ABERYSTWYTH UNIVERSITY

Evolutionary Active Vision System: From 2D to 3D

by

Olalekan Adebayo Lanihun

A thesis submitted in partial fulfilment for the
degree of Doctor of Philosophy

in the

Institute of Mathematics, Physics and Computer Science
Department of Computer Science

January 2018

Declaration of Authorship

DECLARATION

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

Signed: 

Date: 23/01/2018

STATEMENT 1

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

Signed: 

Date: 23/01/2018

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed: 

Date: 23/01/2018

"Wisdom is the principal thing; therefore, get wisdom; and with all thy getting get understanding"

Proverbs 4:7

ABERYSTWYTH UNIVERSITY

Abstract

Institute of Mathematics, Physics and Computer Science
Department of Computer Science

Doctor of Philosophy

by Olalekan Adebayo Lanihun

Humans appear to solve complex vision tasks in an almost effortless manner, as compared to their computer counterparts. One major reason for this is the intelligent cooperation between the sensory and the motor system, which is facilitated by development of motor skills that help to shape visual information that is relevant to a specific vision task. This dynamic interaction of sensory-motor components in biological systems can be a great inspiration to how artificial systems, such as robots could use their visual mechanism to interacts with their world. In this thesis, we seek to explore an approach to active vision inspired by biological evolution, which does not use a predefined framework or assumptions, but develops motor strategies for a given task through progressive adaptation of the evolutionary method. Thus, this kind of approach will give freedom to artificial systems in the discovery of eye movement strategies that may be useful to biological systems but are not known to us. The contributions of this thesis are:

1. We used this type of active vision system for more complex images taken from the camera of the iCub robot.

2. We demonstrated the effectiveness of the active vision system in a more realistic setting for 3D object categorisation using the humanoid robot (iCub) platform.

3. We extended the applicability of the system to the 3D environment for indoor and outdoor environment classification task using the iCub platform.

4. We extended the system with pre-processing using Uniform Local Binary Patterns [1] in both 2D and 3D environment categorisation tasks.

5. We further extended the system with pre-processing using Histogram of Oriented Gradients [2] for classification tasks in the 2D and 3D environments.

Analysis of the results from the system shows that the model was able to complete discrimination tasks through: (i) exploiting sensory-motor coordination to experience sensory stimuli that facilitates the classification tasks; (ii) an indication of integration of perceptual information over time.

Acknowledgements

First, I would like to thank my beloved wife for her total support during this PhD project. I would also like to express my profound gratitude to my supervisors, Dr Bernie Tiddeman and Dr Patricia Shaw for their encouragement, intelligent suggestions and advice over the course of the PhD work. My appreciation also goes to my former second supervisor, Dr Elio Tuci, who provided the robotic simulators used in the experiments of this thesis and for his support. To all my PhD colleagues, particularly, Aparajit Narayan for his help in the learning of the MPI parallel programming language.

Last, but not least, I would like to express my sincere thanks to my parents for their moral, psychological and financial support throughout my life.

This project has been partly funded by the Computer Science Department Overseas PhD Scholarship (CSDOPS).

Contents

List of Figures

List of Tables

Dedicated to my Lord and Saviour Jesus Christ

Chapter 1

Introduction

1.1 Research Background

Numerous studies have shown that action and perception cannot be separated and mutually influence one another [6][7]. In other words the sensory patterns the environment provides to an agent partially determines the agent's motor actions and these motor actions in turn, by modifying the environment, partially shape the type of sensory patterns experienced. Similarly, various studies show that the human eye is constantly searching for visual information mainly in the form of saccadic eye movement [8][9] (Fig. 1.1). These saccadic eye movements are very important because humans possess a very limited high-resolution vision at the fovea, covering the central two degrees of their visual field and have increasingly lower resolution towards the periphery [8]. There is therefore a need for intentional eye movements to perceive an area of interest in high resolution which enhances recognition capability [8]. This concept of dynamic interaction between a biological agent and its visual environment, which underscores the importance of cooperation of sensory-motor components in object perception may also be useful in artificial systems. This is because such dynamic interactions allow the system to intelligently determine the visual resources that are useful for a specific task and at the same time avoid disruptive information, as such they facilitate their cognitive capacities. In this thesis, we investigate an evolutionary approach to active vision ([10][11][12]), that allows an agent to dynamically explore its visual environment through sensory-motor coordination. This model does not use assumptions for eye movements (action strategies); instead, the model progressively adapts to the visual task at hand. Also, it is very important to clarify that we are not trying to model any natural vision system, but our model shares the following properties with natural systems: (i) it is situated in an environment and therefore its future outputs can be determined by its interaction with

this environment; (ii) the tasks performed by our system are also performed by natural agents, and as such similar strategies used by natural systems can be adapted by our system for the same tasks.

This chapter is detailed as follows: Section 1.2 introduces the active vision models and our evolutionary active vision model; this leads to our research questions in Section 1.3; in Section 1.4, we discuss our research methodology; in Section 1.5, we outline the major contributions of the thesis and finally, in Section 1.6, we provide a summary of the chapter.

FIGURE 1.1: Example of a saccadic eye movement of a person while scanning (image from [3])

1.2 Active Vision

Active vision is the process of exploring a visual scene to obtain relevant features for subsequent meaningful and intelligent processing. This is very important and very useful in that visual systems usually have a form of control, and are intelligently guided to only those areas of the image surface being processed that have relevant and valuable information to the task at hand. Vision is not a passive process as has been known in conventional computer vision [1][13][14], but is action dependent [15][16][17]. In most traditional computer vision, the local image sample does not guide the scanning process, but instead use an exhaustive search (e.g window sliding methods [18][19] and the

constellation method [20]). However, research shows that the use of action in perception can reduce the computational cost of vision tasks [21][22], and at the same time simplify very difficult tasks [23][24]. Consequently, as action has been shown to be an integral part of perception, the challenge in developing active vision models is finding intelligent action strategies that will enhance the vision task at hand [25].

In some models the assumption made is that vision is an iterative process of state estimation and the selection of relevant actions [26][27][28], however, in this work our aim is to develop an active vision system that has the following properties: (i) it does not make use of any kind of assumptions or predefined framework for its action strategy (eye movement); and (ii) it does not need any kind of ground truth. This is because such assumptions or ground truth may not allow the model to discover strategies that are not known to us and may be existing in properties of biological agents. We have therefore chosen an evolutionary adaptive model used in the field of evolutionary robotics [29][30] for learning the control of the active vision. This technique does not make use of assumptions or predefined frameworks for its action strategies (eye movements), but delegates the matter to the adaptation process of the evolutionary method. It is important to clarify here that it is not only evolutionary methods that can be used to achieve this objective, other adaptive methods such as reinforcement learning [31][32] can also be employed. However, we have chosen an evolutionary approach because of the following inherent properties: firstly, it is a semi-supervised algorithm and therefore can be used to model a system in which we know the goal but do not know the actions strategies to achieve this goal, and as a result we can optimise the actions towards achieving this task; secondly, because of its semi-supervised nature, it can find non-greedy action strategies, in order to optimise the performance of the model towards the final goal; and thirdly, multiple parts of the model can be optimised at the same time. For instance, we can adapt the visual features and the controller for the active vision at the same time (e.g. [33] and [34]).

Early research work on evolutionary active vision was used as a proof of concept. For instance, an evolutionary algorithm was applied to a robot in [35], that had to approach a triangle and avoid a rectangle, both drawn on the walls of the arena in which the robot had to manoeuvre. In a similar fashion to [35], Kato and Floreano [24] used an active vision model to discriminate between black squares and triangles in static images corrupted by various amount of noise. However, later work involved more complex task. For example, Nolfi and Marroco [12] developed an active system that guides a simple robot placed in a rectangular environment and was able to use its camera for discriminating between different landmarks on the walls. An active vision system controlled by an evolved recurrent neural network was developed by Morimoto and Ikegami [11] which dynamically discriminates between rectangular and triangular objects. In this system

when the agent moves through the environment it develops neural states which are not just a symbolic representation of rectangles or triangles, but allow it to distinguish these objects. Mirolli et al. [23] used an active vision system controlled by an evolved neural network in categorising five handwritten italic letters at different scales. According to Mirolli et al. [23], previous systems that used active categorisation perception were used for fewer than five categories. Furthermore, Guido De Croon [25] also developed an active vision system that used evolutionary adaptation for its eye movements. The system was used for the classification of object images with category ranges from 25 to 100. However, he mentions in his thesis that this model employed an explicit belief state in determining the probabilities of the classes, which is not completely consistent with the evolutionary robotics point of view that believe that the inner working of the classification task should also be self-organised. According to Guido De Croon, the compromise was made so that the model could be used in direct comparison with other existing active vision models (probability models), which also employed an explicit belief update in the object categorisation task. As such, his model is slightly different from our flavour of active vision model and the previously mentioned active vision models that also used a self-organising process for its classification task.

However, our work is different from the previously mentioned evolutionary approaches in the following respects:

(i) We aim to show the plausibility of biological active vision systems in complex artificial systems using our evolutionary method for categorisation tasks. As such, we have extended our method for categorisation to more realistic natural 2D images and to 3D environment using a Humanoid robot platform.

(ii) We investigated two pre-processing techniques in computer vision, i.e. Histogram of Oriented Gradients (HOG) [2] and Uniform Local Binary Patterns (ULBP) [1][36], so as to show how active vision can be enhanced by low level processing [37][38][39].

Our goal, therefore in this thesis is to develop an active system that can work in complex scenes and environments towards classification without the use of assumptions or predefined frameworks for its action strategies (eye movements). In the next section, we progress to the research questions for the thesis.

1.3 Research Questions

In this thesis we investigate the plausibility of evolutionary adaptive methods of control for an active vision in complex environments and how they use their motor skills in learning for classification. This, therefore leads us to the following research questions:

1. Do evolutionary methods of control of active vision systems for categorisation work in complex scenes and environments?

2. Can we make them work better e.g. with pre-processing techniques in computer vision?

1.4 Research Methodology

In order to answer the research questions, we have used the research methodology as follows.

Firstly, we did a thorough literature research on the existing active vision models to gain insight into their theoretical properties. On the basis of this, we identified not only their strengths and weaknesses, but also the ways in which these could be explored in the larger context of fulfilling the goal of the thesis. Secondly, we chose an existing evolutionary active vision system by Mirolli et al [23] as a bench-mark for our proposed system. The decision to use this particular system as our bench-mark was based on the following reasons:

1. The system uses an adaptive neural network controller, which shows its biological plausibility and therefore is similar in principle to our proposed model in building an abstraction of a human biological vision control.

2. The bench-mark system has all the inherent properties of current evolutionary active vision systems in the literature, which exploits coordination of sensory-motor information and/or with integration of experience sensory information over time [10][24][25].

3. The system was also trained in a semi-supervised manner that used an evolutionary optimised control system to improve a categorisation task.

4. The system was used for a complex categorisation task with a considerable number of categories and level of variability as compared to previous evolutionary active vision systems.

We used this benchmark system by Mirolli et al [23] for 2D static images as a proof of concept and also to demonstrate how an active vision system could be enhanced by low-level pre-processing techniques.

This was then extended to a 3D environment using the humanoid robot (iCub) platform, so as to show the plausibility of our system in more complex robotic systems. The review

of the literature is presented in Chapter 2, our methods and Gaze control framework are presented in Chapter 3. In Chapter 4, we demonstrate the enhancement of an active vision system with pre-processing techniques in 2D natural images using our benchmark for object categorisation. In Chapter 5, we instantiate our gaze control framework for object categorisation in 3D environment using the iCub robot simulator, while in Chapter 6 the same platform was used for indoor and outdoor environment classification. The reason for using these vision tasks for classification in our experiments was because the tasks have different problem structures and therefore different sensory-motor strategies are expected to be employed in the solving of these tasks. This will thus give us a more objective and conclusive means to answer our research questions. Finally, Chapter 7 gives a general discussion with conclusion on the research work and suggested areas for future work.

1.5 Outline of Thesis Contributions

1. Our first contribution is the extension of an evolutionary active vision system for object categorisation using more complex (natural) images taken from the camera of the iCub robot. Our bench-mark Mirolli et al. [23], which to the best of our knowledge has the largest number of categories in this type of active vision to date was used for handwritten images (Chapter 4).

2. The extension of evolutionary control active vision for object categorisation in a 3D environment (Chapter 5). Evolutionary active vision systems for object categorisation to the best of our knowledge have only been used in 2D environments (e.g [24][40]), mainly as a proof of concept. To gain a better insight into how this might behave in the real world, we have tested an agent interacting with the 3D environment using the coordination of sensory and motor information. This was implemented with a humanoid robot (iCub) simulator platform.

3. We further proved the use of the evolutionary active vision system for 3D indoor and outdoor environment classification using the humanoid robot platform (Chapter 6). To our knowledge, no computational model has been used for indoor and outdoor environment classification tasks on an humanoid robot platform until now. Various computer vision models have been used for 2D indoor and outdoor image classification for purposes such as categorisation and retrieval from databases [41][42][43]. Others that have been used on the 3D platform were mainly for scene categorisation of indoor or outdoor environments alone [44].

4. We extended the active vision system with pre-processing using Uniform Local Binary Patterns (ULBP)[1] (Chapter 4, 5, 6). The novelty here is using ULBP

originally developed by Ojala et al [1] as a pre-processing technique for the active window. Previous active vision models used simple techniques for pre-processing, such as average grey-scale values [23] and pixel sub-sampling [24]. We have taken advantage of the uniforms patterns of the ULBP method as a texture representation that is robust in terms of monotonic grey-scale transformation and less prone to noise [45]. This method has been shown to be a very good feature descriptor in many recognition tasks in the computer vision literature, such as in face recognition [46][47].

5. We further extended the active vision system with pre-processing using Histogram of Oriented Gradients (HOG)[2] (Chapter 4, 5 and 6). It is known that the low-level processing in the mammalian visual cortex makes use of gradient features which enhances its capability in recognition tasks [48]. It has also been commonly used in state of the art research works in computer vision especially that which involves structure of objects using gradients features such as in human detection [2][49] and object detection [50][51].

1.6 Chapter Summary

In this chapter we have discussed how an evolutionary active vision system does not use assumptions for the eye action strategies, but allows dynamic interaction of the system with its environment, and progressively adapts to a vision task. This gives the model freedom in discovering the action strategies that may be vital for the success in humans but are unknown to us.

However, evolutionary active vision systems have mostly been used in 2D categorisation tasks. This thesis extends evolutionary active vision to more complex categorisation tasks in 2D and 3D environments and enhances the categorisation capabilities with pre-processing techniques in computer vision. In the next chapter, we place our work in context by reviewing the active vision and categorisation models.

Chapter 2

Literature Review

2.1 Introduction

This thesis is about learning control of active vision for categorisation, which may be further improved with pre-processing, given the strong dependencies between perception and motor control. However, active vision models are inspired by the theory of sensory-motor coordination, in which behaviour of an organism emerges from the dynamical interaction between the organism and the external environment [52][53][54][55][56]. The conventional approach to visual perception views vision as a product of the brain, by which it first produces a detailed internal representation of the world and the activation of this internal representation is what gives rise to the experience of seeing [52]. On the other-hand, the sensory-motor approaches view vision as a mode of exploration of the world that is mediated by knowledge of sensory-motor contingencies [52]. In the words of Kevin O'Regan, and Alva Noe [52] "seeing is a way of acting in an outside world that serves as its own external representation". According to them, the experience of seeing occurs when the organism masters the governing rules of sensory-motor contingencies. Within this view, perception and motor action cannot be separated, and the behaviour that leads to visual perception emerges out of dynamic coordination of sensory-motor components. However, most existing gaze control models, that model the attention mechanism and active vision do not closely model the process of active vision, in that they usually set pre-defined features that determines the attention locations [57]. These models generally process the entire image and so do not allow feature selections to be determined by the behaviours emerging from the interaction between the agent and the environment [10]. In this chapter, we start by looking at the existing attention models that fail to meet with the requirements of an active vision model and as such do not closely model the active vision process [25]. A gaze control or attention model typically

models gaze shifts that determine the attention locations in a visual scene (Varella and Wyatt [58]). It is very important to state here that not all gaze control models are active vision models, but all active vision models are gaze control models. Typically active vision models should satisfy the following conditions:

1. They should have a limited high-resolution field of view known as the retina (fovea) which compels the agent to direct it around in order to perceive more information from its environment. It does not process the entire image or scene at a time [8].
2. The gaze control must be task oriented [25].
3. There is a closed loop dynamic relationship between the sensory stimuli and corresponding motor (action) responses [30][10].

However, not all the existing gaze control models satisfy these requirements. We proceed to review the current attention models, active vision models and subsequently models for object and environment categorisation. Section 2.2 discusses the common attention models that fail to meet the requirements for active vision systems, while Section 2.3 reviews the major active vision models in the literature. Section 2.4 discusses object categorisation and gives a review of the current methods use in object categorisation, and Section 2.5 gives a review of environment categorisation. Finally, in Section 2.6 a summary of the chapter is given.

2.2 Attention Models

We discuss the models of attention that do not meet all the requirements as specified above for an active vision system.

The first models in this group fail to meet the first requirement of our active vision system in that they process the entire image. Common among these models are bottom-up, stimulus-driven systems that construct a list of gaze locations ranked according to visual saliency. They predict human gaze locations in images based on the degree of saliency [59][60][61][62][63][64][65].

For instance, Itti et al. [60], constructed a visual saliency map by combining multi-scale image features into a single topographic saliency map. They used a winner-takes-all neural network to detect the next attended location in the image in the order of decreasing saliency. Likewise, Gao et al. [61] also used a visual saliency model for character recognition in natural scenes, such as in billboards and signboards. They deduced that characters have different visual properties from their non-characters neighbours which

make them more salient. However, in some situations, characters belonging to scene text might not be as salient. For example, a signboard is usually very salient, but the characters on it may not be as salient globally. They proposed a hierarchical saliency method that improved on the conventional saliency map model in the character-detection experiment. Furthermore, Perazzi et al. [66] used contrast-based filtering in determining salient locations in an image. Their model decomposes an image into homogeneous elements that abstract unnecessary details, and computes two measures of contrast that rate the uniqueness and contribution of these elements. The common trend in all the above models is that they process the entire image in order to determine the most salient locations, and use the ranking of the order of saliency to determine the eye movement. This kind of model does not closely model the active vision system in that it generally gives a set of pre-defined features which are exploited by the attention model. They do not consider that the type of features extracted each time depends also on the sensory-motor and behavioural characteristics of the organism in the environment (Floreano [10], Croon [25]).

The second models among the attention methods are those that do the determination of the gaze movements independently of a task [67][68]. These methods are mainly devoted to the modelling and prediction of eye movements and the focus is not actually to solve any specific task. For instance, Torralba [67] proposed a top-down attention method that uses contextual and scene information for attention guidance based on the global scene configuration. It was shown using the scheme that statistics of low-level features across an image can be used to prime the presence or absence of objects in a scene and predict their locations, scale and appearance before exploring the image. Also, Zhang et al. [68] proposed a system based on a Bayesian framework that constructs a visual saliency map which is used to predict fixation locations of people involved in the free viewing of an image. Unlike the existing saliency measures which depend on the statistics of a image being viewed, their measure of saliency is derived from natural image statistics, obtained in advance from a collection of natural images. In the same vein Itti and Baldi [59] developed a model that can predict a low-level surprise at every location in a video stream. The algorithm significantly correlates with two humans watching complex video clips which includes television programs of 17936 frames and 2152 saccadic gaze movements. The system allows more sophisticated and time-consuming image analysis to be efficiently focused only on the subsets of incoming data. On the other hand, Boriji and Itti [69] used top-down information for the model of observers playing 3 video games (driving, flight combat and time scheduling) using a dynamic Bayesian network to infer probabilistic distributions over attended objects and spatial locations directly from observed data to determine gaze locations. The common trend among this second

group of attention methods is that the system tries to model the attention locations in comparison to a human subject and they are not actually used to solve any specific task.

2.3 Active Vision Models

Various active vision models have been proposed in the literature that select their actions (eye movements) in different ways and mostly for a specific task. For instance, there are models for detecting edges (e.g. [70]), for controlling the gaze of a simulated fish (e.g. [71]) and for detecting an object in a visual scene (e.g. [72]). However, there are also others that are instances of a more general approach to active vision. We have distinguished two general approaches as: (i) the probabilistic approach [73][74][75][76][77][78][79], and (ii) adaptive approach [6][12][21][10][11][30][24][34][74][80][81].

2.3.1 Probabilistic Approach

The central aim of the probabilistic models is to reduce uncertainty in the world state. It regards active vision as a series of iterative steps of state estimation and action selection, and therefore uses a pre-determined probabilistic framework for action selection [82]. All the probabilistic models have one thing in common: they take action with the goal of reducing uncertainty in the belief state but they use different strategies in their action selection [25][82]. We distinguish probabilistic active vision models into three major groups described in what follows.

The first group of models calculates the expected usefulness of all actions on the basis of mutual information and then select the best one for actual execution [75][73][77]. For instance, Dames Amauric and Marchand [75] proposed a mutual information based system for a vehicle visual navigation that does not rely on an expensive feature extraction technique, matching, and tracking of geometric features such as key-points. Their model instead maximises shared information between the current image and the next key image in a visual path which it uses for successive visual navigation. Their system was tested in simulation and in a real vehicle. In the same vein, Huber et al. [83] provided a probabilistic approach to active vision using a Bayesian model to actively select camera parameters to recognise an object from a finite set of object classes. They used a Gaussian process regression to learn the likelihood of the image features from the object categories and the camera parameters where the object recognition task was treated as Bayesian state estimate. In order to improve recognition accuracy, the selection of the appropriate parameters was formulated as a sequential optimisation problem. The minimisation of the state estimation uncertainties was achieved using mutual information

which maximises the information from camera observations. Furthermore, Pirrone [44] developed an active vision system for classification of indoor environments, such that it could distinguish a bedroom from a kitchen. The system uses context-free and context-dependent analyses to infer high-level scene properties from low-level image features by identifying the probabilistic characteristic connected to the objects contained within the environment and defining the mutual probabilistic relationship and properties.

