proquest.com/scholarly-journals/related-applications-deep-learning-algorithms/docview/3192357872/se-2. DOI: https://doi.org/10.14569/IJACSA.2025.0160338.

[13] X. Wu, "Deep Learning-based Image Recognition Technology in Medical Diagnosis," *Applied Mathematics and Nonlinear Sciences,* vol. 10, *(1),* 2025. Available: https://www.proquest.com/scholarly-journals/deep-learning-based-image-recognition-technology/docview/3190090494/se-2. DOI: https://doi.org/10.2478/amns-2025-0008.

[14] Anonymous "Quantum optical classifier with superexponential speedup," *Communications Physics,* vol. 8, *(1),* pp. 147, 2025. Available: https://www.proquest.com/scholarly-journals/quantum-optical-classifier-with-superexponential/docview/3188768752/se-2. DOI: https://doi.org/10.1038/s42005-025-02020-5.

[15] Yong-Yeol Bae, Dae-Jea Cho and J. Ki-Hyun, "A New Log-Transform Histogram Equalization Technique for Deep Learning-Based Document Forgery Detection," *Symmetry,* vol. 17, *(3),* pp. 395, 2025. Available: https://www.proquest.com/scholarly-journals/new-log-transform-histogram-equalization/docview/3181696348/se-2. DOI: https://doi.org/10.3390/sym17030395.

[16] X. Liu *et al*, "PRNet: A Priori Embedded Network for Real-World Blind Micro-Expression Recognition," *Mathematics,* vol. 13, *(5),* pp. 749, 2025. Available: https://www.proquest.com/scholarly-journals/prnet-priori-embedded-network-real-world-blind/docview/3176336150/se-2. DOI: https://doi.org/10.3390/math13050749.

[17] A. A. Mehmet, K. Kaplan and M. Kuncan, "A Comparative Study of Various Transfer Learning Models on Skin Cancer Confirmation Methods," *Journal of Universal Computer Science,* vol. 31, *(2),* pp. 113-135, 2025. Available: https://www.proquest.com/scholarly-journals/comparative-study-various-transfer-learning/docview/3174359241/se-2. DOI: https://doi.org/10.3897/jucs.118220.

[18] Z. Liu *et al*, "Identification of diabetic retinopathy lesions in fundus images by integrating CNN and vision mamba models," *PLoS One,* vol. 20, *(1),* 2025. Available: https://www.proquest.com/scholarly-journals/identification-diabetic-retinopathy-lesions/docview/3160818689/se-2. DOI: https://doi.org/10.1371/journal.pone.0318264.

[19] L. Li *et al*, "Enhanced ResNet-50 for garbage classification: Feature fusion and depth-separable convolutions," *PLoS One,* vol. 20, *(1),* 2025. Available: https://www.proquest.com/scholarly-journals/enhanced-resnet-50-garbage-classification-feature/docview/3160322890/se-2. DOI: https://doi.org/10.1371/journal.pone.0317999.

[20] N. Akter *et al*, "Glaucoma detection and staging from visual field images using machine learning techniques," *PLoS One,* vol. 20, *(1),* 2025. Available: https://www.proquest.com/scholarly-journals/glaucoma-detection-staging-visual-field-images/docview/3156772699/se-2. DOI: https://doi.org/10.1371/journal.pone.0316919.

[21] G. Logavitool *et al*, "Field-scale detection of Bacterial Leaf Blight in rice based on UAV multispectral imaging and deep learning frameworks," *PLoS One,* vol. 20, *(1),* 2025. Available: https://www.proquest.com/scholarly-journals/field-scale-detection-bacterial-leaf-blight-rice/docview/3156772544/se-2. DOI: https://doi.org/10.1371/journal.pone.0314535.

[22] X. Xu, F. Lei and H. Liu, "Automated Extraction and Classification of Product Design Elements Based on Image Recognition," *Traitement Du Signal,* vol. 41, *(6),* pp. 2883-2893, 2024. Available: https://www.proquest.com/scholarly-journals/automated-extraction-classification-product/docview/3157167002/se-2. DOI: https://doi.org/10.18280/ts.410609.

[23] F. Lin *et al*, "Enhancing multimedia management: cloud-based movie type recognition with hybrid deep learning architecture," *Journal of Cloud Computing,* vol. 13, *(1),* pp. 104, 2024. Available: https://www.proquest.com/scholarly-journals/enhancing-multimedia-management-cloud-based-movie/docview/3055690015/se-2. DOI: https://doi.org/10.1186/s13677-024-00668-y.

[24] H. P. Bui *et al*, "Classification of Fruits using Image Processing and Deep Neural Networks," *International Journal of Advanced Research in Computer Science,* vol. 15, *(6),* pp. 19-22, 2024. Available: https://www.proquest.com/scholarly-journals/classification-fruits-using-image-processing-deep/docview/3174037812/se-2. DOI: https://doi.org/10.26483/ijarcs.v15i6.7156.

[25] S. Sanchez *et al*, "Data augmentation via warping transforms for modelling natural variability in the corneal endothelium enhances semi-supervised segmentation," *PLoS One,* vol. 19, *(11),* 2024. Available: https://www.proquest.com/scholarly-journals/data-augmentation-via-warping-transforms-modelling/docview/3127538751/se-2. DOI: https://doi.org/10.1371/journal.pone.0311849.

[26] H. Hsieh *et al*, "Deep learning-based automatic image classification of oral cancer cells acquiring chemoresistance in vitro," *PLoS One,* vol. 19, *(11),* 2024. Available: https://www.proquest.com/scholarly-journals/deep-learning-based-automatic-image/docview/3123292878/se-2. DOI: https://doi.org/10.1371/journal.pone.0310304.

[27] A. H. Ali and S. N. Pramod, "An Efficient Deep Learning Based AgriResUpNet Architecture for Semantic Segmentation of Crop and Weed Images," *Ingenierie Des Systemes d'Information,* vol. 29, *(5),* pp. 1829-1845, 2024. Available: https://www.proquest.com/scholarly-journals/efficient-deep-learning-based-agriresupnet/docview/3131480087/se-2. DOI: https://doi.org/10.18280/isi.290516.

[28] M. Zhou *et al*, "LostNet: A smart way for lost and find," *PLoS One,* vol. 19, *(10),* 2024. Available: https://www.proquest.com/scholarly-journals/lostnet-smart-way-lost-find/docview/3122530344/se-2. DOI: https://doi.org/10.1371/journal.pone.0310998.

[29] Y. Li, "DPNet: Scene text detection based on dual perspective CNN-transformer," *PLoS One,* vol. 19, *(10),* 2024. Available: https://www.proquest.com/scholarly-journals/dpnet-scene-text-detection-based-on-dual/docview/3119112732/se-2. DOI: https://doi.org/10.1371/journal.pone.0309286.

[30] M. Motiejauskas and G. Dzemyda, "EfficientNet Convolutional Neural Network with Gram Matrices Modules for Predicting Sadness Emotion," *International Journal of Computers, Communications and Control,* vol. 19, *(5),* 2024. Available: https://www.proquest.com/scholarly-journals/efficientnet-convolutional-neural-network-with/docview/3101782105/se-2. DOI: https://doi.org/10.15837/ijccc.2024.5.6697.

[31] T. Reza *et al*, "Domain affiliated distilled knowledge transfer for improved convergence of Ph-negative MPN identifier," *PLoS One,* vol. 19, *(9),* 2024. Available: https://www.proquest.com/scholarly-journals/domain-affiliated-distilled-knowledge-transfer/docview/3110685977/se-2. DOI: https://doi.org/10.1371/journal.pone.0303541.

[32] Y. Zhu *et al*, "Implementation of resource-efficient fetal echocardiography detection algorithms in edge computing," *PLoS One,* vol. 19, *(9),* 2024. Available: https://www.proquest.com/scholarly-journals/implementation-resource-efficient-fetal/docview/3108569286/se-2. DOI: https://doi.org/10.1371/journal.pone.0305250.

[33] J. Liu *et al*, "DLGRAFE-Net: A double loss guided residual attention and feature enhancement network for polyp segmentation," *PLoS One,* vol. 19, *(9),* 2024. Available: https://www.proquest.com/scholarly-journals/dlgrafe-net-double-loss-guided-residual-attention/docview/3103840237/se-2. DOI: https://doi.org/10.1371/journal.pone.0308237.

[34] H. H. Gabriel Toshio *et al*, "Smartphone region-wise image indoor localization using deep learning for indoor tourist attraction," *PLoS One,* vol. 19, *(9),* 2024. Available: https://www.proquest.com/scholarly-journals/smartphone-region-wise-image-indoor-localization/docview/3102338031/se-2. DOI: https://doi.org/10.1371/journal.pone.0307569.

[35] N. Aziz *et al*, "Precision meets generalization: Enhancing brain tumor classification via pretrained DenseNet with global average pooling and hyperparameter tuning," *PLoS One,* vol. 19, *(9),* 2024. Available: https://www.proquest.com/scholarly-journals/precision-meets-generalization-enhancing-brain/docview/3101516849/se-2. DOI: https://doi.org/10.1371/journal.pone.0307825.

[36] M. K. Baowaly *et al*, "Deep transfer learning-based bird species classification using mel spectrogram images," *PLoS One,* vol. 19, *(8),* 2024. Available: https://www.proquest.com/scholarly-journals/deep-transfer-learning-based-bird-species/docview/3092273839/se-2. DOI: https://doi.org/10.1371/journal.pone.0305708.