Among the second group of models are those that learn their action policy on the basis of entropy loss in the belief state (e.g. [78][76][84][85]). For instance, Ramanathan and Pinz [86] presented a multi-view approach to object categorisation using a humanoid robot (Nao) platform. The robot was presented with various 3D objects by a human-operator. Hand and head motion were used by the robot to actively obtain several different view points, and a view-planning scheme that uses entropy minimisation was used to reduce the number of views required in order to achieve the categorisation task. The results, obtained on a database of 3D objects of 4-classes, shows that the multi-view approach attained a significantly higher level of performance as compared to a single-view approach. Also, Porta et al. [78] used an efficient entropy reduction method for robot localisation, where the robot can execute actions with the sole purpose of gaining information on its localisation in an environment. While, Seekircher et al. [84] proposed a model according to which estimation of the robot's world can be improved by actively sensing the environment through consideration of the current world estimate, and therefore reducing the entropy of the underlying particle distribution for active control of the robot's head.

Lastly, the third group of models are those that rank all actions in advance e.g. for a class and execute the action that has the highest ranking for the most probable class ([87][88][89][79]). For example, Browatzki et al. [87] developed an active vision approach for a humanoid robot (iCub) to resolve the view-point problem in 3D-object recognition. They proposed an active vision gaze planning algorithm to obtain and optimise the best view-point that may be selected, among infinite viewpoints in a 3D scene, in order to facilitate the recognition process. This was done to resolve the usual visual ambiguities that are common to a view-point in a 3D-object. Their method was inspired by the fact that humans effortlessly resolve this ambiguity with proprioceptive information to augment the information obtained from the current view-point, and based on this, move to the best view- point location (e.g using their hands, heads, and bodies). To illustrate the usefulness of their work, the active system allows an efficient in-hand object exploration and perception-driven recognition process. In the same vein, Arbel and Ferrie [88] used gaze-planning that employs an entropy map to guide mobile observer: from a single monochrome television camera for recognising objects in an unstructured

environment, the observer is guided along an optimal trajectory that minimises the ambiguity of recognition.

The common theme among these models of active vision is that they all make some explicit assumptions for the eye movement as a predefined probabilistic framework.

2.3.2 Adaptive approach

Adaptive approaches do not use assumptions for their action (eye movement) strategy, but they are progressively adapted in order to optimise the performance of the task at hand. That aside, there are additional predefined attributes which also impose some limitations, such as the choice of the controller (e.g neural network) and the optimisation technique. However, in this model the goal is not to predetermine what the active vision system does internally. Typical tasks executed by these models are behavioural classification and control.

For instance, in Harvey et al. [35], an evolved neural network was applied to a robot in a real world: both its neural network control system and visual morphology were evolved to perform a discrimination task by generating the correct behaviour. The robot was given a classification task of discriminating between a triangle and a rectangle drawn on the opposite wall of the arena in which it was situated. At the beginning of each 4 trials, the robot was randomly located at different positions and orientations, such that it was not biased towards any of the opposite walls. The best evolved individuals from 15 evolutionary run exhibited the behaviour of moving towards the triangular shapes and avoiding the rectangles. The robot performed the categorisation task by exhibiting a behaviour in which it had to move toward the target shape on the wall.

In the same vein, Kato and Floreano [24] investigated a similar task but of static images in which the simulated active vision system had to discriminate between triangular and square shapes corrupted with some noise. The evolved controller used a simple neural network without hidden units. Two units of the output layer encoded the two different geometric shapes, with the most activated unit being the correct response. This model was similar to that of Harvey et al.[35] in that the two systems discriminate geometric shapes based on visual features; however, the mode of the discrimination were different. The system in [35] used a behavioural method of discrimination of moving towards the desired shape for the discrimination task, while that of [24] used an encoding system of the output units for discrimination.

Marrocco and Floreano [30] also extended the active vision network architecture in [24] to an all-terrain mobile robot equipped with mobile camera. The camera (pan and tilt)

was autonomously controlled through evolution of the neural network controller. Just as in [35][24], the active vision system dynamically select relevant features in the visual scene for the vision task. However, the strategy used in [30] was to allow the camera to select the correct features enabling the generation of efficient navigation trajectories along which obstacles would be avoided, in contrast to the strategy of [35] and [24] in which active features selection were mainly used to enhance discrimination between two shapes.

Leopold et al. [90] used a combination of reinforcement learning and belief revision in the context of adaptive vision environment. The active vision model interacted with the environment by rotating objects depending on past perceptions with the aim of acquiring views which were advantageous for the requisite recognition demanded by object categorisation tasks. This active vision system differed from that in [35][24][30] in that the adaptive active vision process used a rule-based system and a numerical learning method.

On the whole, even though there are other approaches to adaptive active vision,such as reinforcement learning, the approach used in this thesis is based on Evolutionary Robotics [29] [91][33]. The system possesses desirable properties for our active vision model such as: (i) a semi-supervised nature that optimises its action strategy for eye movements oriented towards a desired known task; and (ii) different parts of the model such as visual extraction and the controller, can be optimised together.

2.3.2.1 Evolutionary Robotics

Evolutionary robotics is a research field that uses simulated evolution to produce robot controllers. The aim is to build dynamic robot control systems, in which behaviours exhibited when interacting with the environment are generated autonomously, without actually programming each individual behaviour. There are many methods that can be used to evolve controllers, such as: genetic algorithm [92][93]; genetic programming [94]; and evolution strategy [95]. Also, apart from the commonly used neural networks, other forms of robot-evolved controllers can be used, such as evolving rule-based control [96]. However, neural networks have the following desirable properties: (i) they are resistant to the noise that is often present in robot/environment interaction (Nolfi [92]); and (ii) the low-level primitives, such as synaptic weights and nodes, are very good for the evolutionary process and avoid undesirable choices made by a human designer (Cliff, Harvey and Husband [97]).

In this approach, an evolutionary process normally involves an initial population of different "genotypes" each of which codifies the control system of the robot that are

generated randomly. Each robot is evaluated in an environment and assigned a fitness score based on the ability of the robot to perform some task. The robots that have obtained the highest fitness scores are then allowed to reproduce by generating copies of their genotypes with the addition of random changes ("mutations"). The process is repeated until a desired performance is achieved (for methodological information, see Cangelosi and Parisi [98], Cangelosi [99], Nolfi et al. [12], Tuci [93], Suzuki [33], Chapter 3 and experiment chapters of this thesis).

The most natural way of applying evolutionary computation to robotics is to perform direct evaluation of control systems on real robot hardware. However, evolution is a long term process, which may require many control system evaluations to obtain satisfactorily results, leading to significant run-times. Also, the robots may enter some dangerous states in which the hardware may be damaged, especially in the early stages of evolution [100]. These issues have led most researchers in evolutionary robotics to first evolve robots in simulation and then transfer the best evolved individuals into real robots [100]. However, simulated evolution of a robot requires the designer to carefully choose the simulated conditions of the real robotic environment, to give a greater chance of transferring the learned skills to the real environment. One of such ways is to add noise: Reynoid [101] pointed out that, without adding noise to the simulation, evolutionary computation will find brittle solutions that would not work in real robots. Jakobi et al. [102] also discovered that if there is significantly more noise in the simulation than on the real system, then new random strategies become feasible that do not work in actual practice.

2.3.2.2 Evolutionary Active Vision System

Evolutionary active vision is a research area in which evolutionary robotics methods are used to design control mechanisms for vision systems that autonomously explore the environment to perform a visual discrimination task. The significance of using an evolutionary approach to active vision is to allow the model to evolve for itself action strategies for eye movements through dynamic interaction of the agent (controller) with the environment, rather than imposing restrictions by a model. This kind of system has the advantage of discovering strategies that are unknown to us, the designers, that may help in solving a given vision task ([25]). The vision system usually has a limited or restricted field of view [8]. In some models this limited view is divided into a central smaller high-resolution view (fovea) and a wider outer periphery area with radially decreasing resolution ([23]). There also exist some architectures that do not have a division within the visual field (e.g. [24][30][103]). These types of model of the active vision system have great advantages: (i) they reduces the computational resources required

to complete a vision task, since they only process information within this limited area; and (ii) the eye is guided to the local visual information within the visual scene that enhances the solution to a given vision task.

In general we have discerned two different areas of research based on the complexity of the controller (i) reactive systems that rely solely on sensory-motor co-ordinations in determining the motor behaviours; and (ii) proactive systems that are provided with neural mechanisms that allow the system to integrate sensory-motor information through time to internal states to co-determine the motor behaviours.

Reactive active vision systems

Active vision systems that rely solely on sensory-motor coordination are also known as reactive systems [104]. Reactive systems make use of controllers without an internal state and always reacts with the same motor actions to the same sensory states. In neural network contexts, they are mainly feed-forward neural networks with or without hidden units, but without any form of memory or recurrent connections.

For instance, Nolfi and Marroco [12] developed an active vision system in which mobile robots were able to visually discriminate between different landmarks. Individuals were evolved in simulation and tested on physical robots. The controller used a very simple feed-forward neural network without any form of memory. The sensory states were very ambiguous, i.e. a large and a short rectangular stripe, but nevertheless the evolved individuals were still able to visually navigate towards the right landmark (large stripe). This was only possible through the coordination of evolved individuals' sensory-motor components. For example, when the robots were initially placed in the north-east of the environment facing south, the robots rotated until they faced one of the two corners of the landmarks and started to move forward slightly on the right. This allowed the robots to lose visual contact more quickly with short stripe than with the large, and this allowed the robots to reach the large landmark by moving significantly more towards it.

Similarly, Schembri [40] implemented an active vision system using a simple 3-layer feed-forward neural network controller evolved with a genetic algorithm. In this experiment, the simulated agent moved in a 2D square arena populated with small and big circles randomly placed in a grid of 5 x 5 positions. There were 10 small circles and 10 large circles. The agent, represented also by a circle was provided with a linear array of visual receptors by which it was able to see the object in front of it. The goal of the agent was to hit as many small circles as possible and to avoid the big ones over the course of a lifetime that of 10000 simulation steps. The genetic algorithm was used to run 10 replications of the evolutionary run with different seeds. Analysis of the best individual showed that

the agent showed some behaviours that enhanced the categorisation tasks. In this case, the agent developed some exploratory behaviours that consisted of: circumnavigating their centres and moving slowly until an object falls within the receptive field; and then moving close to the object in back and forth oscillations. If the object was a small circle, the agent moved forward and hit it; and if it was a large circle, the oscillating behaviour ended with the agent distancing itself from the object.

Generally, the common features shared by the systems described above was that, despite their very simple architecture they were able to use their intelligent sensory-motor coordination to select sensory patterns that were favourable to the given vision tasks.

Proactive active vision systems

Reactive systems that use strict sensory-motor coordination in determining motor behaviours are not common for vision tasks. In most situations the system may not be able to find stimuli regularities that can be used to solve the problem through sensory-motor coordination alone. To bridge this gap it will also need the addition of internal state dynamics of the network to integrate partial discriminative visual evidence over time [23]. In that case, the active vision system must have recourse to more complex strategies based on the internal states in addition to sensory-motor coordination. Proactive systems extract internal states by integrating sensory-motor information over time and later use these internal states in modifying their behaviour according to the current environmental circumstances [105]. Most works in evolutionary active vision systems use some form of internal states; however, the complexity of internal states also varied which may be due to the nature and complexity of the vision task.

Firstly, there are some systems in which the internal states are determined solely by the recurrent connections or feedback of memory provided in the controllers and may or may not have hidden layers [24][30] [57][10][34][106].

This is the case of an active vision system in Kato and Floreano [24] that autonomously interacts with different 2D shapes (triangles or squares). The controller of the system has a very simple discrete time recurrent neural network architecture, with no hidden nodes, and was evolved by a genetic algorithm. The active vision system was able to discriminate between different shapes irrespective of their locations and sizes in different trials by developing a behavioural strategy of exploring different areas of the shapes in order to enhance the categorisation task. In this case, the best evolved individual exhibited two behaviours in which: (i) the retina slides back and forth along the vertical edges of the shapes; and (ii) the retina scans the corners of the edges to enhance the discrimination tasks.

In the same vein, Marocco and Floreano [30] extended the simple active vision model in [24] for a robot navigation problem posed for a mobile robot equipped with a pan and tilt camera. The robot was positioned in a square arena and asked to navigate as far as possible without hitting a wall. The evolved robots were able to solve this problem by exhibiting a behaviour where they select simple visual features and actively maintain them on the same retina position. This kind of behaviour exhibited by the evolved robots was able to simplify the recognition task, in order to generate efficient navigation trajectories. The evolved robots developed behaviour for navigation that maintain the edge between the floor and the wall in sight of the camera.

Furthermore, Peniak et al. [57] evolved an active vision system of similar architecture to [24][30], that had the ability to navigate and avoid obstacles in unfamiliar and unstructured environments of planetary terrains. The active vision system was implemented on a 3D simulated model platform, the Mar Science Laboratory (MSL) rover. Simulated test environments were also generated to model the planetary terrain that had various obstacles such as rocks and holes on a very rough terrain. The free parameters of the controller were evolved by a genetic algorithm. The results obtained from 5 evolutionary runs showed that the evolved robots developed effective behaviours that allowed them to navigate in the environment and to avoid obstacles of different kinds (rocks and holes) by relying on the active vision system. The behaviours exhibited by the evolved individuals in which active cameras were used to select features that allowed them to maintain the correct navigation trajectory and to detect obstacles had some resemblance with that of [30], in which the evolved individuals used strategies of detecting edge features between the dark floor and white wall arena in which they were located to maintain a successful trajectory for navigation.

The common theme with these active vision systems is that even though the controllers have very reduced internal states in the form of only recurrent connections or memory feedback, by their dynamic interactions with the environment, however, they were able to generate behaviours that allowed them to exploit regularities in way appropriate to the vision tasks.

There are also active vision systems that have more complex internal sates, such as those that are provided by Continuous Recurrent Neural Networks (CTRNN) [23][25][107][108]. In this case, in addition to the recurrent connections, the neurons also have some dynamics that realises internal states.

For instance, Mirolli et al. [23] used an active vision system with a 3-layer Continuous Recurrent Neural Network, which was evolved by a genetic algorithm. The active vision system was given the task of categorising five italics letters at different scales (sizes), i.e. 25 sizes in the training stage and 50 sizes in the testing stage (re-evaluation). The

movement of the artificial eye was controlled by motor neurons of the output units, which determined the eye location per time step, in order to capture relevant input features for the neural network controller. The system was rewarded only for its ability to discriminate between the shapes of the letter and left free to determine how to explore the visual scene. Subsequent analysis based on the best individual of all replications of the evolutionary run showed that the agent was able to solve the problem by: (i) using sensory-motor co-ordination to generate behaviours that allowed the agent to experience visual regularities in different categorical contexts; and (ii) the integration of perceptual and motor information over time.

By way of further example, Guido de Croon [25] developed an active vision model that uses Continuous Recurrent Neural Networks for a car-driving simulation. Unlike the active vision system in [23], the system had a modular structure of two Continuous Recurrent Neural Networks, i.e. one controlling the eye movement and the other for controlling the movement of a simulated car. The output units of the eye controller determined the visual features that were being extracted as the car moved through a simulated road per time step which formed the corresponding inputs to the two controllers. The task of the agent was to drive over a simulated track as quickly as possible, while avoiding various obstacles on the way. The controller parameters were optimised with a genetic algorithm. Subsequent analysis showed that the system used the gaze shifts: (i) to find relevant features that contributed to successful driving; (ii) to keep relevant features in sight; and (iii) to avoid disruptive visual inputs while driving.

It was also noticed that there were some oscillatory kinds of movements exhibited by both the car body and gaze within a certain time-step range. Consequently, further analysis was done by fixing the visual inputs to confirm if the oscillatory movements were caused by the internal states alone since CTRNN are capable of complex internal dynamics. However, when the visual inputs were fixed the car went off track. This, according to them, showed that the active vision system uses the oscillatory behaviour to stay on track, especially when navigating curves on the road. It was therefore deduced that the oscillatory behaviour must have arisen as a result of a coupling between controllers and visual inputs from the environment, and this helped the car to successfully stay on track.

Finally, the common trend among these systems that used more complex internal states was that they used the additional internal states in addition to recurrent connections, and this helped the system to generate more complex dynamics for integrating sensory-motor information over time.

2.3.2.3 Evolutionary active vision system: from 2D to 3D in categorisation

Categorisation of objects by artificial systems such as robots is extremely difficult. The main challenge is that regions in the robot input sensor space that belong to the same category are not located contiguously or in close approximation, but are rather scattered. Also, regions that correspond to different categories may be located not only close to one another but also sometimes over-lap. The aim of evolutionary active vision is for the system to act in intelligent ways so as to experience sensor information that is not ambiguous in the input space but can be uniquely associated with a particular categorical context.

In most active vision systems, sensory-motor coordination alone may not be enough to solve object categorisation tasks; such systems will also require the integration of perceptual information over time through internal state dynamics of the controller. One such active vision systems is that of Mirolli et al. [23]. In particular, the complexity of their task was due to: (i) the large number of categories that were involved as compared to other evolutionary active vision systems; (ii) the possibility of sensing only a part of the object that was being categorised; and (iii) the differences of scales in each category. The agent therefore had to employ the extra internal state dynamics of the system in order to complement the sensory-motor strategies by integrating the perceptual-motor information over time.

Our work extends on Mirolli et al. [23] work with pre-processing techniques for more complex 2D images taken from the camera of the iCub robot. The pre-processing was inspired by the low-level processing that takes place in the human visual cortex [109], in order to have improved categorisation capability, given the strong dependencies between visual perception and eye movement.

Furthermore, we have extended their work with pre-processing for object categorisation in 3D using the iCub platform. We have chosen the humanoid platform because it will enable us to demonstrate this kind of categorisation problem with our method in a complex robotic system. Subsequently, we extended the work also for indoor and outdoor environment categorisation in 3D using the same humanoid platform. Our work differs from previous studies of environment categorisation in the following ways: (i) previous studies on indoor-outdoor environment categorisation were mainly oriented towards image and video classification, database retrieval, and the like, and not on a humanoid robotic system [41][42][110]; and (ii) previous work on active vision systems for scene classification (e.g bedroom from kitchen) uses assumptions, such as “carpets are laid on the floors”, “beds are in bedrooms”, “sinks are attached to the wall” (see Pironne [44] on indoor environment categorisation). Our model does not use these kinds

of assumption for its eye movement but is based on free exploration of the active vision to obtain relevant information for a specific task. In this way, it will give freedom to the active vision system in developing novel strategies for solving a particular problem rather than imposing some fixed assumptions.

2.4 Object categorisation

Although this thesis is about learning the control of active vision for categorisation using an evolutionary approach and not about categorisation itself, we feel it necessary to discuss here the subject of categorisation. Although there is an extensive literature on object categorisation, here, we review the literature mainly in the context of this PhD research work.

Object categorisation is a generic type of object recognition, in that it involves the recognition of an object from among many categories of object, by contrast, object recognition involves identification and recognition of the same category. Object categorisation inherently faces most of the challenges of object recognition, such as: view-point variation, illumination, occlusion, scale, background clutter etc. It also faces problems that are specific to categorisation, such as intra-class variation and inter-class dependence. These issues clearly make categorisation a non-trivial problem. Humans, however, find categorisation very easy [111], while machines such as computers and robots find it very difficult [111]. The computer models used in solving the problem of categorisation are either passive or active. The existing passive approaches involve scanning of the entire image, in which local image samples are not intelligently used to guide the process of categorisation. This makes them computationally inefficient (e.g window-scanning method [112][113] and the constellation method [114]). Generally speaking, the passive models of object categorisation in computer vision could be divided into generative models [115][116][117][118][119] and discriminative models [120][121][122][113][123]. Typically in passive methods of object categorisation, the first thing that is considered is how one can best represent object categories in images, using feature descriptors, key-point detectors, salient points etc. Then generative or discriminative models are learned from these representation. Based on these models (generative or discriminative), a new set of images can then be correctly classified for the object categories.

Discriminative models

Discriminative methods generally differ from generative models in that they attach no importance to the image's surface appearance but instead focus on the category itself.

They try to map the category of an image directly to an image sample without consideration of image details. Also, discriminative models seems to have higher classification accuracies when similar categories have to be distinguished because each model is created for each category [111]. There are approaches that learn discriminative models from bag of key-points such as [124][125]. These approaches do not make use of any geometric information about the key-points in the images. Other approaches learn discriminative models based on Support Vector Machine or nearest-neighbour classification [121][126]. In general, even though all these discriminative models can be used for categorisation, they are different from our active vision model in that there is no intelligent control over the way the local image patches are being selected for processing.

Generative models

Generative models give consideration to the details of the image, for example, the geometry. Categories are described as joint probability distributions of local salient patches and shapes [111]. Generative models have some advantages over the discriminative models, such as: prior knowledge can be integrated; new categories can be added; many categories can be represented; and handling of correspondences between objects parts can easily be accomplished [111]. Some generative methods of categorisation uses "bag of visual words" also known as bag of visual words models [127][128][129][130] [131][132][133][134][135][136][137], others are part-based models [114] [138][139] [140] [141][142][143][144][145][146] and window-sliding models [147][148][149][150][151]. Some discussion of bag of visual words and part-based models will now follow.

Bag of visual words models: In bag of visual words models images are divided into different parts without consideration of the initial location of the parts or the geometric relationship between them. In this approach salient points or parts are extracted from images and descriptors are calculated to form a feature vector. The parts are put into a code-words dictionary from which a classifier is built. Prominent among these models are: (i) Probabilistic latent Semantic Analysis (pLSA) [127][128] and (ii) Latent Dirichlet Allocation (LDA) [129][130][152].

(i) Probabilistic latent Semantic Analysis (pLSA): In these models the probability of each co-occurrence of word (image parts) and the image itself as a mixture of conditionally independent multinomial distributions. They employ a passive approach in the detection of features and their descriptors.

(ii) Latent Dirichlet Allocation (LDA): This model allows sets of observed words (image parts) in an image to be explained by unobserved categories, where each image is seen

as a mixture of a small number of categories and each word creation is attributable to one of the image's categories.

Part-based models: These models make use of a lot of geometrical information. They also make use of a lot of prior knowledge by applying priors to the parameters of the images. Most common among these are constellation models ([146][153][154]) which employs geometry in terms of spatial relations between key parts of objects models. They usually proceeds in two major stages: detection of key-points and constellation evaluations. An object is recognised if there is a constellation of recognised parts that is sufficiently similar to a learned constellation object model. Others are sliding-window methods ([148][155]): they scan passively to check for object presence at all locations of an evenly spaced grid and extract a local sample at each grid point to classify as either an object or as a part of background. Other examples of part-based models are one-shot learning methods proposed by Fei-Fei et al.[156], that aim to learn information about categories from one or few training images.

On the whole, all the models that we have discussed above used exhaustive scanning at one time or another and the local image samples are not intelligently guided to the next sample location. They are therefore different from the active vision models that have been discussed in the previous sections.

2.5 Environment categorisation

In this section, we review some studies that have been conducted in environment categorisation. They mainly focus on outdoor environment for scene categorisation, such as forest, beach, urban areas etc. [43][157][158][159][160][161][162][163] and indoor scene categorisation such as bedrooms, kitchens, dining rooms, sitting rooms etc. [164] [165] [166][167][168][169] and 2D image classification for indoor and outdoor environment [170][171][172][173][174] [175][176][177][178][179][180]. The applications range from classifying the environment location of smart-phone devices [181][182], scene categorisation by mobile robots ([183]) and indoor environment categorisation [168]. Environment categorisation is a more difficult problem than ordinary object categorisation in that there are more variables involved in each image to be considered in terms of colour, texture and structures (objects). Also, unlike object categorisation which mostly involves an instance of an object in an image, there could be multiple instances of distinct objects in an image and each object may be in a different spatial location. There could also be the problem of multi-labels, where an image may contain multiple labels or categories ([184]). For instance, a beach environment may contain a mountainous background and may be properly labelled as a mountainous-beach environment. Various approaches have been

used in the problem of environment categorisation such as models that involve either generative or discriminative learning or a hybrid of these two approaches ([43][158][41]). There are approaches that use varied representation techniques for feature description (e.g. [157]).

For instance Bosch et al.[43] uses a probabilistic Latent Semantic Analysis (pLSA) in an unsupervised manner to first discover objects in images that contain multiple object (categories). The pLSA was applied to a bag of visual words that described each image in the data-set, and the object distributions were then used to perform outdoor scene classification using k-nearest neighbour classifier. Zou et al. [157] proposed a method for scene classification that used collaboration representation fusion with local and global features. In their method, a visual word code-book was first constructed by dividing an image into dense regions, linear coding was employed on the dense regions via the code-book and a pyramid matching strategy was then used to combine local features. A method known as multi-scale completed local binary patterns was used to extract global features. Kernel collaborative representation based classification was then finally applied to the global and local features extracted and the class label of the testing image was given according to the minimal approximation residual after fusion.

On the other hand, there are approaches that categorically distinguish between indoor and outdoor environments that are used for smart-phone devices. For example, they are able to automatically locate the environment of the current user and determine the signal strength in these environments for Global System Networks (see [181][182]). Other research into categorising indoor and outdoor environments focuses on image and video retrievals in databases (e.g. Szummer and Piccard [41], Yailaya et al. [185], Luo et al. [42], Balal et al. [186]). There are also studies on robotic application used to determine the operational environment of an Un-manned Ground Vehicle (UGV) (e.g. [183]).

In general, all of the methods described are unsuitable for a robotic system because in most cases exhaustive processing is involved; our proposed gaze control model by contrast uses intelligent control of the eye to optimise visual resources.

2.6 Chapter Summary

We began this chapter with a review of the current gaze control models and pointed out that, whereas not all gaze control models are active vision models, all active vision models are gaze control models. A gaze control model cannot be classified as an active vision model if the model processes the entire image by defining some predetermined features for the entire image.

We noted that the two major types of active vision models in the literature are probabilistic and adaptive. The probabilistic models make use of some predefined framework that define an iterative process of state estimation for its actions (eye movements). Our model belongs to the second group, "the adaptive model", which does not require the designer to pre-determine what the model should do, but generates an active vision model that progressively gets better at its task.

Evolutionary active vision systems uses intelligent sensory-motor interaction in order to experience stimuli that will enhance the given vision task. However, in most vision tasks, the system also uses internal state dynamics to complement the intelligent sensory-motor co-ordinations. Typical among these tasks are categorisation tasks when stimuli that belong to each category are very ambiguous or when the number of categories are considerable, with large variations in things such as scale and orientation.

Most previous work using evolutionary active vision systems for categorisation have been used for simpler problems in 2D environments. Our work builds on Mirolli et al. [23] which deals with a considerably large number of categories and complex images as compared to the previous evolutionary active vision systems. We extended their work with pre-processing for more complex images taken from the camera of a humanoid robot (iCub). We further extended their work with pre-processing to the 3D environment for object and environment categorisation with the humanoid robot platform. In the next chapter, we discuss our Gaze control framework and the methods used in the implementation.

Chapter 3

Gaze Control Framework and Methods

3.1 Introduction

In this chapter, we look at the requirements for the design of our active vision gaze control framework, the gaze control framework and the computational methods that were used in the implementation. In Section 3.2, we highlight and discuss the requirements for the design of our framework. In Section 3.3 we discuss the gaze control framework. Section 3.4 describes the controller for the active vision model, while Section 3.5 discusses the optimisation method. In Section 3.6 we discuss the visual extraction methods that were used for sensory representation. Section 3.7 gives a discussion of the iCub platform, vision kinematics and the integration of the evolutionary methods. In Section 3.8, we discuss the software libraries and platforms used in the implementation of the experiments. Finally, in Section 3,9, a summary of the chapter is given.

3.2 Requirements for the gaze control framework

In the active vision literature, we were able to distinguish some basic requirements of an active vision system for classification tasks. We have used these requirements as guidance for the design of our gaze control framework. We highlight and discuss these requirements below:

1. A foveating vision system that processes a restricted part of the image per time step based on the motor response. Normally the sensor is modelled as having high

resolution at the centre and decreasing low resolution at the periphery, as used in [23].

2. A pan and tilt movement for the active vision system which is influenced by the input features and/or the internal state of the model.

3. A classification module that determines the categorisation tasks.

4. A visual representation module that pre-processes the visual stimuli using a grey-scale averaging method as in [23][24] or Histogram of Oriented Gradients (HOG)[2] or Uniform Local Binary Patterns (ULBP)[1].

5. Feedback of visual information, sensor values (e.g in the form of pan and tilt rotation angles) and category estimates.

3.3 The Gaze Control Framework

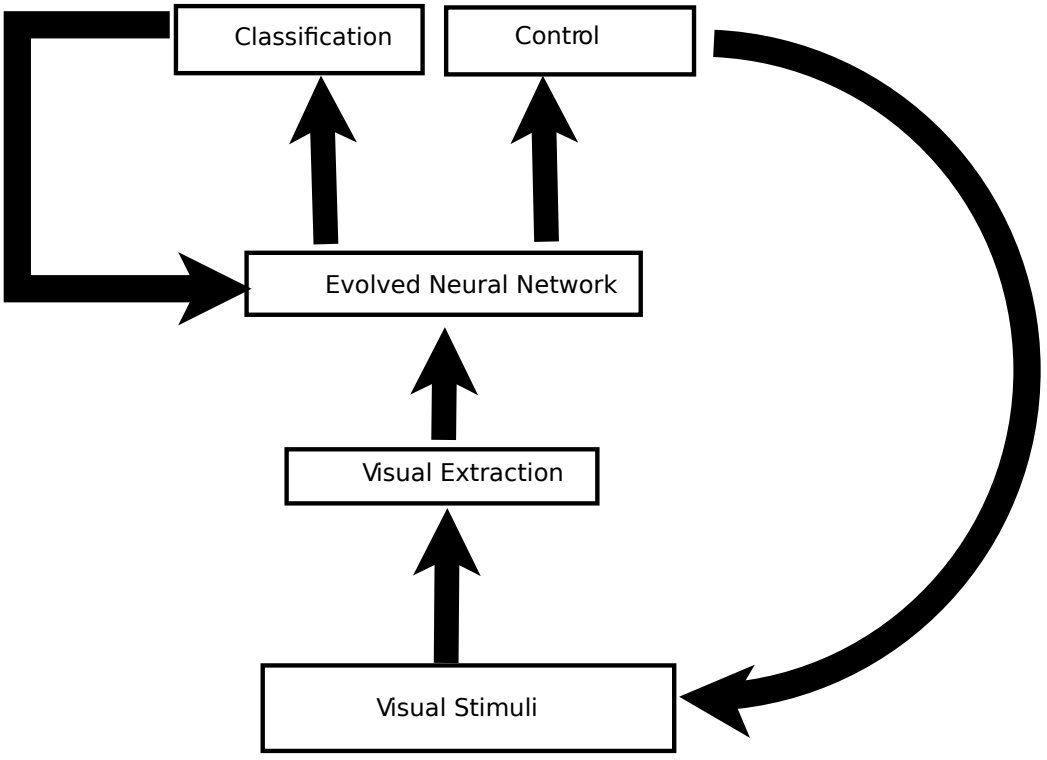

FIGURE 3.1: The Gaze Control Framework

The gaze control framework is inspired by the model in Mirolli et al [23] and is supplemented with the neural network update equations of Tuci [29] (Fig. 3.1). We have built our framework on their periphery-only architecture, which gave the best performance among all the architectures experimented with in [23]. The periphery-only architecture implies that there is no division between fovea and the periphery and that the visual

inputs neurons are connected to both the motor neurons and the internal neurons. They also mentioned that based on their investigation of the different architectures, separate fovea and periphery units may not be necessary in improving categorisation performance. Since our emphasis in this thesis is on improving performance and enhancement with pre-processing techniques, we do not find it necessary to replicate the fovea architecture, which was done in [23] for 2D object categorisation. However, we extended this framework with pre-processing techniques for visual representation. The active vision system autonomously takes an input from a visual scene restricted by the active window. The visual stimuli are processed by a visual extraction method and are mapped by an evolved neural network controller to gaze shifts and classification units. The visual extraction module is processed by either a grey-scale average method as used in [23] or pre-processing techniques [2][1] are adopted. The gaze shifts which enhance the performance of the task are determined by the visual features, previous gaze shifts/categorisation outputs at time $t-1$, and/or the internal state of the controller. At the new gaze location the process of feature extraction and gaze shifting are repeated. The iterative process stops when a stopping criterion is reached.

3.4 The Controller

We have used a neural network controller, and our justification for this is based on the following: (i) a neural network resembles natural vision systems in the sense that it processes information in a distributed manner; (ii) it can be extended to include memory and other dynamic capabilities; (iii) it can handle a continuous flow of input and output; and (iv) it is well studied and implemented as a controller in robotic research.

In all of our experiments, we predefined a specific architecture for the neural network. However, there are other methods that optimises both structure and the weights (e.g. Mattiussi and Floreano [187], and Stanley and Miikkulainen [188]). Even though such methods have met with some success, they are not guaranteed to give better solutions than methods that use predefined structures (Floreano, Dürr and Mattiussi [189]).

The gaze control model uses a continuous time recurrent neural network similar in design to Mirolli et al. [23], but with similar update equations as used by Tuci [29]. It has three layers: (i) an input layer, whose vector size is determined by the visual feature extraction method, and a copy of the motor/gaze control units and classification units at the previous time step; (ii) recurrent hidden layer units; and (iii) an output layer of motor/gaze control units and classification units.

The values of the input, hidden, and output neurons are updated using equations 3.1, 3.2 and 3.3 respectively. In these equations, using terms derived from an analogy with real neurons, y_i represents the cell potential, g is a gain factor, τ_i the decay constant. I_i with $i = 1, ..., n-1$ is the activation of the i^{th} input neuron. Also, $i = n....k-1$ and $i = k....u$ are the range of the number of hidden and output neurons respectively. w_{ji} is the weight of the synaptic connection from pre-synaptic neuron j to post-synaptic neuron i. β_j is the bias term and $\sigma(y_j + \beta_j)$ is the firing rate. All input neurons share the same bias β^I, and the same holds for all output neurons β^O. $\sigma(x) = \frac{1}{(1+e^{-x})}$ is the sigmoid function. The decay constants, bias terms, weights and gain factor are genetically specified network parameters. We approximated the dynamics of differential equation 3.2 using the standard forward Euler method with an integration time step ΔT = 0.1.

$$y_i = gI_i; i = 1, ..., n-1 \tag{3.1}$$

$$\tau_i \dot{y_i} = -y_i + \sum_{j=1}^{j=k-1} w_{ji}\sigma(y_j + \beta_j); i = n, ..., k-1 \tag{3.2}$$

$$y_i = \sum_{j=n}^{j=k-1} w_{ji}\sigma(y_j + \beta_j); i = k, ..., u \tag{3.3}$$

3.5 Optimisation method

Although, there are other adaptive optimisation algorithms such as reinforcement learning [32][190], simulated annealing [191][192], cross-entropy search [193][194] and random search[195][196], we have chosen an evolutionary algorithm [197][198][199] for our gaze control model based on the following reasoning:

(i) Evolutionary algorithms have been shown to perform better than reinforcement learning for ambiguous visual inputs [200]. They also allow for the optimisation of any part of the model that can be parametrised, while reinforcement learning focuses solely on action strategy. For instance, evolutionary algorithms can be used to optimise visual features and the action strategy simultaneously [25].

(ii) Random search does not exploit any structure in the search space which makes it very inefficient [25].

(iii) Evolutionary algorithm, cross-entropy search and simulated annealing are very similar optimisation algorithms. However, our choice of evolutionary algorithm is based on the premise that it has been consistently proven with good results and, as such, is a common choice in the field of Evolutionary Robotic Research [6][92].

3.5.1 The adaptive task and the evolutionary process

In this section we explain the adaptive task and the general evolutionary framework, where the specific instances are described in the experiments in Chapters 4, 5 and 6.

In each trial of the evolutionary adaptation process, the artificial eye (active window) is left to freely explore the visual scene in the first part of the trial. The task of the active vision agent is to correctly classify an object when it has explored the image for a sufficient length of time, that is during the second half of a trial. The agent is evaluated by the fitness function F as used in Mirolli et al.[23], and is comprised of two components: the first, $F_1(t,c)$ rewards the agent's ability to rank the correct category higher than the other categories; the second, $F_2(t,c)$ rewards the ability to maximise the activation of the correct unit while minimising the activations of the wrong units, with both terms given equal weighting:

$$F = \frac{\sum_{t=1}^{T}\sum_{c=S}^{C}\left(0.5 * F_1(t,c) + 0.5 * F_2(t,c)\right)}{T * (C - S)} \tag{3.4}$$

$$F_1(t,c) = 2^{-rank(t,c)} \tag{3.5}$$

$$F_2(t,c) = 0.5 * y_r^{t,c} + \sum_{w \in \Omega}\left(1 - y_w^{t,c}\right) * \frac{0.5}{N-1} \tag{3.6}$$

where $F_1(t,c)$ and $F_2(t,c)$ are the values of the two fitness components at time step c of trial t, $rank(t,c)$ is the ranking of the activation of the categorisation corresponding to the correct category (that is, from 0, meaning the most activated and l, meaning the least activated: where l is 1 less than number of categories), $y_r^{t,c}$ is the activation of the output corresponding to the current (correct) category, $y_w^{t,c}$ is the activation output of the wrong category w at trial t and time step c (where Ω is the set of wrong categories). N is the number of categories, T is the number of trials, C is the number of time steps in a trial and S is the time step in which we start to compute fitness.

The free parameters of the neural controller are adapted through a genetic algorithm. The initial population consists of n randomly-generated genotypes, each encoding the free parameters of the corresponding neural controller, which include all the connection weights, gain factors, biases, and time constants. The genotypes encoding for the free parameters of the agent controllers are vectors comprising of n real values chosen with uniform randomness from the range $[0, 1]$. In order to generate the phenotypes, weights and biases are linearly mapped in the range $[-x, x]$ and $[-y, y]$ respectively, while the time constants are mapped in $[-t1, t2]$. Note: the "variables values" of the evolutionary framework and other related details are given in the specific implementation of the framework in Chapters 4, 5 and 6.

3.6 Visual Feature Extraction

We used the following visual extraction methods in all the experiments in this thesis: the grey-scale averaging method [23], Uniform Local Binary Patterns (ULBP) [1] and Histogram of Oriented Gradients (HOG) [2]. We have chosen ULBP and HOG because they are simple to implement as well as their usefulness as feature descriptors in many computer vision applications, such as face recognition [47] and object detection [51]. It is very important to state here that we did not use the pre-processing methods mentioned above to process the entire image; instead we allowed the active vision to dynamically select an area to be processed per time step and afterwards used one of the visual extraction methods to process the pixels within the active window (Fig. 3.2). As such, we still keep to our philosophy of an active vision model that does not process the entire image or give a predefined set of features for the model but instead allows the system to actively select features through the dynamic interaction of sensory-motor components [57][25].

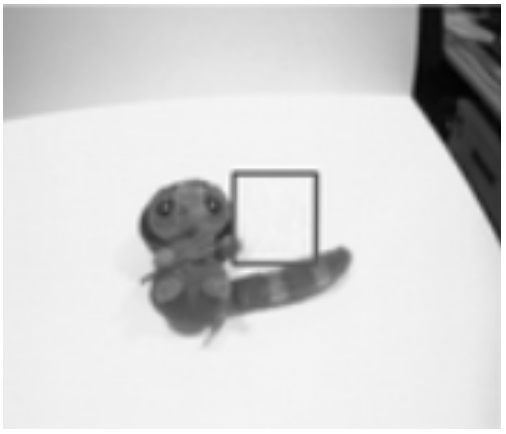

FIGURE 3.2: Shows the image of a soft-toy and the active window dynamically selecting the area to be processed as was done in our experiment (Chapter 4).

3.6.1 Grey-scale averaging method

In the grey-scale averaging method, the coloured image is first converted to a grey-scale image. The active vision model then takes a visual input from a gaze window of s x s pixels extracted from the grey-image of m x m in a time step. The window is sub-divided into k x k input cells and the average value calculated in each cell, resulting in k^2 visual inputs. The iterative step continue until a certain stopping criterion is met.

3.6.2 Local Binary Patterns

Local Binary Patterns (LBP) are a modification and improvement on a method used for texture classification using a texture spectrum by Wang and He [201]. Wang and He [201] proposed a model of texture analysis based on texture unit, where a texture image can be characterised by its texture spectrum. A texture unit is represented by eight elements each of which has possible values of (0,1,2) and which is obtained from a neighbourhood of 3 x 3 pixels. In this case there are a total of $3^8 = 6561$ possible texture units describing spatial patterns in a 3 x 3 neighbourhood.

However, Ojala et al. [202] proposed Local Binary Patterns involving a two-level pattern also in a 3 x 3 neighbourhood for a texture unit (Fig. 3.3). In this two-level version there are only $2^8 = 256$ possible texture units instead of 6561. It is a grey-scale invariant method and provides a robust way of describing pure local binary patterns in a texture. The reduced size of possible numbers of pattern (256 as opposed to 6561) originally proposed by Wang and He [201] makes it a more computationally efficient method to describe a texture region. The LBP operator computes the feature vector (descriptor) for an examined window of an image in these simple steps: (i) it divides the examined window into cells of y x y pixels; (ii) for each pixel in the cell it compares it to its 8 neighbours (i.e. in a clockwise or counter-clockwise direction) and where the centre pixel is greater or equal, considers the result as 1 or otherwise 0; (iii) it converts the resulting bit string to decimal; (iv) it computes the histogram of frequency of occurrence of the binary patterns in each cell (the histogram is a 256-dimensional feature vector) and optionally normalises the histogram (iv) it concatenates the histogram of all cells.

The basic version of LBP which considered only an eight-pixel neighbourhood can easily be extended to include all circular neighbourhoods with any number of pixels [1], where g_c represents the grey value of the centre pixel (x_c, y_c) of a local neighbourhood, g_p the grey value of P equally-spaced pixels on a circle of radius R. The values of neighbours that do not fall exactly on pixels are estimated with bi-linear interpolation.

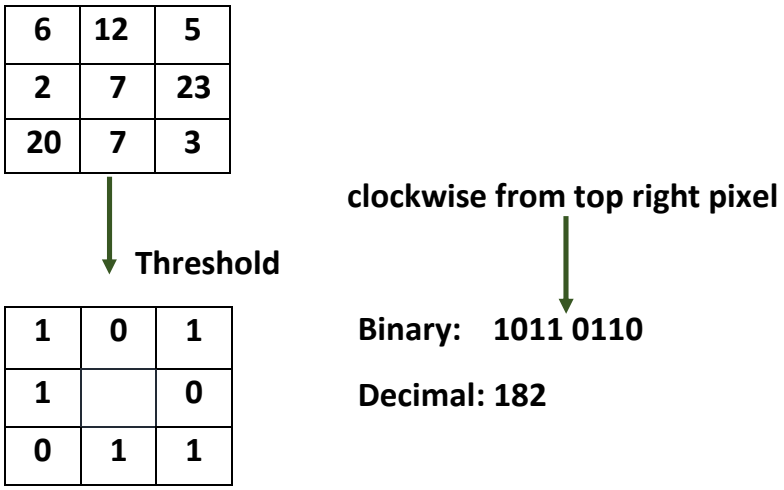

FIGURE 3.3: Illustrate the basic LBP algorithm which threshold the centre pixel in an image with its 8 neighbours in a clockwise direction and expresses the result as binary.