[37] N. M. Krishnan, S. Kumar and B. Panda, "Fruit-In-Sight: A deep learning-based framework for secondary metabolite class prediction using fruit and leaf images," *PLoS One,* vol. 19, *(8),* 2024. Available: https://www.proquest.com/scholarly-journals/i-fruit-sight-deep-learning-based-framework/docview/3090883186/se-2. DOI: https://doi.org/10.1371/journal.pone.0308708.

[38] J. Park, J. R. Mi and M. H. Moon, "Enhanced deep learning model for precise nodule localization and recurrence risk prediction following curative-intent surgery for lung cancer," *PLoS One,* vol. 19, *(7),* 2024. Available: https://www.proquest.com/scholarly-journals/enhanced-deep-learning-model-precise-nodule/docview/3079802753/se-2. DOI: https://doi.org/10.1371/journal.pone.0300442.

[39] M. Bhasin *et al*, "Unveiling the Hidden: Leveraging Medical Imaging Data for Enhanced Brain Tumor Detection Using CNN Architectures," *Traitement Du Signal,* vol. 41, *(3),* pp. 1575-1582, 2024. Available: https://www.proquest.com/scholarly-journals/unveiling-hidden-leveraging-medical-imaging-data/docview/3097397910/se-2. DOI: https://doi.org/10.18280/ts.410345.

[40] S. Fuladi *et al*, "Efficient Approach for Kidney Stone Treatment Using Convolutional Neural Network," *Traitement Du Signal,* vol. 41, *(2),* pp. 929-937, 2024. Available: https://www.proquest.com/scholarly-journals/efficient-approach-kidney-stone-treatment-using/docview/3097397980/se-2. DOI: https://doi.org/10.18280/ts.410233.

[41] J. E. Cubillas *et al*, "Land Classification Plugin for QGIS Using Pix2Pix," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. X-5-2024, pp. 33-39, 2024. Available: https://www.proquest.com/scholarly-journals/land-classification-plugin-qgis-using-pix2pix/docview/3126763021/se-2. DOI: https://doi.org/10.5194/isprs-annals-X-5-2024-33-2024.

[42] A. Tulsyan, U. Chaturvedi and N. V. Sharma, "Transferability of Learned Knowledge in Neural Networks: Impact of Trained Weights on Untrained Networks," *Journal of Electrical Systems,* vol. 20, *(2),* pp. 1549-1554, 2024. Available: https://www.proquest.com/scholarly-journals/transferability-learned-knowledge-neural-networks/docview/3074172415/se-2.

[43] V. C. Sai *et al*, "A reliable diabetic retinopathy grading via transfer learning and ensemble learning with quadratic weighted kappa metric," *BMC Medical Informatics and Decision Making,* vol. 24, pp. 1-12, 2024. Available: https://www.proquest.com/scholarly-journals/reliable-diabetic-retinopathy-grading-via/docview/2925563844/se-2. DOI: https://doi.org/10.1186/s12911-024-02446-x.

[44] K. Sharada *et al*, "Deep Learning Techniques for Image Recognition and Object Detection," *E3S Web of Conferences,* vol. 399, 2023. Available: https://www.proquest.com/conference-papers-proceedings/deep-learning-techniques-image-recognition-object/docview/2894390938/se-2. DOI: https://doi.org/10.1051/e3sconf/202339904032.

[45] M. U. Saeed *et al*, "An Automated Deep Learning Approach for Spine Segmentation and Vertebrae Recognition Using Computed Tomography Images," *Diagnostics,* vol. 13, *(16),* pp. 2658, 2023. Available: https://www.proquest.com/scholarly-journals/automated-deep-learning-approach-spine/docview/2856977694/se-2. DOI: https://doi.org/10.3390/diagnostics13162658.

[46] Z. Zhao and S. Zhang, "Style Transfer Based on VGG Network," *International Journal of Advanced Network, Monitoring, and Controls,* vol. 7, *(1),* pp. 54-72, 2022. Available: https://www.proquest.com/scholarly-journals/style-transfer-based-on-vgg-network/docview/3156503501/se-2. DOI: https://doi.org/10.2478/ijanmc-2022-0005.

[47] M. Shafiq and Z. Gu, "Deep Residual Learning for Image Recognition: A Survey," *Applied Sciences,* vol. 12, *(18),* pp. 8972, 2022. Available: https://www.proquest.com/scholarly-journals/deep-residual-learning-image-recognition-survey/docview/2716490462/se-2. DOI: https://doi.org/10.3390/app12188972.

[48] P. Sobti *et al*, "EnsemV3X: a novel ensembled deep learning architecture for multi-label scene classification," *PeerJ Computer Science,* 2021. Available: https://www.proquest.com/scholarly-journals/ensemv3x-novel-ensembled-deep-learning/docview/2531838353/se-2. DOI: https://doi.org/10.7717/peerj-cs.557.

[49] OpenAI (2025) *ChatGPT (GPT-5 Pro)*. Available at: https://chat.openai.com/ (Accessed: 29 September 2025).