3.6.2.1 Uniform Local Binary Patterns

Uniform Local Binary Patterns (ULBP) are an extension of LBP that considers only uniform patterns. Uniform patterns of texture units are those that have a maximum of 2 bit-wise transitions, i.e from 0 to 1. For instance, in an eight-circle neighbourhood texture unit, bits patterns '00000000' (0 transition), '00110000' (2 transition) are uniform patterns, while non-uniform patterns such as '00010100' (4 transitions) and '00101010' (6 transitions) are not. In ULBP, there is a separate output label for each uniform pattern and one output label for all the non-uniform patterns. Thus, the number of output label for the mapping of patterns P is $P(P-1)+3$. For instance, ULBP produces 59 output labels for an eight-neighbourhood texture unit and 243 for 16 circular neighbourhood sampling points. There are two justifications for omitting the non-uniform patterns:

1. Most of the LBP patterns in natural images are uniform. Ojala et al. [1] found in their research investigation that about 90 percent of LBP patterns in (8,1) neighbourhoods are uniform patterns, and they account for about 70 percent in (16, 2) neighbourhoods.

2. Uniform patterns have proved to be more robust in terms of recognition results and less prone to noise in many applications [45][203]. Also considering only uniform patterns makes the number of possible LBP patterns considerably lower and therefore reduces the size of the feature descriptor.

Active-Uniform Local Binary Patterns

We therefore chose ULBP as a pre-processing technique for feature extraction of the active vision system based on the following reasons: (i) it has been proven with good recognition results in computer visions applications [45][203]; (ii) it has also proven to be less prone to noise in natural images [45]; and (iii) it has a lower feature descriptor size as compared with the original LBP which make it more suitable for our active vision model on grounds of computational efficiency.

However, because of the peculiar nature of active vision systems and the computational cost of evolutionary methods in training, we have implemented the ULBP method so that it will be suitable for the model. For instance, all forms of pre-processing have to be done within the active window (retina region) per time step, instead of processing the entire image. We also have to use a considerably reduced number of cells. We therefore prefer to term it Active-Uniform Local Binary Patterns (Active-ULBP). The Active-ULBP algorithm was implemented as follows:

1. An image was presented to the active vision model in each trial of the evolutionary run.

2. In each time step of a trial: (a) a Gaussian blur function was used to reduce the noise within the active window (retina region); (b) the retina region was divided into 4 cells and a histogram of uniform patterns of size 59 was constructed for each cell; (c) the histogram of each cell was normalised with an $L2$-norm scheme; (d) the normalised histograms of all cells were concatenated to form a feature vector of size 236; (e) the feature vector was combined with the copies of the movement and categorisation output units at the previous time step which formed the input vector for the neural network.

3.6.3 Histogram of Oriented Gradients

The Histogram of Oriented Gradients (HOG) descriptor was originally developed by Dalal and Triggs [2] for describing edges and gradients over a local image region using a sliding window over an entire image. It computes histograms over dense grids of uniformly spaced cells and contrast normalises for improved performance (Fig. 3.4). They are reminiscent of Sift descriptor [204], edge orientation histograms [205][206] and shape contexts [207].

In their research work Dalal and Triggs [2] used HOG as a local image feature set for human recognition in pedestrian image data set and using Linear Support Vector Machine as a classifier of the normalised histogram features.

The fundamental idea is that object appearance and shape over a local region can be characterised very well with intensity gradients distribution. The image window is divided into small spatial cells over dense grids. Histograms are computed for the cells and contrast normalised to form the feature sets. The general steps in their method of implementation are:

(i) The input image is optionally processed with gamma equalisation.

(ii) A detector window tiled with a grid of overlapping blocks is scanned across the image at all positions and scales.

(iii) A histogram is computed for each cell in each block using spatial orientation binning. Orientation bins are evenly spaced over $0-180$ degrees (unsigned gradients) and $0-360$ degrees (signed gradients). To reduce aliasing, votes (using magnitude of gradients in x and y direction) are interpolated bi-linearly between neighbouring bins in both orientations and positions.

(iv) Contrast normalisation of the histograms is done across overlapping cells in each block using various normalisation schemes such as $L1$-norm, $L2$-norm.

In their implementation they tested the effects of various parameters on the overall results of the descriptor and their findings were:

(i) Smoothing using Gaussian blur drastically reduces its performance.

(ii) Gamma normalisation has little or no effect on the different colour space used, e.g. RGB and LAB for the different colour channels.

(iii) Among the different derivative masks used, such as, 1-D point derivative (uncentred $[-1, 1]$ and centred $[-1, 0, 1]$), cubic corrected $[1, -8, 0, 8, -1]$, and 3 x 3 sobel masks. The simple 1-D $[-1, 0, 1]$ masks at $\sigma = 0$ worked best.

(iv) Spatial orientation binning is essential for good performance and performance increases up to 9 bins with extra bins not having an effect. Orientation bin space of 0-180 degrees gave the best performance, that is using only the unsigned gradients.

(v) An effective local normalisation scheme over cells grids is essential for good performance.

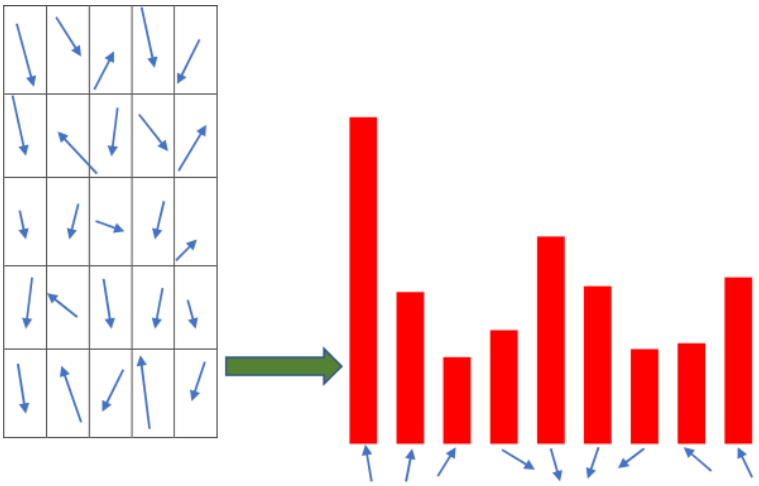

FIGURE 3.4: Illustate the Histogram Orientation (HOG) algorithm which calculate the gradient orientation and magnitude of each pixel of a cell in an image and adds the magnitudes into a corresponding bin of 9.

Active-Histogram of Oriented Gradients

We adopted the Histogram of Oriented Gradients [2] for the pre-processing of visual stimuli of the active vision in order to give improved visual representation, and on this basis to provide better control of the active vision. However, in the adoption of HOG in our model we considered two major factors: (i) the computational complexity of the pre-processing, since evolving a neural network will only be practicable with lower dimensional feature vectors; and (ii) suitability for the active vision concept, which processes a part of the image scene at each time step. Consequently, the HOG used in our model is a very simple version of the original algorithm and we prefer to call it Active-Histogram of Oriented Gradients (Active-HOG) because of its adoption in the Active Vision System.

Listed below are the differences from and similarities to the original HOG.

(i) Just as in the original HOG by Dalal and Triggs [2] we did not use any form of smoothing as we also noticed a reduced performance.

(ii) The original HOG processes the entire image. However, in our implementation we process only the restricted active window region per time step. This was done in order to reduce the computation complexity of our method and also to maintain consistency with the active vision concept.

(iii) In order to reduce computational cost, we did not use weighted overlapping cells, instead only four non-overlapping cells were used. This vastly reduced the size of our feature vector to 36 as compared to that of the original HOG implementation of 3780.

(iv) The preferred bin orientation space for the histograms in the original implementation was 0-180 degrees using only the unsigned gradients. However, in our implementation signed gradients of a bin orientation space of 0-360 degrees gave us better results.

(v) Normalisation over all the cells in the window with an $L2$-norm scheme also gave us improved results just as stated in the original paper [2].

We list the complete steps of the Active-HOG algorithm below:

(i) Input an image for each trial of the active vision evolutionary process.

(ii) In each time step of each trial perform the following process: (a) compute the gradients for each pixel in the active window in x and y direction i.e dx and dy; (b) divide the active window into 2 x 2 cells giving a total of 4 cells; (c) in each cell compute gradient magnitudes as $\sqrt{dy^2 + dx^2}$ and gradient directions as $\Theta = arctan(\frac{dy}{dx})$; (d) quantize gradient orientations into 9 bins with a bin size of 40 degrees of orientation space between 0-360 degrees; (e) add magnitude into each bin; (f) concatenate all histograms into a feature descriptor of dimension 4 cells x 9 bins giving a feature vector of size 36; (g) normalise the feature vector with $L2$-norm, i.e. $V = \frac{V}{||V||}$; (h) input a normalised feature vector into the neural network along with the copies of motor and categorisation outputs in the previous time steps.

3.7 The Gaze Control Framework: iCub platform

In this section, we introduce the iCub humanoid robot platform that we used for the implementation in the 3D environment. Here, we discuss the iCub vision and kinematics mainly in the context of its extension to the Gaze Control Framework. The iCub is a humanoid robot designed to simulate a 3.5 year old child and developed by the European Robocub research project [208][209]. The design of the iCub has two main goals: (i) provide a common platform for research in embodied recognition; and (ii) improve the understanding of cognitive systems by exploiting this platform in the study of cognitive development. The iCub is provided with the ability to learn how to interact with the environment through complex manipulation and how to develop its perceptual and motor capabilities for the purpose of goal-oriented tasks. It is about $90cm$ tall, weighs 23 kg and has a total of 53 degrees of freedom specified as follows: 6 for the head, 3 for the torso, 8 for each hand, 7 for each leg and 7 for each arm.

In our experiments (Chapters 5 and 6), we used a simple iCub simulator developed by Tuci [210] because of the computational overhead that would have been involved in using the original iCub simulator for our evolutionary method. However, the use of simulator

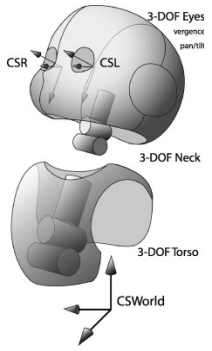

FIGURE 3.5: A simple illustration of iCub vision kinematics (image from [4])

might not provide a full model of the complexity present in the real environment and as such might not guarantee the transfer-ability of the controller from the simulation environment to the real one. That said, use of the iCub simulator (robotic set-up) has the following benefits:

(i) It will enable us to study the behaviour of embodied agents such as the iCub without facing the problem of maintaining a complex hardware device.

(ii) It can be a quick way of testing our algorithm in order to identify any problem before actual transfer of the controller to the real iCub robot. This also will allow us to study the plausibility of our models in such a complex robotic system.

3.7.1 iCub Vision Platform and Evolutionary Active Vision

The iCub vision has 3 degrees of freedom (DOF). Both eyes can pan for vergence and version control (2 DOF) and tilt simultaneously (1 DOF) Fig. 3.5. However, we only make use of the 2 degrees of freedom of the right eye (pan and tilt) and as such exclude the vergence and version control. We also exclude head, neck and other proprioceptive information from our experiments. This allows us to study the robustness of our system because of the limited degrees of freedom and exclusion of proprioceptive information that may enhance the recognition tasks.

In the evolving of the iCub system we only make use of the modules that are directly related to our experiments, i.e. 2 degrees of freedom of the right eye. In each time step of every trial in the evolutionary run, we calculate the tilt ($Tilt_{step}$), equation 3.7 and pan (Pan_{step}), equation 3.8 and normalise the updates initially in radians that go back as inputs ($Tilt_{input}$), equation 3.11 and (Pan_{input}), equation 3.12 to the neural network as follows:

$$Tilt_{step} = ((O_1) - 0.5) * MAX_{step} \tag{3.7}$$

$$Pan_{step} = ((O_2) - 0.5) * MAX_{step} \tag{3.8}$$

where O_1 and O_2 are the neural network controller outputs for the eye movements (i.e the tilt and pan) respectively. $MAX_{step} = ONE\_PI * 5$ is the maximum step for the pan and tilt in radian ($ONE\_PI = \pi/180$), and the value 0.5 is used to scale the tilt and pan eye movements for negative and positive angles (radians). For instance, an output of pan or tilt of 0.5 will give 0 radian (i.e when there is no eye movement in the pan or tilt).

$$Tilt_{new} = Tilt_{new-1} + Tilt_{step} \tag{3.9}$$

$$Pan_{new} = Pan_{new-1} + Pan_{step} \tag{3.10}$$

$$Tilt_{input} = \frac{Tilt_{new} - Tilt_{low\_limit}}{Tilt_{high\_limit} - Tilt_{low\_limit}} \tag{3.11}$$

$$Pan_{input} = \frac{Pan_{new} - Pan_{low\_limit}}{Pan_{high\_limit} - Pan_{low\_limit}} \tag{3.12}$$

Where the new tilt ($Tilt_{new} = \theta_6$), equation 3.9 and pan ($Pan_{new} = \theta_7$), equation 3.10 updates, and the link parameters, a and d (Table 3.1) are used to calculate the forward kinematics for the iCub right eye (equation 3.13), based on Denavit-Hartenberg notation. Note: link parameters i=6 and i=7 are at the end of the iCub kinematic chain, starting from torso, but we are only interested in the last two joints, i.e of the right eye, and so considered the offset from the head centre.

TABLE 3.1: Shows the link parameters a, d, α, θ of the iCub right eye (for the tilt i=6 and pan i=7), where a and d are in millimetres, and α and θ are in radians

Link (i)	a_i (mm)	d_i (mm)	α_i (radian)	θ_i (radian)
i=6	0	34	-pi/2	Θ_6
i=7	0	0	pi/2	Θ_7 - pi/2

$$\mathbf{A_i} = \begin{pmatrix} cos(\Theta) & -sin(\Theta)cos(\alpha) & sin(\Theta)sin(\alpha) & cos(\Theta) * a \\ sin(\Theta) & cos(\Theta)cos(\alpha) & -cos(\Theta)sin(\alpha) & sin(\Theta) * a \\ 0 & sin(\alpha) & cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.13}$$

3.8 Software Libraries and Platforms

The conventional programming language used in implementing the experiments in this research work was C++. However, it was interfaced with several software libraries in-order to enhance application re-usability. In this section we will discuss the software platforms used in this work, but with a focus on how they are relevant to the implementation of our experiments.

3.8.1 Open Source Computer Vision (OpenCV) Library

OpenCV as the name implies is an open source computer vision library. It is free for both academic and commercial purpose, and is basically targeted towards real-time applications. The application of OpenCV in this research work was limited and mostly used in the 2D experiments. OpenCV has several re-usable libraries for computer visions algorithms: Gaussian blur, median blur, and bilateral filter for smoothing operations; Sobel derivatives, Scharr derivatives, and Canny edge detector for gradients and edge detection. However, we will discuss only those that were used in our experiments and as related to the research work.

(i) Smoothing: We used Gaussian blur for the removal of noise from the active window area of images in the ULBP [1] method. This enhances the performance of the ULBP method. However when used in the HOG [2] method it gave a reduced performance. The Gaussian filtering is done by multiplying the values in the local neighbourhood of each pixel with a Gaussian kernel centred on the pixel, and then summing the values to produce the output.

(ii) Gradients Computations: In the implementation of the HOG [2], we investigated three derivatives filters: (i) $1D$ $[-1, 0, 1]$; (ii) Sobel operator; and (iii) Scharr operator for the computation of an approximation of gradient of image intensity. We investigated a small kernel size of 3 for the Sobel and Scharr operators for computational efficiency. However, in our investigation the Scharr operator performed better than the other two operators. For this reason, we used the Scharr operator for the gradient computation in all HOG implementations in this thesis. The Scharr operator computes gradients of an image with 3 x 3 kernel filter values of $[-3, 0, 3]$, $[-10, 0, 10]$, $[-3, 0, 3]$ for the gradients in x direction and $[-3, -10, -3]$, $[0, 0, 0]$, $[3, 10, 3]$ for gradients in y direction by convolving the pixels with the filter.

(iii) Edge Detection: As will be discussed later in Chapter 4, a Canny Edge Detector was only used in the 2D image experiments so as to successfully adopt the method used by Mirolli et al. [23] for the image dataset used in our experiment. It has no bearing

on our method and the input values used in the training. A Canny Edge Detector just like the Sobel and Scharr operator uses a mask (filter) to calculate gradients along the x and y directions of an image. In addition: (i) it finds the gradient intensity and the direction rounded up for one of four possible angles $(0°, 45°, 90°, 135°)$; (ii) it uses non-maximum suppression to remove pixels that are not considered edges; and (iii) it sets two thresholds high and low, with gradient of pixels higher than the high threshold accepted as edges, and those lower than the low threshold rejected, and those in between are accepted only if they are connected to pixels above the high threshold.

3.8.2 Open Graphics Library (OpenGL)

The OpenGL is a cross-language and cross-platform Application Programming Interface (API) for rendering 2D and 3D vector graphics. It is typically used to interact with a graphic processing units (GPU) in order to achieve hardware-accelerated rendering. The OpenGL platform was used in the design of the 3D environments implementation in this PhD project. We used OpenGL because it is well documented, the availability of tuturials to support its user base, and it is easily portable to different platforms. However, because of the particular nature of our experiments and the evolutionary method, some features of the OpenGL played very significant roles in the project.

1. Display Lists: Display lists may improve performance since they can be used to store OpenGL commands for later execution. This characteristic may significantly enhance performance in graphics models especially when the commands are used to redraw the same geometry multiple times. By using display lists it is possible to define the geometry and/or state changes once and execute them multiple times. Some OpenGL operations such as **glRotate** involve heavy trigonometric computations, and these may result in onerous computational overheads for the system when executed many times. However, when such commands are cached in display lists this cost could be reduced. We therefore took advantage of the display list feature in optimising parts of our code that involve high graphics, and at the same time were redrawn in each time step of every trial of an evolutionary run. This improved system performance, especially in the environment classification experiments over the course of which we used texture mapping to a spherical shape in each time step to simulate our environment.

2. Texture and texture mapping: This feature of OpenGL was mainly used in the indoor and outdoor environment classification experiments. We used texture images to simulate our 3D indoor and outdoor environments. Texture mapping is the mapping of textures to one or more faces of a 3D model. In the environment

classification experiment (presented in Chapter 6), the 3D model was a sphere and the texture images for the indoor and outdoor environments were dynamically mapped to it at run-time. To make texture mapping operational, first the texture image has to be loaded into the OpenGL environment, texture coordinates have to be supplied with the vertices to map the texture, and sampling operations need to be performed from the texture using the coordinates where, for example, the texture coordinates are UV maps which are generated and the UV coordinates are scaled between 0 and 1 in order to derive the pixel colour (Fig. 3.6).

The basic implementation could be summarised with the following steps: (i) we initialised all the texture images in memory; (ii) we used the **glEnable** function to enable our texture type which is `GL_TEXTURE_2D`; (iii) we used the OpenGL function **glGenTextures**(*index*, `m_texture_objects`) to generate a number of texture objects and used their handles in a **GLUint** array which was a pointer to the texture images that were in memory; (iv) the texture objects were bound to the texture targets which in this case were 2D images using the **glBindTexture**(`GL_TEXTURE_2D`, `m_texture_objects`) function; (v) we used the **glTexParameteri** function to specify the filtering type to be used for magnification and minification and the texture target, while linear filtering type was used for precision; (vi) the **glTexImage2D** function was used to load the texture data itself; several parameters were specified in this function such as texture target, width and height of the texture, and the internal format in which OpenGL stored the texture.

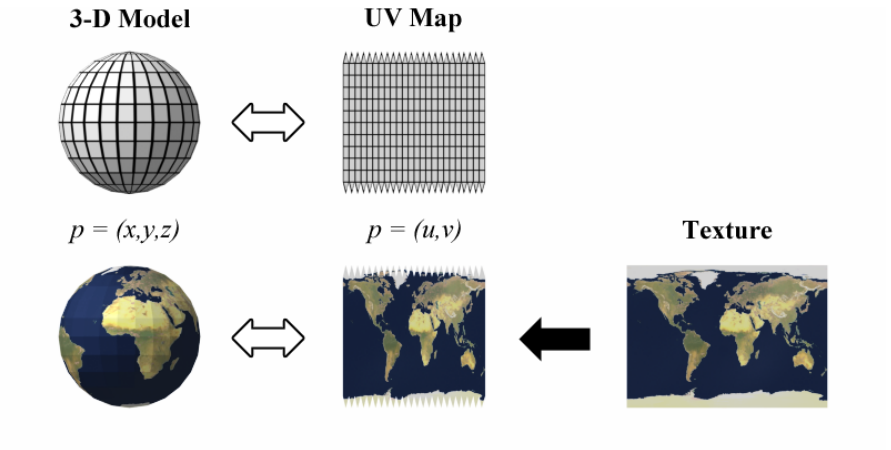

FIGURE 3.6: Texture mapping application in the UV space and as effected on a 3D model (image from [5])

3.8.3 OSMesa

This is an off-screen API for rendering into user allocated memory without any sort of window system or operating system dependencies. Its is used for rendering interactive 3D graphics. In our implementation we used the OSMesa platform for off-screen rendering. This off-screen rendering is very important in that our program needed to be parallelised and run on High Performance Computing machines (HPC Wales). In this way, during the training mode we would be able to avoid the additional computational overhead involved in the graphics.

3.8.4 Message Passing Interface (MPI)

MPI is a message passing API, which is designed for programming parallel computers. It has the goal of performance, scalability, and portability. The MPI API is able to interface with programming languages such as C, C++ and Fortran.

Implementation of the process of artificial evolution which often involves repetitive evaluations over several thousands of generations may incur prohibitively expensive computational overheads in a sequential mode. Thus, the evolutionary process using a genetic algorithm was parallelised with the MPI. Each individual ran its evaluation as a separate process and the respective fitness was communicated to the root process which in turn carried out the evolution and subsequently generation of a new set of controllers. The details of the MPI process are as follows: firstly, we initialised the MPI environment with the function `MPI_Init` and the type of the communication that was used with `MPI_Comm_rank`. The `MPI_COMM_WORLD` communicator was used to communicate among the processes and each process was assigned a rank. In the evolutionary method each genotype encoding controller parameters was dedicated as a process and assigned a rank. The root rank was used to send all the genotypes as processes to the other ranks with the `MPI_Send` function; the size of the ranks was equal to the number of genotypes. The remaining ranks received the genotypes for processing using `MPI_Recv`. The root rank went through the entire evolutionary process that is up to the breeding stage and other ranks had to wait for the root rank at the beginning of the evaluation process using the `MPI_Barrier` function. The root rank also used the `MPI_Gather` function to receive all the evaluated fitness from the ranks and performed the breeding of new genotypes. Finally, `MPI_Finalize` was called to terminate the MPI environment at the end of the execution of the program.

3.9 Chapter Summary

We started this chapter by outlining the requirements for the design of our proposed gaze control framework. We also described the computational methods that were used in the modelling of the framework. Subsequently, we discussed the software libraries and platforms used in the implementation of the computational models. In the subsequent chapters (4, 5, 6), we describe the various experiments in which our gaze control framework was instantiated.

Chapter 4

Experiment 1: Gaze Control in 2D Object Categorisation

4.1 Introduction

In Chapter 3, we discussed our gaze control framework and the requirements for the design. It was also mentioned that the gaze control framework was inspired by Mirolli et al. [23]. However, we have extended their control architecture with a gaze control that uses pre-processing techniques from computer vision. We also discussed software libraries used in our experiments. This chapter documents 2D object categorisation experiments using pre-processing techniques to improve object categorisation capability. This was done with the conjecture that human vision performs some kind of low-level processing in the mammalian visual cortex [109][211], which gives better visual representation for recognition. In addition, the pre-processing techniques we investigated have proved to be useful in a number of computer vision tasks, such as object detection [50][51], and provide robustness to variations in brightness, illumination, etc. Thus, we demonstrate this by first repeating the experiment as performed in [23] for italic letter categorisation to show that our system can correctly replicate the performance of their system. Secondly, we extended their architecture with pre-processing methods for more complex images taken from the camera of a humanoid robot, and we did so in order to show the effectiveness of pre-processing for active vision in categorisation tasks. We did not use pre-processing on the letter images as we were interested in more realistic images. In Section 4.2, we describe the experimental set up, while in Section 4.3, the results are presented. Section 4.4 gives a general discussion of the results of our experiments. Finally, in Section 4.5 we give a summary of the chapter.

4.2 Experimental Set-Up

We begin this section by introducing the neural network controller architecture and the optimisation technique used in our experiment. This was inspired by Miroli et al. [23]; however, it also incorporated the update equations of Tuci [29]. As mentioned in Chapter 3, our gaze control framework is based on the periphery-only architecture of Mirolli et al [23], and as such, it is used in our experiments which test categorisation of 5 italic letters ('l','u','n','o','j') and of the iCub images categorisation in this chapter. Table 4.1 lists the main terms and their meanings as used throughout the chapter.

TABLE 4.1: List the main terms and their meanings as used in this chapter.

Terms	**Meanings**
Genotype	A set of real values representing the parameters of a neural work controller.
Agent	A robot simulated as a neural network controller.
Generation	A time length in which a new population of genotypes is generated and subsequently evaluated for performance.
Evolutionary run	Such a run consists of many generations and is instantiated with a new seed to randomly generate a new population of genotypes for the first generation.
Best evolved genotype	Genotype that produced the best solution in each generation of an evolutionary run.
Trial	A time length in which an image is presented to an agent with a random initial eye position.
Time step	A single time frame in which the sensory patterns of the retina are input into the neural network.
Re-evaluate	Evaluation of genotypes derived from an evolutionary run for performance with a set of images different from the training set, and with a different initial eye position in each trial.
Best evolutionary run	The replication of an evolutionary run in which one of its re-evaluated best evolved genotypes produced the best performance among all evolutionary runs.
Best Fitness	Fitness of the best evolved genotype from each generation of an evolutionary run.

The neural network controller

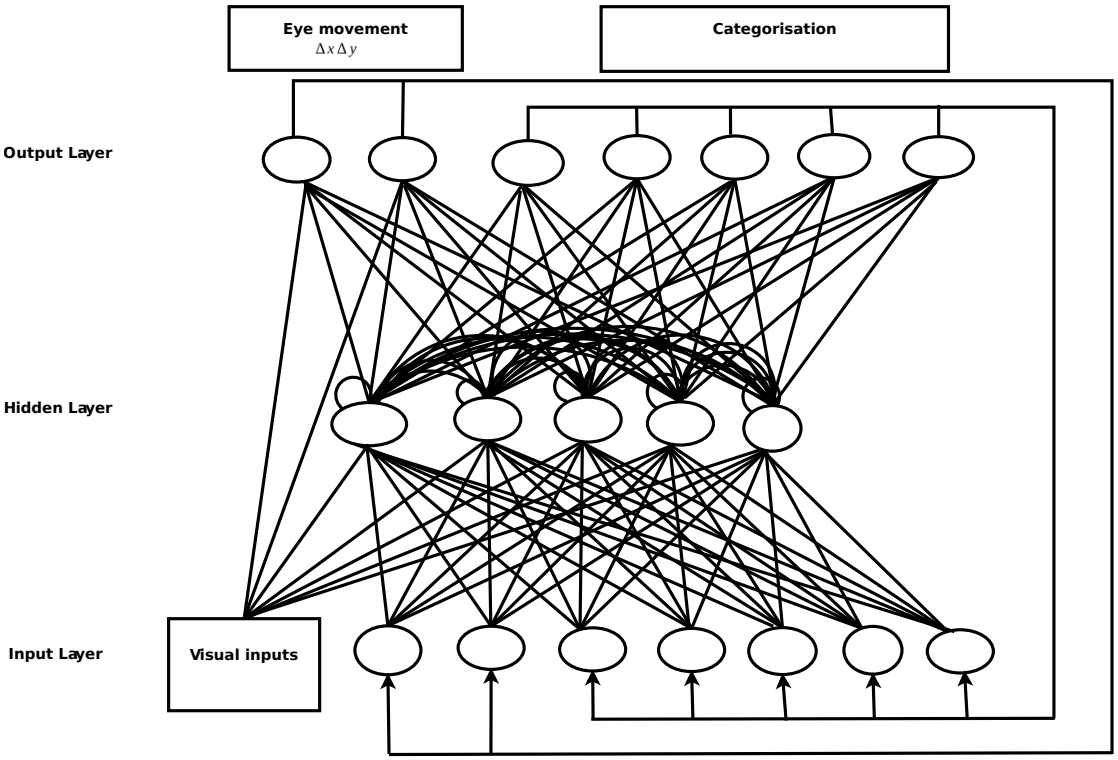

FIGURE 4.1: The architecture of our adopted periphery only Continuous Recurrent Neural Network, with recurrent hidden neurons.

The neural network controller is a continuous time recurrent three-layer architecture, with the updates equation as described in Chapter 3 (Fig. 4.1). The visual input vector size was determined by the method chosen for the pre-processing of the active window. It also has one hidden layer of 5 recurrent neurons, and an output layer of 7 neurons. In the output layer, 2 of the neurons determine the movement of the eye (size 50 x 50 pixels) per time step (maximal displacement of $[-12, 12]$ pixels in X and Y directions), and the other 5 neurons are for labelling the 5 categories. The input layer consists of units which encode the current activation state of the neurons for the visual stimuli of the active window, the copies of the 2 motor neurons, and the 5 categorisation units at the previous time step $t - 1$.

The activations of the input neurons were normalised between 0 and 1, however with 0 representing a fully white visual field, while 1 represents fully black for the grey-scale (as it was done in Mirolli et al. [23]). A random value with a uniform distribution within the range of [-0.05,0.05] was added to the activation values of the grey-scale method, Active-ULBP and Active-HOG at each time step, in order to take into account that sensor data are subject to noise. Note: we have adopted the parameters such as eye (active window size), maximal displacement per time step (i.e [12,-12] pixels) and the

number of hidden neurons (5) from the architecture of Mirolli et al. [23] in order to maintain consistency with their system.

The task and the evolutionary process

In this section, we explain the evolutionary process described in Chapter 3, as it pertains to the two major experiments described in this chapter, i.e. the grey-scale letter and iCub images categorisation.

In each trial, the eye was left to freely explore the image, however, a trial was terminated when the eye could no longer perceive any part of the object in the image for three consecutive time steps. The task of the agent was to correctly label the category of the current object during the second half of the trial, i.e., when the agent had explored the image for a sufficient length of time.

The initial population for each generation of the evolutionary process consisted of 100 randomly-generated genotypes sampled from a uniform distribution in the range $[0, 1]$, each encoding the free parameters of the corresponding neural controller, which includes all the connection weights, gain factors, biases, and the time constants of the hidden neurons. In order to generate the phenotypes, weights and biases were linearly mapped in the range $[-10, 10]$ and $[-5, 5]$ respectively, while the time constants were mapped in $[-1, 1.8]$. Generations following the first were produced by a combination of selection with elitism, recombination and mutation. For each new generation, the genotype with the highest fitness value ("the elite") from the previous generation was retained unchanged. The worst 30 were then removed. The remaining 99 genotypes of the new generation were formed by randomly selecting two genotypes from the older generation using roulette wheel selection, and a new genotype was created by combining the genetic material of these two old genotypes with a probability of 0.3 with cross-over point selected during the recombination. Mutation was done with the probability of 0.05, which entails that a random Gaussian offset was applied to each real-valued component encoded in the genotype. The mean was 0 and its standard deviation was 0.1. Note: the parameter values used as specified above both for the genotype/phenotype (neural network) mapping and the genetic algorithm were adopted from Tuci [29].

4.2.1 Letter Categorisation Experiment

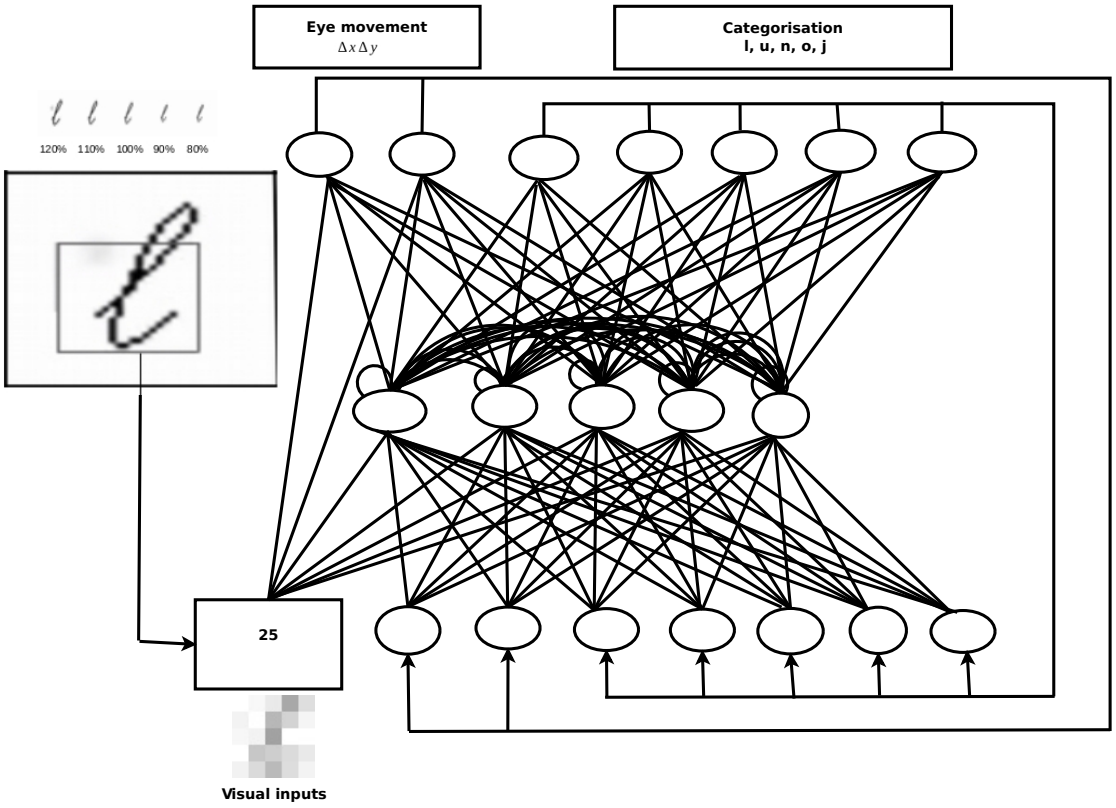

FIGURE 4.2: The architecture of the adopted periphery-only Continuous Recurrent Neural Network in the letter categorisation experiment

FIGURE 4.3: The above figure shows the five italic letter images

This experiment involved a moving eye located on an image of 100 x 100 pixels and was used to display the letters one at a time. The artificial eye consisted of 5 x 5 photo-receptors uniformly distributed over a square and which covered the entire retina. Each photo-receptor detected an average grey level of an area corresponding to 10 x 10 pixels of the image. The activation of each photo-receptor ranged from 0 to 1. The image was used to display five italic letters ('l','u','n','o','j') each of five different sizes, with a variation of ±10% and ±20% to the original size. Fig. 4.2 shows the letter 'l' displayed on the image and scanned by the moving eye, and Fig. 4.3 shows all the letters. The letters are displayed in black and grey over a white background as shown in Fig. 4.2 for the letter *l*.

At the beginning of each trial: (i) one of the 25 letter images of varied sizes was presented to each individual (the controller/genotype); (ii) the state of the internal neurons of the agent's neural controller was initialised to 0.0; and (iii) the eye was initialised in a random position within the central third of the image. The entire evolutionary run lasted for 3000 generations, with each individual/genotype evaluated for 50 trials (i.e. each image was presented twice to each individual), and each trial lasted 100 time steps (a presumably sufficient length of time for exploration in a trial). The results are presented in Section 4.3.

4.2.2 iCub-Images Categorisation Experiment

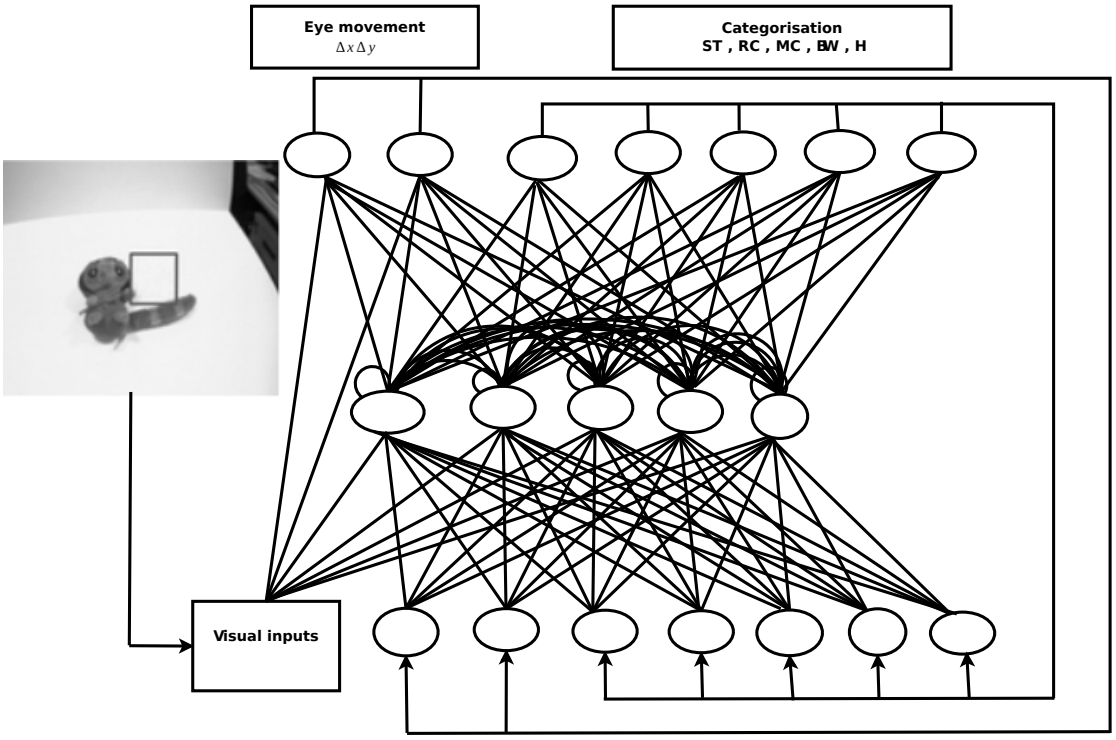

FIGURE 4.4: The Continuous Recurrent Neural Network architecture and the active window scanning the presented soft toy grey image in a trial for categorisation. In the categorisation units, the objects are represented as soft toy: **ST**, remote control: **RC**, microphone: **MC**, board wiper: **BW**, hammer: **H**

In this experiment we tested the ability of the proposed feature extraction methods (i.e Active-ULBP and Active-HOG) to enhance object categorisation tasks in an active vision system. This is shown, with a comparative experiment of the proposed methods with the grey-scale averaging method [23] for categorisation of objects in images taken from the camera of the iCub. The images are coloured, and of size 320 x 240 pixels of five different objects, namely: soft toy, TV remote control, microphone, board wiper, and hammer. The data-set consists of 350 images divided into two folds for training and validation. The first fold of 7 different sizes for each object varied between $[-20\%, 20\%]$

with respect to the original size; and each size of 5 different orientations varied between $[-4, 4]$ degrees with respect to the original orientation. The second fold also of 7 different sizes varied between $[-30\%, 30\%]$ of the original size; and each size of 5 different orientations varied between $-9, 9]$ degrees with respect to the original orientation. We used a larger range of scale and orientation in the second fold so as to make the categorisation task more challenging. The original coloured images were first converted into grey-images. Also, in order to make the images suited for the system, in which trials were terminated when the active window of (50 x 50 pixels) could no longer perceive any part of the object for three consecutive time steps, we used a Canny Edge Detector to detect the edges in each image presented. Subsequently, in each trial, a rectangular mask was set on the object in the image, and every white (edge) pixel outside the boundary of the rectangular mask were set to black. Through this means, we were able to get edge images that consisted of total outside boundaries of black, and objects of white and black. Fig. 4.5 shows the original coloured images, Fig. 4.6 shows the grey-images, Fig. 4.7 shows the images after being processed by the Canny Edge Detector and Fig. 4.8 shows the images after setting the rectangular masks on the Canny Edge Detector processed images. It should be noted that the above processing of the grey images by the Canny Edge Detector and rectangular masking, which finally led to the images shown in Fig. 4.8 were only used to terminate each trial after the active window lost total focus of the object for more than 3 consecutive time steps and as a result time was saved during training. The input vector into the neural work was obtained from the grey-images processed by the visual extraction methods, and the copies of the movement and categorisation units at previous time step $t-1$ as shown in Fig. 4.4.

At the beginning of each trial: (i) one of the 175 images (in a fold) was presented to each individual; (ii) the state of the internal neurons of the agent's controller was initialised to 0.0; and (iii) the eye was initialised in a random position within the central third of the image. The entire evolutionary process lasted for 3000 generations, with each individual/genotype evaluated for 350 trials (i.e each image was presented twice to each individual) and each trial lasted for 100 time steps. The results are presented in Section 4.3.

FIGURE 4.5: The original coloured images.

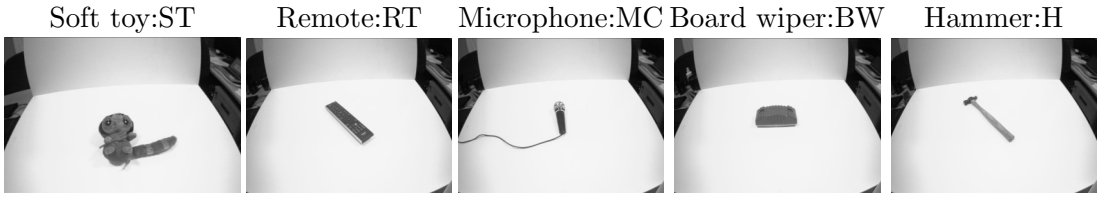

FIGURE 4.6: The converted grey-images.

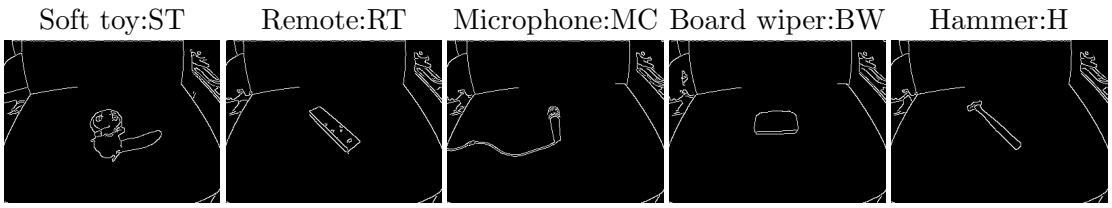

FIGURE 4.7: The images after being processed by the Canny Edge Detector.

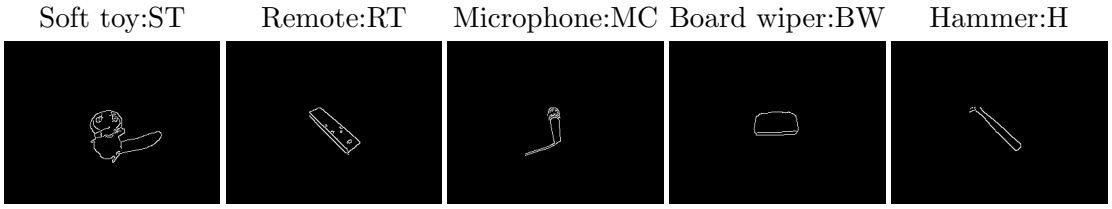

FIGURE 4.8: The images after setting rectangular masks on the Canny Edge Detector processed grey-images.

Grey-scale averaging

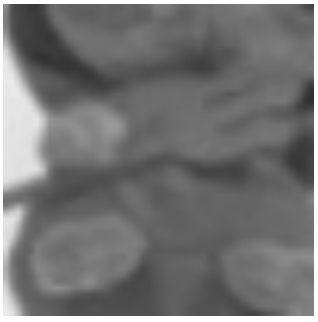

FIGURE 4.9: Original active window area of soft-toy grey-image

FIGURE 4.10: the active window area after grey-scale averaging

In this experiment, we adopted the grey-scale averaging method in Mirolli et al. [23], described in Chapter 3, for the processing of the active window. We have used the same number of parameters as used in their work, so as to be consistent with their system. The inputs were the average grey-levels of 10 x 10 pixels for each of the 25 inputs into the neural network. Fig. 4.9 shows the active window grey-image patch that was processed

in each iteration. Fig. 4.10 shows the average pixels of the active-window that were input into neural network.

Active-Uniform Local Binary Patterns

We instantiated the Active-ULBP described in Chapter 3 to process the active window, so as to take advantage of the uniform patterns that are present in texture images. Fig. 4.11 below shows the histograms and the concatenated histograms of the 4-cells of the active-window of a patch of the soft-toy image. The histograms were normalised and input to the neural network along with the output copies of motor units and classification units at the previous time step of each trial.

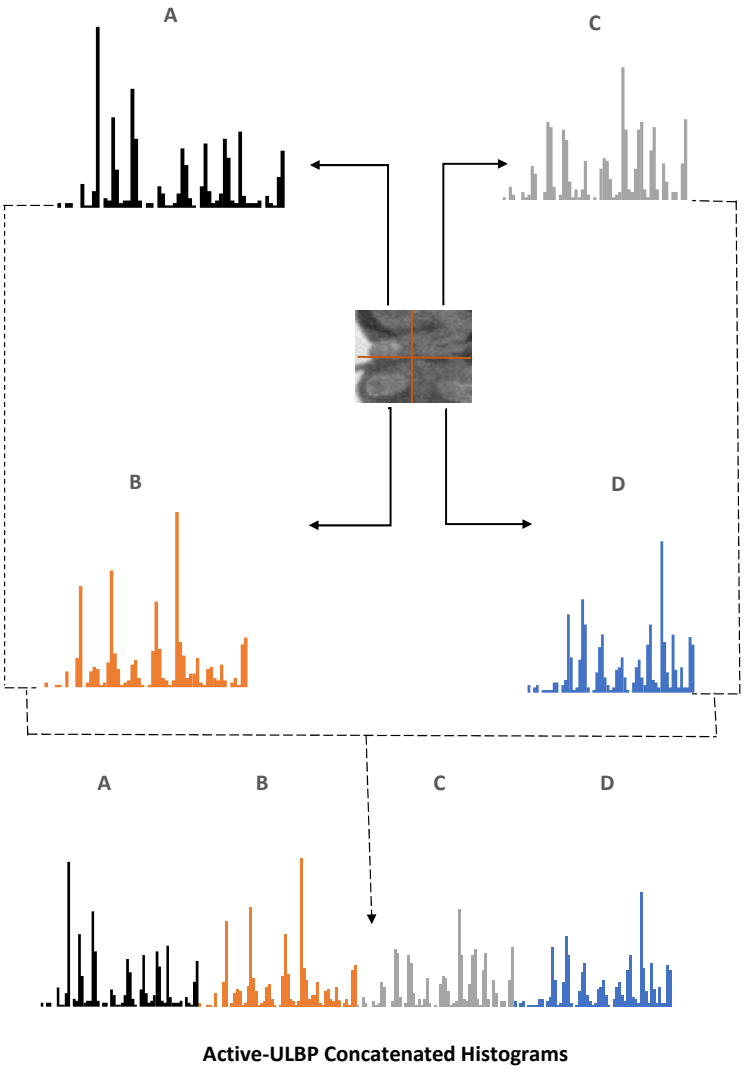

FIGURE 4.11: Active-ULBP histograms of the cells of the active window, and the concatenated histograms

Active-Histogram of Oriented Gradients

In order to further evaluate whether pre-processing can be used to improve the performance of active vision for categorisation tasks, we instantiated the Active-HOG described in Chapter 3. The visual inputs of the active vision in this case were normalised HOG features. As shown in Fig. 4.12, the concatenated histograms had a much smaller vector size of 36 as compared to that of Active-ULBP. The normalised concatenated histograms were input to the neural network controller along with the copies of categorisation and motor units in every time step of each trial.

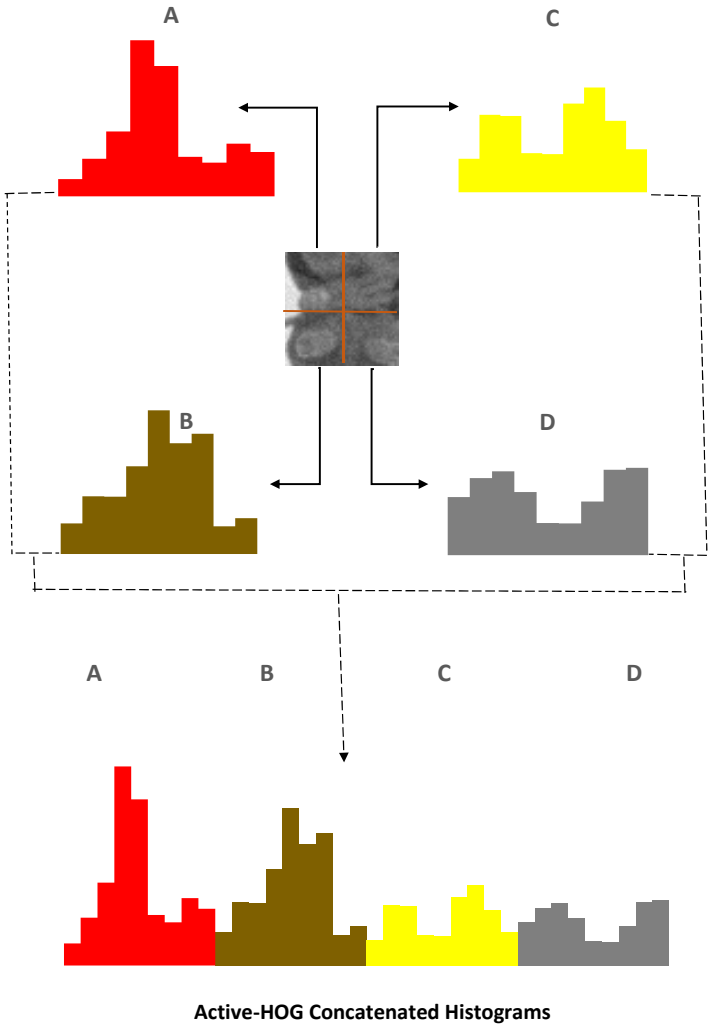

FIGURE 4.12: Active-HOG histograms of the active window image patch and the concatenated histograms

4.3 Results

In this section, we present the results and analysis for all the experiments. We first present the results of the replicated letter categorisation experiment which is only used to show that our system can reproduce the results of Mirolli et al. [23]. The results of the iCub image experiment are also presented, with a comparative analysis of the three methods of visual extractions, i.e. grey-scale averaging [23], Active-ULBP and Active-HOG. In order to assess the ability of the agent to correctly categorise the current objects in all of these experiments, we calculated the percentage of times, over the course of the second half of each trial, the categorisation unit corresponding to the current object (correct class) was the most activated.

4.3.1 Grey-Letters Categorisation

We performed 12 evolutionary runs, with each run lasting for 3000 generations (as shown in Appendix A, Fig. A.1). Each individual was evaluated for 50 trials and each trial lasted 100 time steps. Also, in order to assess the performance of the agent and its ability to generalise its skill, we performed another experiment in which we re-evaluated the best evolved genotypes from each of the evolutionary runs on the 50-sized set of letter images that were of different scales from those in the training. We present the results of the evolution, and the categorisation performance from the re-evaluation of the best evolved genotypes here.

4.3.1.1 Evolution

We discuss here the evolution of the letter-categorisation experiment. The best run from the 12 evolutionary runs (Fig. 4.13) started with sharp growth for about the first 400 generations and afterwards showed a more steadier growth. It finally peaked at a fitness close to the optimum value. However, considering the pattern of fitness for all evolutionary runs (Fig. 4.14), which shows the average (mean) of the best fitness in all generations of the 12 evolutionary runs and their positive and negative standard deviation. One can observe that at the beginning of the evolutionary runs (i.e in about the first 50 generations) the fitness of all the runs was very close to the mean, but after that point it largely deviated from the average for approximately another 2700 generations, and reduced in deviation in the remainder of the evolutionary runs. This suggest that on average all evolutionary runs improve towards their completion.

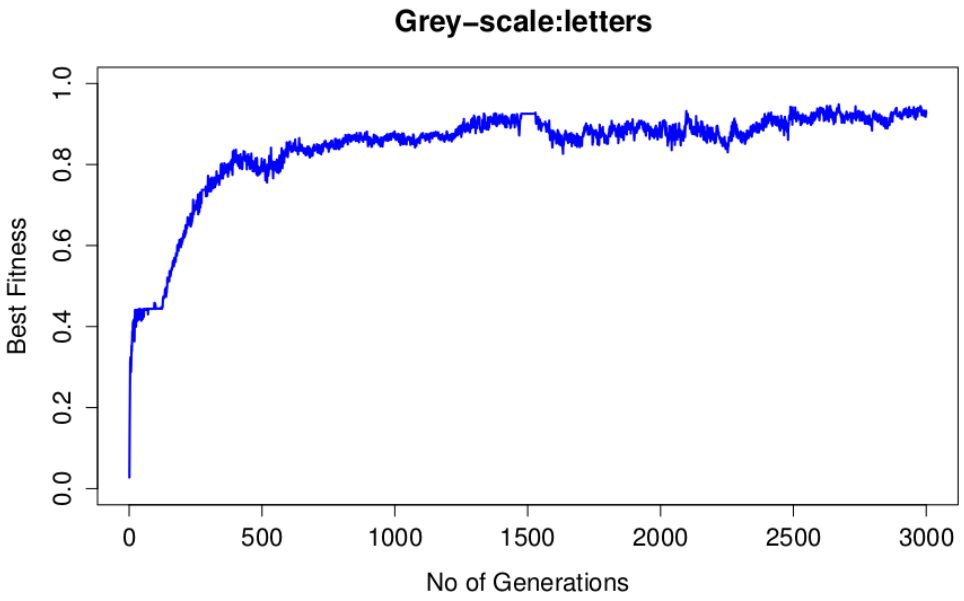

FIGURE 4.13: The best fitness graph of the best evolutionary run

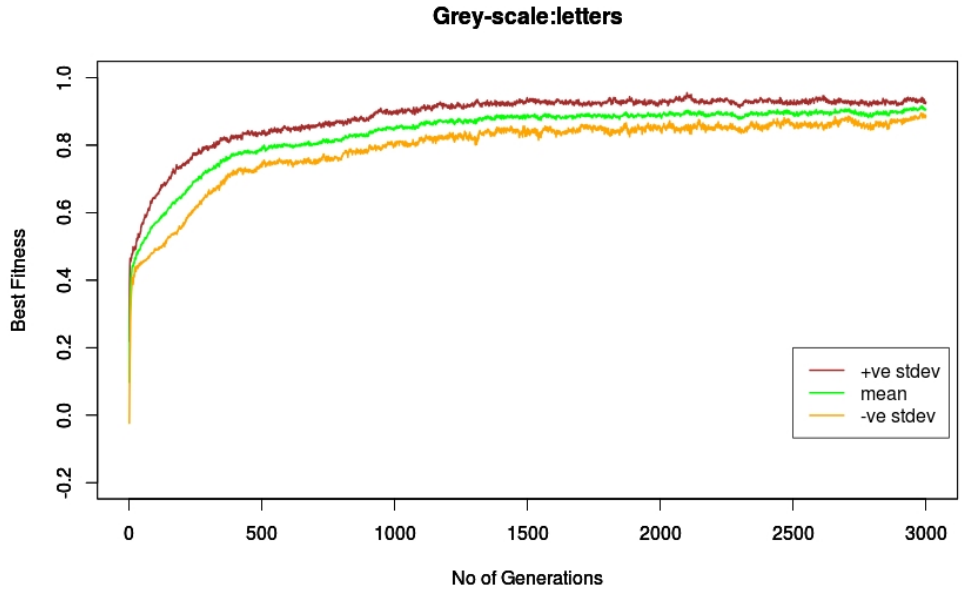

FIGURE 4.14: Shows the graph of the mean (average) of all best fitness in each generation of the 3000 generations for 12 evolutionary runs and their positive (+ve stdev) and negative (-ve stdev) standard deviation in each generation.

4.3.1.2 Categorisation Performance

In order to assess the best genotypes in all evolutionary runs for categorisation performance, we re-evaluated (for testing) the best genotypes of the last 1000 generations of the evolutionary runs for the categorisation task. This was simply because the last 1000 generations should have a relatively higher fitness pattern than the other generations

and as a result yield better solutions. The best genotypes were exposed to 50 images of the same italic letters used in the evolutionary runs (training) but of different scales. The letters were scaled within the range $[-15\%, 15\%]$ to the original size with a uniform variation. The re-evaluation was done for 10000 trials during which the eye was randomly located in 200 different initial starting positions on each image size.

The results are as shown in Table 4.2 and Table 4.3. The confusion matrix in Table 4.2 shows the average performance of the best performing re-evaluated genotype for all trials. Table 4.3 gives a statistical summary of categorisation performance. The metrics used in statistical summary of (Table 4.3) are as follows: **Max** represents the best performance from the best evolved genotypes re-evaluated in all runs; **Average** represents the average of the best performance in each run; **Worst** is the worst of the best performance in each run; and **stdev** is the standard deviation of the best performance of all runs. The best performance from all re-evaluated best evolved genotypes in all runs was 96.70%, while the average of the best performance from all runs was 92.76%, and the worst performance was 82.60%. The performance result from our experiment of 12 evolutionary runs was comparable to that of the original implementation by Mirolli et al. [23] of 20 evolutionary runs. Their best and average performance was 99.87% and 86.85% respectively, as compared to our best and average performance of 96.70% and 92.76% respectively. The difference in the results may be due to the following reasons:

(i) The difference between the Continuous Recurrent Neural Network (CTRNN) update equations used, as we have used the update equation in Tuci [29].

(ii) The difference in the number of replications of evolutionary run (12 versus 20).

(iii) The random elements involved, including the seed and selection process. It is likely that the differences in results are not statistically significant.

TABLE 4.2: The confusion matrix showing the average performance of the best performing re-evaluated genotype for all trials of letters

	Average Activation Rates (Highest in Bold)				
Current category	**l**	**u**	**n**	**o**	**j**
l	**89.81**	0.00	0.00	0.02	10.17
u	0.00	**99.27**	0.37	0.36	0.00
n	0.00	1.11	**97.30**	1.58	0.00
o	0.15	0.03	2.65	**97.12**	0.06
j	0.00	0.00	0.00	0.00	**100.00**

TABLE 4.3: Best, average and worst performance in all runs.

Max	Average	Worst	Stdev
96.70	92.76	82.60	±3.52

4.3.2 iCub-Images Categorisation

This section presents the results of the three methods of visual representation for active vision. As comparative analysis was conducted for the purpose of method comparison, all other conditions in the evolutionary process were constant; the only difference was the input vector size which was determined by the visual extraction method.

4.3.2.1 Evolution

During the evolutionary stage, we performed 20 evolutionary runs for each of the visual extraction methods for the 2-fold cross validation (as shown in Appendix A, Fig. A.2, Fig. A.3 and Fig. A.4). Each evolutionary run had 3000 generations, with each genotype evaluated for 350 trials, and each trial consisting of 100 time steps. The first 10 runs were for the first fold, while the last 10 runs were for the second fold of the 2-fold cross validation.

Observation of the best fitness graphs of the best runs of the three methods (Fig. 4.15) reveals that the Active-ULBP had a fitness pattern that was higher overall than the other two methods, while the grey-scale was slightly higher than the Active-HOG. Active-HOG also seems to have peaked earlier than the other methods. However, in Fig. 4.16, that shows the average (mean) of the best fitness in all generations of all evolutionary runs and their positive and negative standard deviations for the three methods of visual extraction, one can observe that the general average (mean) fitness pattern was higher for the Active-ULBP than for the other two methods in most generations. Also, the mean pattern for grey-scale was slightly higher than that of Active-HOG. Observing standard deviation from the mean for the three methods, one can observe that all three methods produced a best fitness that was very close to the mean in the first few generations; however larger deviations are observed in the remaining generations. Moreover, Active-HOG exhibits a larger deviation from the mean at an earlier stage than the other two methods.

Overall, the fitness patterns of all runs seems to be closer to the mean for the Active-ULBP than for the other two methods, especially from approximately 700 generations onwards. By contrast, the fitness patterns for the grey-scale and Active-HOG methods were very similar. This suggests that the fitness patterns for all runs of the Active-ULBP in general seem to progressively improve in all generations as compared to the other two visual extraction methods.

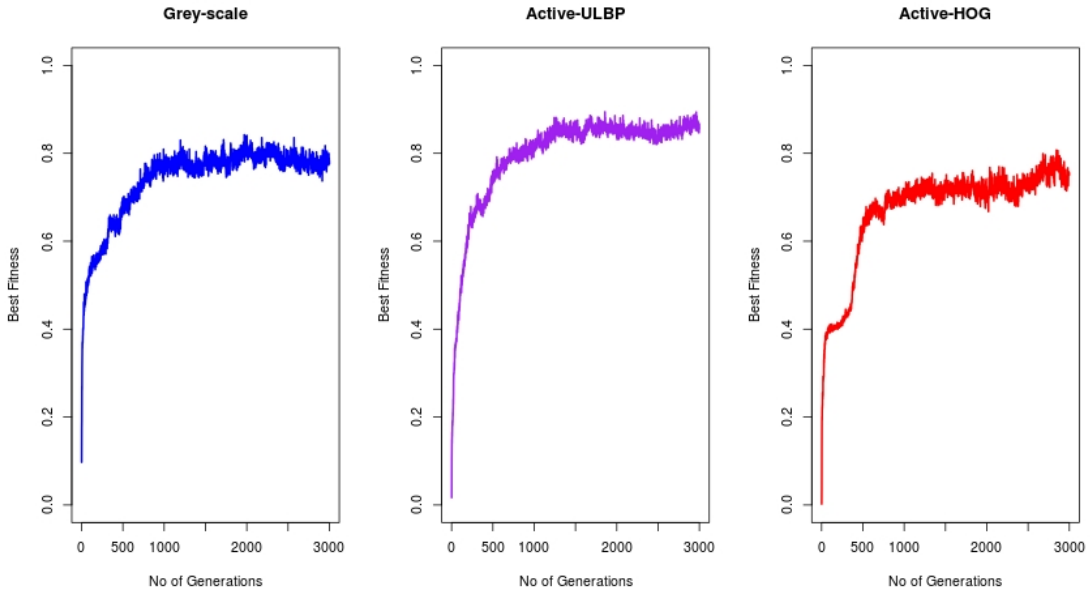

FIGURE 4.15: The best fitness graphs of the best evolutionary runs of the three visual extraction methods in the 2-fold cross-validation.

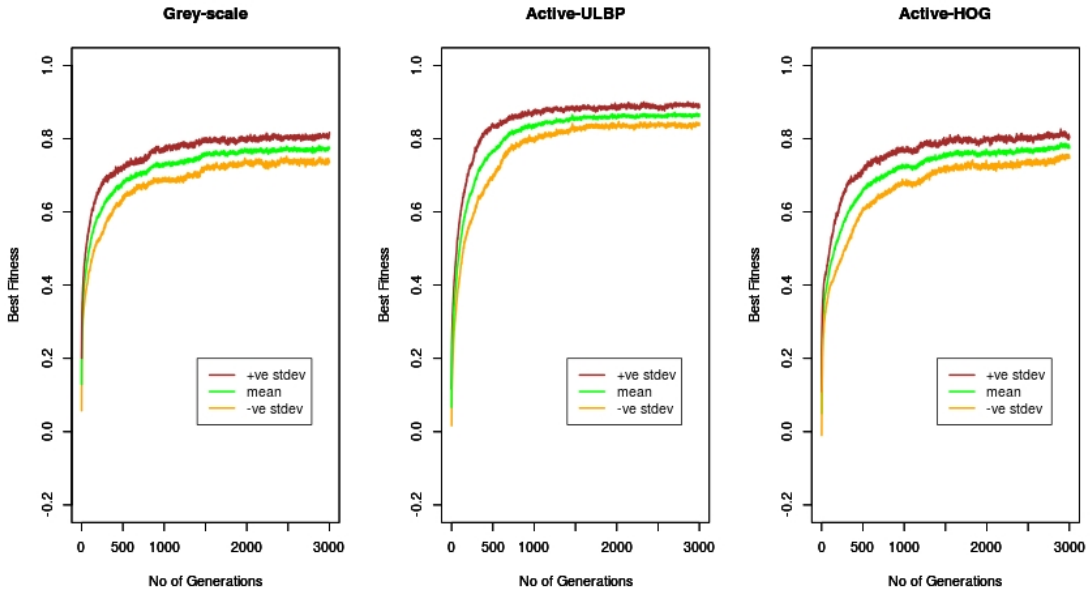

FIGURE 4.16: Shows the graph of the mean (average) of all best fitness in each generation of the 3000 generations for 20 evolutionary runs and their positive (+ve stdev) and negative (-ve stdev) standard deviation in each generation for the three methods of visual extraction

4.3.2.2 Categorisation Performance

In order to assess the performance of the system, we re-evaluated the best genotypes of the last 1000 generations of the evolutionary runs for the categorisation task for each

fold of the 2-fold cross validation for the three methods of visual extraction. A total of 700 trials were done, with each image of each fold (175 images) presented 4 times to the agent with a random initial eye position in each trial. The categorisation performance was based on the percentage of times in which the categorisation unit corresponding to the current (correct) category was the most activated in all trials. The confusion matrices of the best performing re-evaluated evolved genotypes for the three methods of visual extraction show that the current category had the highest percentage of correct categorisation in all trials (Table 4.4, Table 4.5, Table 4.6). However, that of the Active-ULBP and the grey-scale methods were slightly better than that of the Active-HOG.

TABLE 4.4: The average performance of the best performing re-evaluated genotype of **grey-scale averaging** in all trials of the iCub-images.

	Average Activation Rates (Highest in Bold)				
Current category	**soft toy**	**remote control**	**microphone**	**board wiper**	**hammer**
soft toy	**99.60**	0.32	0.00	0.08	0.00
remote control	0.00	**99.58**	0.00	0.00	0.42
microphone	0.00	0.00	**100.00**	0.00	0.00
board wiper	0.00	0.00	0.00	**100.00**	0.00
hammer	0.00	0.31	0.60	0.04	**99.06**

TABLE 4.5: The average performance of the best performing re-evaluated genotype of **Active-ULBP** in all trials of the iCub-images.

	Average Activation Rates (Highest in Bold)				
Current category	**soft toy**	**remote control**	**microphone**	**board wiper**	**hammer**
soft toy	**99.95**	0.00	0.00	0.05	0.00
remote control	0.00	**98.93**	0.00	0.00	1.07
microphone	0.00	0.04	**99.96**	0.00	0.00
board wiper	0.02	0.00	0.00	**99.98**	0.00
hammer	0.00	0.00	0.00	0.00	**100.00**

TABLE 4.6: The average performance of the best performing re-evaluated genotype of **Active-HOG** in all trials of the iCub-images.

	Average Activation Rates (Highest in Bold)				
Current category	**soft toy**	**remote control**	**microphone**	**board wiper**	**hammer**
soft toy	**95.98**	0.00	0.00	4.02	0.00
remote control	0.00	**100.00**	0.00	0.00	0.00
microphone	1.17	0.33	**97.07**	1.28	0.16
board wiper	1.62	0.00	0.00	**98.38**	0.00
hammer	0.00	0.00	0.62	0.00	**99.38**

The overall performance for the three methods of visual extraction can be estimated by observing the summary of statistics of categorisation performance in (Table 4.7). The metrics used are: **Max** is the best performance from all re-evaluated best evolved genotypes of all runs; **Average** represents the average of the best performance in each run; **Worst** is the worst of the best performance in each run; and **stdev** is the standard

TABLE 4.7: The summary of performance statistics of the three visual extraction methods in the 2-fold cross-validation (i.e. 20 evolutionary runs).

Visual extraction methods	Max	Average	Worst	Stdev
Grey-scale averaging	99.65	95.77	87.26	±4.13
Active-ULBP	99.77	96.82	91.75	±2.49
Active-HOG	98.16	92.87	77.81	±5.26

deviation of the best performance of all runs. From the table one can see that the Active-ULBP best performance of 99.77% was slightly better than that of grey-scale (99.65%) and Active-HOG (98.16%). Also, Active-ULBP exhibited the highest average performance of 96.82% as compared to that of grey-scale (95.77%) and Active-HOG (92.87%). Active-ULBP, also showed the best worst performance of 91.75% as compared to that of grey-scale (87.26%) and Active-HOG (77.81%). Furthermore, the standard deviation also shows that the best performance of Active-ULBP in all runs were less sparsely distributed than those of the grey-scale and Active-HOG, while those of grey-scale were slightly less sparsely distributed than those of Active-HOG. Fig. 4.17 also shows the average of the best performance of all runs for each method of visual extraction.

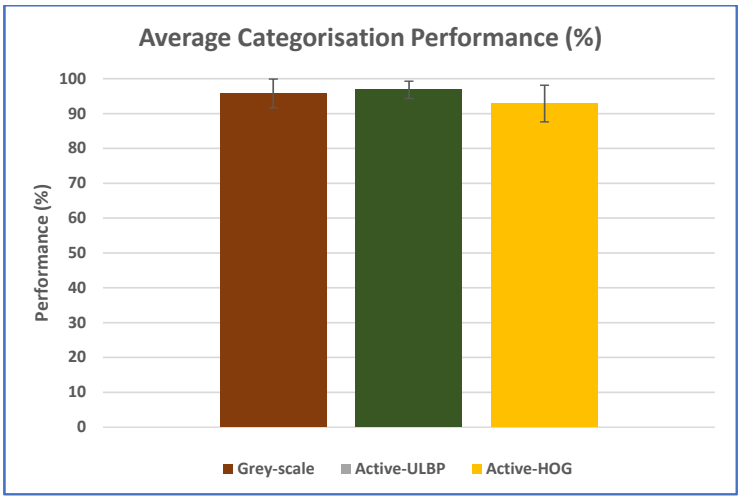

FIGURE 4.17: The bar-charts above shows the average categorisation performance and standard deviations of the three methods of visual extraction in all runs.

Statistical Analysis

We further tested if the averages of the three visual extraction methods were significantly different with the t-test. However, the commonly used t-test is mainly used for comparison between two means. Since we made our comparison among three methods, we used the extension of the t-test that exams if there is a significant difference among three or more means (averages). This test is also known as the one-way analysis of

variance (ANOVA). We tested the significance of the differences of the averages (means) with a (p-value<0.05) and a more stringent (p-value<0.01). Table 4.8 shows a statistical summary of the three visual extraction methods that was used to calculate the values of the results of the ANOVA test. The tables metrics are as follows: the **Visual extraction methods** column indicate the type of pre-processing techniques; the **Count** gives the number of evolutionary runs; the **Sum** is the sum of the individual performances of the best performing re-evaluated genotypes from all runs of the three methods of visual extraction; and the **Average** and **Variance** columns indicate respectively the averages and variance of the performance of the best performing re-evaluated genotypes from all runs of the three methods. Table 4.9 statistically summarises the results of the ANOVA test, and its metrics are as follows: the first column represents the **Source of variations** (between and within the groups) for which averages were compared (i.e. grey-scale averaging, Active-ULBP, Active-HOG); **SS** represents the sum of squares; **df** is the degree of freedom; **MS** represents the means squares; **F** refers to the F distribution value; **P-value** is the significance level of the averages that were considered; and **F crit** denotes the F critical value.

TABLE 4.8: Summary of the statistics of the best performing re-evaluated genotypes of the three visual extraction methods from 20 evolutionary runs that was used in the anova test.

SUMMARY				
Visual Extraction Methods	***Count***	***Sum***	***Average***	***Variance***
Grey-scale averaging	20	1915.33	95.77	17.04
Active-ULBP	20	1936.30	96.82	6.22
Active-HOG	20	1857.34	92.87	27.65

TABLE 4.9: The results of the anova test.

ANOVA						
Source of Variation	***SS***	***df***	***MS***	***F***	***P-value***	***F crit***
Between Groups	167.29	2	83.65	4.93	0.0106	3.16
Within Groups	967.33	57	16.97			
Total	1134.62	59				

TABLE 4.10: The significant test results using a paired t-test with test conditions of (p-value<0.05) and (p-value<0.01).

			Signf. Level=0.05	***Signf. Level=0.01***
Compared Groups	***t-value***	***p-value***	***Bonf. Corr=0.0167***	***Bonf. Corr=0.003***
Active-ULBP and Grey-scale	0.81	0.2862	Not Significant	Not Significant
Active-ULBP and Active-HOG	3.03	0.0052	Significant	Not Significant
Grey-scale and Active-HOG	2.23	0.0354	Not Significant	Not Significant

The obtained p-value of 0.0106 as shown in the Table 4.9 was less than our first significance level of 0.05, this represents a strong evidence against the null-hypothesis that

the averages for the three visual extraction were equal. Therefore, we reject the null hypothesis. On the other hand, the obtained value was slightly higher than the second significance level of 0.01, or what we might consider to be "highly significant". We then investigated which method pairs were significantly different by using pairwise t-tests and applying a Bonferroni correction for the two significance levels of 0.05 and 0.01. The purpose of the ANOVA test is to indicate if there is a need to proceed with pairwise significance tests, or if there is insufficient evidence to have confidence that any differences were not due to chance. Nevertheless, we still computed the Bonferroni correction for the second significance level of 0.01, since the obtained p-value of 0.0106 was only slightly higher than 0.01 and. as such, could be referred to as showing a trend towards being "highly significant". As shown in Table 4.10, the Bonferroni corrected value for significance level of 0.05 is 0.0167, while that of 0.01 is 0.003.

The result of the paired t-test of the averages of the three groups (visual extraction methods) using Bonferroni correction is shown in Table 4.10. The first column indicates the paired groups that were compared, the second and third columns indicate the t-values and the p-values of the averages compared for each paired group, while the fourth and fifth columns indicate the level of significance of each paired group averages using the Bonferroni corrected values. Comparing the three groups in the table at the 0.05 significance level with Bonferroni correction of 0.0167, the variation in averages between Active-ULBP and grey-scale was not significant, i.e. the difference could have arisen by chance and, while that of Active-ULBP and Active-HOG was significant and that of grey-scale and Active-HOG was not also significant. Therefore, for the significance level of 0.05, we fail to reject the null-hypothesis that the averages of the two groups Active-ULBP and grey-scale, and grey-scale and Active-HOG were equal, but we reject the null hypothesis for the case of Active-ULBP and Active-HOG.

On the other-hand, for the significance level of 0.01, the averages of all three groups compared (i.e. Active-ULBP and grey-scale, Active-ULBP and Active-HOG, grey-scale and Active-HOG) were not significantly different. We therefore fail to reject the null hypothesis that the averages were equal for these three groups.

4.3.2.3 Dynamics of the Categorisation Process

This section deals with the dynamics of the categorisation process in the case of iCub images experiment. In particular, we investigate:

(i) To what extent the sensory stimuli provided by one of the visual extraction techniques, and experienced by the agent during interaction with the images, have been sufficient to provide the regularities required to facilitate the categorisation process.

(ii) To what extent the agent succeeded in self-selecting the stimuli that can be unambiguously associated with a particular category.

Note: stimulus ambiguity may depend on the nature of the stimulus, the field of view of the artificial eye and the eye location.

The categorisation answers given by our system were dependent on the visual information that was provided, apart from the copy of the outputs categorisation and motor units at the previous time step. However, since our focus is mainly on the influence of visual representation on control of the active vision in order to improve learning for categorisation, we only investigate here the visual sensory channel (i.e. we exclude the motor and the categorisation copies). In order to do this we focus our analysis on computing the Modified Version of the Geometric Separability Index (MGSI). The Geometric Separability Index (GSI) was originally proposed by Thorton [212], while the MGSI is a modified version of the GSI and was proposed by Mirolli et al. [23]. The GSI computes the percentage rate at which the nearest pattern of each experienced pattern belonged to the same category; however the MGSI is more demanding in that it takes into account not only the nearest neighbour but all the stimuli belonging to the same category. We chose to use this more demanding measure because the nature of our problem is very similar to that of Mirolli et al. [23]. The MGSI is defined by the equation below:

$$MGSI(P) = \frac{\sum\limits_{s \in P} \frac{\sum\limits_{n \in N_s} I_{C_s}(n)}{|C_s|}}{|P|}$$

Which is defined as the average over all patterns, of the proportions of patterns belonging to the same category, that are in the $|C_s|$ nearest patterns (computed from Euclidean distance), where $|C_s|$ represents the total number of patterns in the same category as pattern s. Where P is the set comprising all the patterns, $|P|$ is the cardinality of the set P, C_s is the set of all patterns belonging to the same category as pattern s (s does not belong to C_s), N_s is the set of the $|C_s|$ patterns nearest to pattern s, and $I_{C_s}(n)$ is the indicator function of set C_s, that returns 1 if n is in set C_s and 0 otherwise. We computed the MGSI of the best performing re-evaluated evolved genotypes for all three visual extraction methods for 1750 trials during which the agent experiences the five different categories (i.e. soft toy, remote control set, microphone, board wiper and hammer) of the 35 different samples for each category, 10 times each with different initial eye positions. For each type of visual extraction method of the sensory patterns the MGSI has been calculated for each of the 100 time steps of a trial.

Observing the change in the MGSI for the three methods of visual extraction (Fig. 4.18, Fig. 4.19, Fig. 4.20):

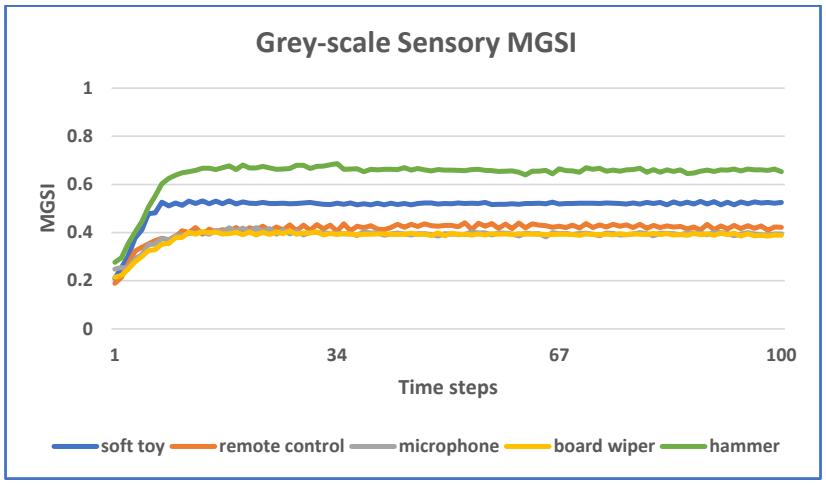

FIGURE 4.18: Modified Geometric Separability (MGSI) of the stimuli provided by grey-scale averaging.

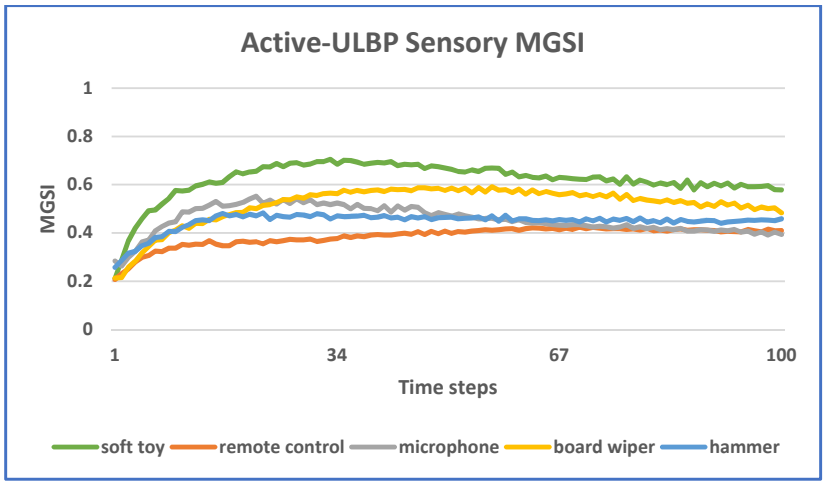

FIGURE 4.19: Modified Geometric Separability (MGSI) of the stimuli provided by the Active-ULBP method.

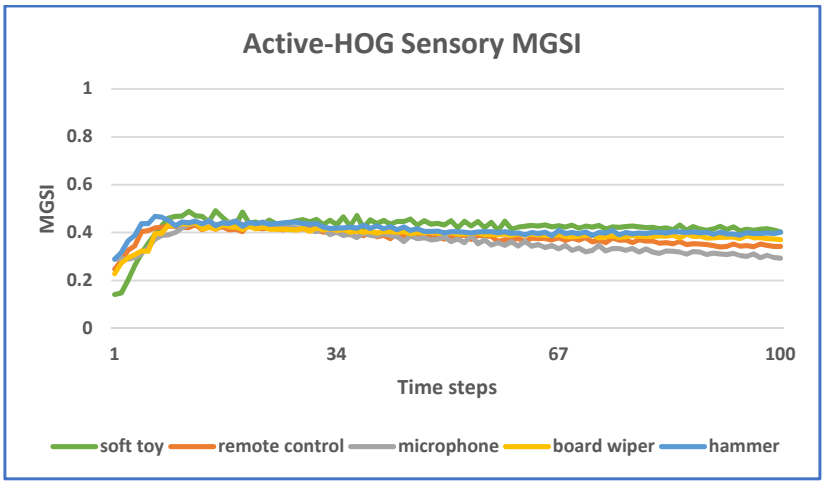

FIGURE 4.20: Modified Geometric Separability (MGSI) of the stimuli provided by the Active-HOG method.

.

(i) The fact that the MGSI increased for all conditions (visual extraction methods) and for all objects, shows that the system moved away from very ambiguous to less ambiguous stimuli.

(ii) The MGSI never reached a value of 1. This means that the system never managed to generate unambiguous stimuli for all the visual extraction methods. This was obviously not a problem given by the performance of the three visual extraction techniques.

(iii) The Active-ULBP method generated less ambiguous stimuli than the grey-scale and Active-HOG methods, however grey-scale was more consistent. This means that the stimuli generated by the system for Active-ULBP were generally more separated in sensory space than the other two, but with the grey-scale more consistent.

(iv) Active-HOG reached a peak that was almost equal the lowest levels of Active-ULBP and grey-scale and maintained the same approximate level over time. This means that in the case of the Active-HOG, the system did not exhibit as great a tendency to move towards less ambiguous stimuli when compared with the other two methods.

(v) For some objects, the system managed to generate less ambiguous patterns than for other objects. This means the system produced more discriminative patterns for those objects than for others.

4.4 Discussion

We started this chapter by trying to replicate our benchmark model (Mirolli et al. [23]) for letter categorisation. The average performance of our system was comparable and the difference may be due to the use of different update equations, the number of repeated evolutionary runs and the random elements involved. We then extended the benchmark system with pre-processing for images taken from the iCub camera. We discuss: (i) Visual representation and active vision categorisation; (ii) Learning control of the active vision system.

Visual Representation and Active Vision Categorisation

We investigated three methods of visual extraction, i.e. grey-scale averaging method [23], Uniform Local Binary Patterns [1] (as Active-ULBP) and Histogram of Oriented Gradients [2] (as Active-HOG). Consequently, we discuss here the impacts of the visual representation methods on categorisation performance. In our investigation, Active-ULBP demonstrated the best average performance, while grey-scale also outperformed

Active-HOG. However, further investigation based on statistical analysis using two significance levels of 0.05 and 0.01, showed that at the 0.05 level of significance, Active-ULBP's higher level of performance compared to grey-scale was not statistically significant, while its higher level of performance compared to Active-HOG was significant. The better performance of grey-scale in comparison to Active-HOG was also not statistically significant.

On the other-hand, at the 0.01 level of significance, none of the three visual extraction methods was found to be significantly better than the others when compared.

Therefore, we could deduce that based on a more comprehensive significance test (i.e. using significance levels of 0.05 and 0.01), higher average performance of Active-ULBP relative to Active-HOG was significant but not "highly significant".

On the whole, the very good performance of Active-ULBP lends further support to ULBP as an effective feature descriptor for texture information. The performance of Active-HOG also showed that it can be an effective feature representation for images characterised by some level of structural information. Overall, the good performance of the two pre-processing methods investigated has demonstrated their potential for good visual representation in active vision systems.

Learning Control of the Active Vision System

Intelligent cooperation between sensory and motor systems can help to facilitate a categorisation process. Not only does the motor system help to shape the visual stimuli experienced by an agent, but the type of stimuli experienced by the agent can also determine the corresponding motor responses that help to improve recognition capability.

The improvement of MGSI values over time for the three visual extraction methods investigated shows the impacts these kind of sensory patterns can have on corresponding motor actions which can in turn facilitate categorisation performance.

Moreover, the fact that Active-ULBP generated less ambiguous stimuli over time than the other two methods, showed that the sensory patterns provided by Active-ULBP have more productively generated motor behaviours that facilitate learning for categorisation. This may be due to the fact that ULBP has been shown to work well for texture images, and as a result must have given better sensory patterns that facilitate the learning process. The fact that grey-scale improvement was generally more consistent, also shows its ability to assist the system in experiencing more discriminative stimuli with more consistency than the other two methods. Active-HOG also helped the active vision

system in experiencing less ambiguous stimuli over time, though with not as much impact when compared to Active-ULBP and grey-scale.

Finally, the performance of the three visual extraction methods was close to optimum, in spite that their stimuli were not fully separated in the input space (i.e. the MGSI never reached a value of 1). This indicates that the system must have integrated sequences of experienced sensory states over time through internal dynamics of the neural network controller.

4.5 Chapter Summary

In this chapter, we began by trying to replicate the experiment of Mirolli et al. [23] for letter categorisation, which was used as the bench-mark for our active vision model. We found it necessary to do this to determine if our system can effectively reproduce the performance of their system. Our active vision system in the letter categorisation experiment had a best performance of 96.70% in 12 evolutionary runs, as compared to the best performance of 99.87% recorded by Mirolli et al. [23] in 20 evolutionary runs. Moreover, the average performance of our system was 92.76% as compared to the equivalent value 86.85% documented for their system. The difference in performance may be due not only to the use of different update equations, but also the different number of replications of evolutionary runs used in our system.

Subsequently, the bench-mark model was extended with pre-processing using Uniform Local Binary Patterns [1] (as Active-ULBP) and Histogram of Oriented Gradient [2] (as Active -HOG) on images taken from the iCub camera. The results achieved by pre-processing the visual stimuli showed that Active-ULBP had an average performance of 96.77% which compared favourably with Active-HOG (92.87%) and grey-scale (95.77%).

However, statistical analysis showed that the average performance of Active-ULBP was not significantly different from that of grey-scale and not "highly significantly" different from that of Active-HOG. There was also no significant difference between the average performance of grey-scale and Active-HOG.

The analysis result of the Modified Geometric Separability Index (MGSI) [23] showed that the performance of the active vision system using the three visual extraction methods was based on intelligent coordination between sensory and motor units, and should also involve integration of the perceptual information over time, since the stimuli provided by the three methods were never completely separated in the input space.

The next chapter extends the active vision system for object categorisation in the 3D environment using a simulated iCub robot platform.

Chapter 5

Experiment 2: Gaze Control in 3D Object Categorisation

5.1 Introduction

We investigated pre-processing techniques in the last chapter for improving the categorisation capability of an active vision system in a 2D environment. In particular, we extended the work of Mirolli et al. [23] with Histogram of Oriented Gradients [2] and Uniform Local Binary Patterns [1] for improved visual perception. However, in order to demonstrate how active vision is performed in the real world, we further explore the 3D motor-space. Consequently, we chose the humanoid iCub platform to investigate how biological agents use their vision system to perform object categorisation. Section 5.2 describes the experimental set-up, while Section 5.3 provides the results. Section 5.4 gives general discussion of the chapter and the results, and finally in Section 5.5, a summary of the chapter is given.

5.2 Experimental Set-Up

This experiment is designed to investigate how a simulated agent (the iCub) can exploit its eye movement to improve object categorisation. Furthermore, given the strong interdependencies between motor responses and sensory stimuli, we also investigated how this categorisation capability can be improved with pre-processing techniques. We trained the simulated robot controller through an evolutionary technique, in order to investigate how the agent exploits its eye movement to improve perception for object categorisation. The encoded free parameters of the evolutionary technique, that regulate how the agent

interacts with its environment and the agent's categorisation responses were randomly varied, and variations were retained or discarded on the basis of the agent's ability to perform the categorisation task.

5.2.1 The iCub agent and the environment

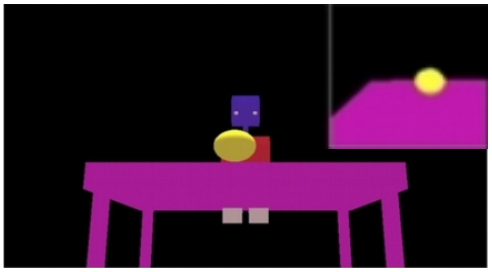

FIGURE 5.1: Shows iCub scanning the sphere object. Inset on top right shows the object from the iCub point of view

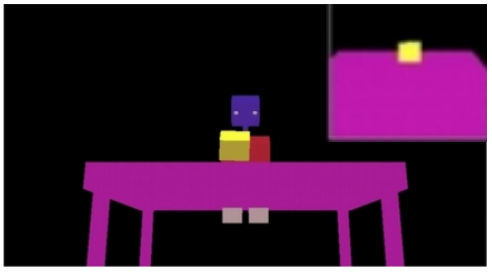

FIGURE 5.2: Shows the iCub scanning the cube object. Inset on top right shows the object from the iCub point of view

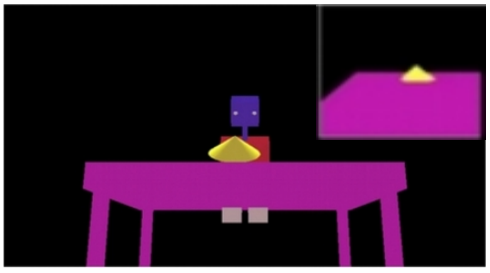

FIGURE 5.3: Shows iCub scanning the cone object. Inset on top right shows the object from the iCub point of view

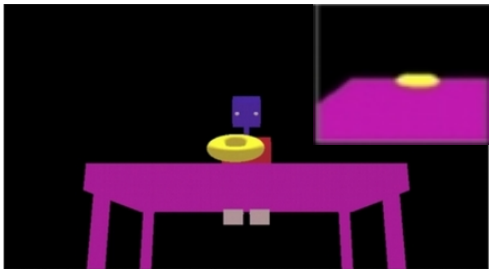

FIGURE 5.4: Shows the iCub scanning the torus object. Inset on top right shows the object from the iCub point of view

The experimental scenario involved a simulated iCub agent equipped with a right-eye vision. The agent was situated in a 3D environment in-front of a coloured object on a coloured table against a black background (e.g. Fig. 5.1). We chose four different coloured objects, i.e. a sphere, cube, cone and torus, in which the stimuli are highly ambiguous, and render the categorisation task more arduous. The four different coloured objects were presented to the agent for categorisation one at a time (Fig. 5.1, Fig. 5.2, Fig. 5.3 and Fig. 5.4). In each presentation, the objects were uniformly randomly scaled with a variation of [-10%,10%] to the original size, and uniformly randomly rotated within the range $[-10°, 10°]$ on the y axis. In each trial, the agent eye perceived each object presented with visual extraction from grey-scale averaging [23], ULBP [1] or HOG [2].

It is also very important to emphasise here that the agent located in the 3D environment never perceived the entire environment at the same time as the virtual camera was located in the eye position and the degree of freedom was limited to that of the eye. Therefore, the part of the object and the table perceived each time step was determined by eye orientation as a result of the pan and tilt, as shown in the iCub view, inset in Fig. 5.1, Fig. 5.2, Fig. 5.3 and Fig. 5.4.

5.2.2 The neural network controller

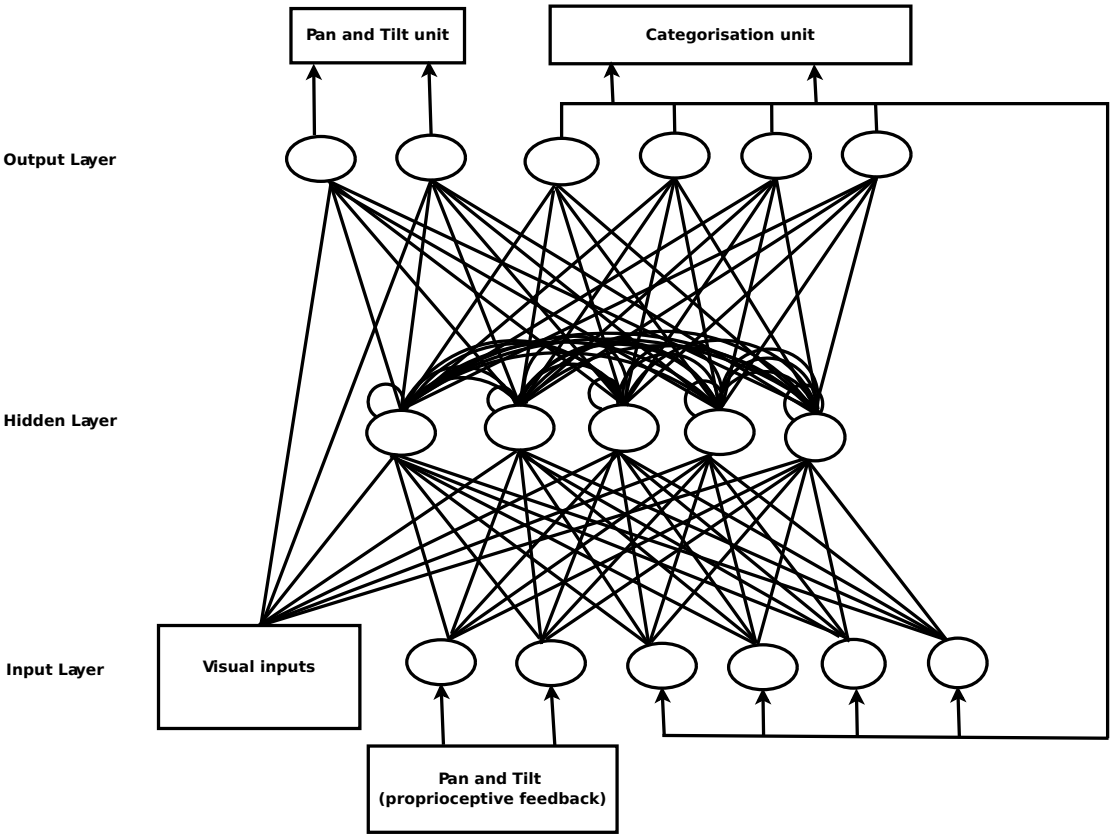

FIGURE 5.5: The architecture of the Continuous Recurrent Neural Network controller. In the input layer: the left block consists of the visual inputs of one of the three visual extraction methods, the middle block of two input units encodes the state of the proprioceptive inputs from pan and tilt, and the last four inputs encode the state of the categorisation output units at the previous time step. The hidden layer has five hidden recurrence neurons, while the left and right blocks of the output layer are the two units for the pan and tilt, and four units for categorisation respectively at time step t

The neural network is a three-layer continuous time recurrent architecture inspired by [23], with the updates equation as described in Chapter 3 (Fig. 5.5). It has an input-layer whose vector size is determined by the method of visual extraction. It also has one hidden layer of 5 recurrent neurons, and an output layer of 6 neurons.

In the output layer, 2 of the neurons determine the movement, i.e. the pan and tilt in the iCub visual motor space, and the other 4 neurons are for labelling the categories per time step. The input layer consists of units which encode the current activation state of the neurons of the retina region, the copies of 4 classification units at previous time step $t-1$, and proprioceptive information of the pan and tilt (i.e. normalised pan and tilt values between 0 and 1, as pan_{input}, and $tilt_{input}$). A random value of uniform distribution within the range of [-0.05,0.05] was added to the inputs of the visual stimuli in order to take into account that sensor data are subject to noise.

5.2.3 The task and the evolutionary process

The agent was evaluated for 48 trials in which each of the four objects (sphere, cube, cone and torus) was presented to the iCub agent 12 times; and each trial lasting 100 time steps (a presumably sufficient length of time for exploration in a trial). At the beginning of each trial: (i) each object was randomly scaled, rotated and presented to each individual (iCub agent); (ii) the state of the internal neurons of the agent's controller was initialised to 0.0; and (iii) the eye was initialised in each quadrant of the iCub gaze-space, but randomly located in each initialisation within a quadrant, and with the object within the eye view. During each time step of a trial, we calculated the pan_{step} and $tilt_{step}$ and normalised their updates and input as proprioceptive feedback (pan_{input}, and $tilt_{input}$) along with the categorisation outputs at the previous time step into the network (as described in Chapter 3). In each trial the eye was left to freely explore the environment; however, in order to save time and improve exploration, a trial was terminated when the eye (pan or tilt) reached the iCub pan limit ([-0.523616, 0.523616] radians) or tilt limit ([-0.663243, 0.314177] radians) for three consecutive time steps. The task of the agent was to correctly label the category of the current object during the second half of the trial, i.e. when the agent had explored the environment for a sufficient duration of time.

The initial population consisted of 60 randomly-generated genotypes within the range $[0, 1]$, each encoding the free parameters which were determined by a genetic algorithm for the corresponding neural controller, and included all the connection weights, gain factors, biases, and time constants of the hidden neurons. In order to generate the phenotypes, weights and biases were linearly mapped in the range $[-10, 10]$ and $[-5, 5]$ respectively, while the time constants were mapped in the range $[-1, 2.2]$. Subsequent generations to the first were produced by a combination of selection with elitism, recombination and mutation. For each new generation "the elite", i.e the genotype with the highest fitness value was copied unchanged from the previous to the new generation, while the worst 10 were dropped. The remaining 59 genotypes of the new generation were

formed by randomly selecting two genotypes from the older generation using roulette wheel selection scheme, and a new genotype was formed by combining the genetic material of these two old genotypes with a probability of 0.3 with a cross-over point selected during the recombination. Mutation, which entails that a random Gaussian offset was applied to each real-valued component encoded in the genotype, was performed with a probability of 0.04. The mean was 0 and the standard deviation was 0.1. Note: the parameter values indicated above for both the genotype/phenotype (network controller) mapping and the genetic algorithm were adopted from Tuci [29].

5.2.4 Visual Extraction Methods

We investigated three visual extraction methods that were used as sensory inputs into neural network controlled active vision system. We present the three methods, i.e grey-scale averaging [23], Uniform Local Binary Patterns [1] (as Active-ULBP) and Histogram of Oriented Gradients [2] (as Active-HOG) as they were used in the experiment. In the 3D experiment we have used a larger active window size of 100 x 100 pixels for the following reasons:

(i) The larger window size of 100 x 100 pixels (as opposed to the window size of 50 x 50 pixels used in the 2D experiment) would give more visual information with a view to improving the object categorisation performance.

(ii) Preliminary experiments using 2 replications of an evolutionary run of 5000 generations for object categorisation for the 100 x 100 pixels window size showed significant improvement over window size of 50 x 50 pixels. Also, initial work estimating object location for 1 replication of an evolutionary run of 5000 generations showed better performance using a window size of 100 x 100 pixels rather than a window size of 50 x 50 pixels.

Given the long timescales involved in the evolutionary training it was decided to only proceed with 100 x100 pixels.

Grey-scale averaging

The grey-scale averaging method [23] of visual vector size 25 discussed in Chapter 3 and as used in Mirolli et al. [23] was instantiated in every time step of a trial to process the visual stimuli of the receptive field of the iCub. The visual input to the neural network in every time step was as shown in Fig. 5.7.

FIGURE 5.6: shows the grey-scale image patch of the area covered by the iCub retina.

FIGURE 5.7: shows the grey-scale average values that were input to the neural network.

Active-Uniform Local Binary Patterns

We instantiated the Active-ULBP of feature vector size 236 discussed in Chapter 3 as a pre-processing method of the visual receptive field. The features were extracted as uniform patterns from four equally divided cells of the receptive field and concatenated to form a vector size of 236 (Fig. 5.8). The normalised histograms were input to the neural network in every time step of the trials of an evolutionary run.

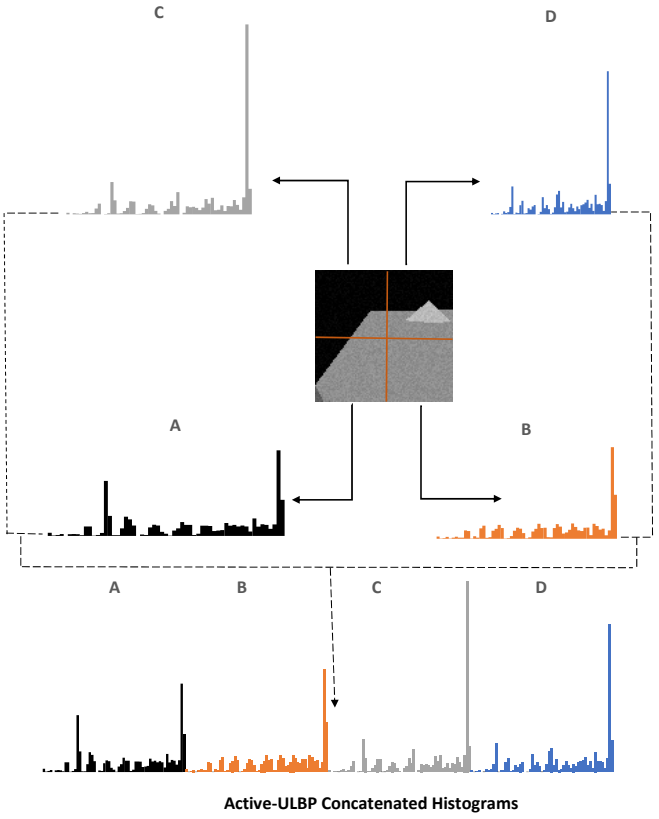

FIGURE 5.8: shows the concatenated Active-ULBP histogram features that were normalised and input into the neural network

Active-Histograms of Oriented Gradients

In order to further investigate the use of pre-processing techniques for object categorisation in the 3D environment, the Active-HOG described in Chapter 3 was instantiated for feature extraction of the simulated iCub receptive field region. The features extracted as gradient magnitudes in 9-histograms bins of four cells in the receptive field were concatenated and normalised as an input vector of the neural controller (Fig. 5.9).

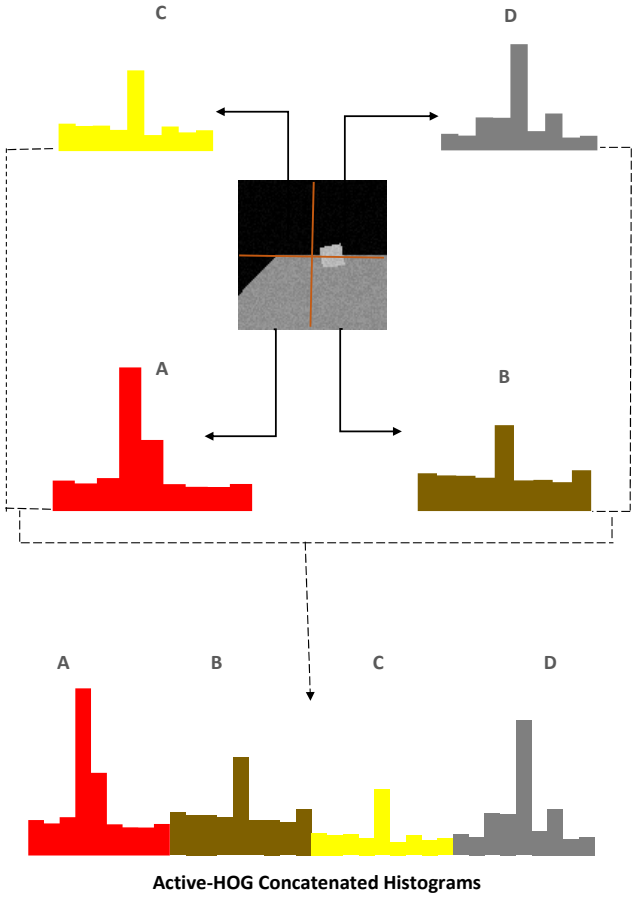

FIGURE 5.9: shows the concatenated Active-HOG histogram features that were normalised and input into the neural network

5.3 Results

In this section, we present the results and the analysis of all three methods of visual extraction used by the active vision system for the object categorisation. We assess the ability of the iCub agent to correctly categorise the objects by calculating the percentage of times in the second half of each trial, the categorisation unit corresponding to the correct object was the most activated. Finally, we give a comparative analysis of the

evolution, performance and dynamic process of categorisation used by the active vision system for the three methods of visual extraction.

5.3.1 Evolution

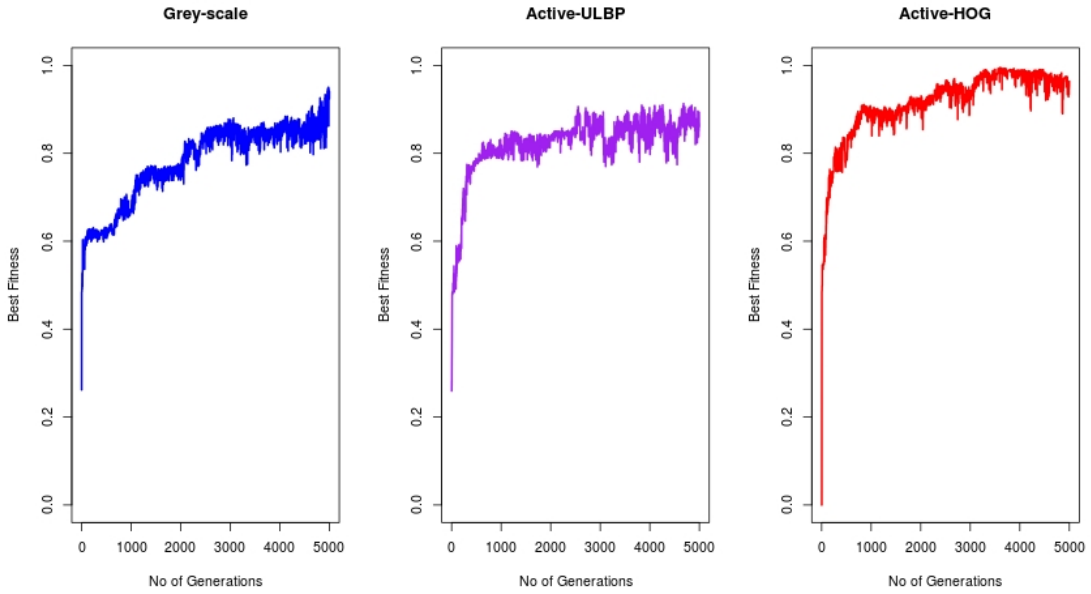

FIGURE 5.10: The best fitness graphs of the best evolutionary runs of the three methods of visual extractions. **Left**: The best fitness graph of the best run of the grey-scale averaging method. **Middle**: The best fitness graph of the best run of the Active-ULBP method. **Right**: The best fitness graph of the best run of the Active-HOG method. The y-axis represents the fitness of the best evolved genotype of each generation, while the x-axis represents the number of generations.

In the evolution of the active vision system, we performed 6 evolutionary runs for each of the visual extraction techniques (as shown in Appendix B, Fig. B.1, Fig. B.2 and Fig. B.3). Each evolutionary run lasted 5000 generations with 48 trials for each genotype, and 100 time steps in each trial. We present here a comparison of the best runs and all evolutionary runs for the three methods of visual extraction. Comparing the pattern of evolution in the best runs of the three methods of visual extractions (Fig. 5.10), the grey-scale and the Active-ULBP start at the same level with a jump-start in the fitness to approximately 0.24 in both methods, but the curves differs from about the 0.6 fitness mark and the grey-scale peaks at a slightly higher level than the Active-ULBP at the end of the runs. On the other hand, the Active-HOG starts from the 0 fitness mark and increases sharply to about 0.87 in around 1000 generations, and terminates at a higher fitness value than the other two methods. Also, comparing the pattern of fitness of all runs of the three visual extraction methods, which shows the mean of the best fitness in all generations of all evolutionary runs and their positive and negative standard deviation from the mean (Fig. 5.11), one can observe that the mean fitness pattern of the

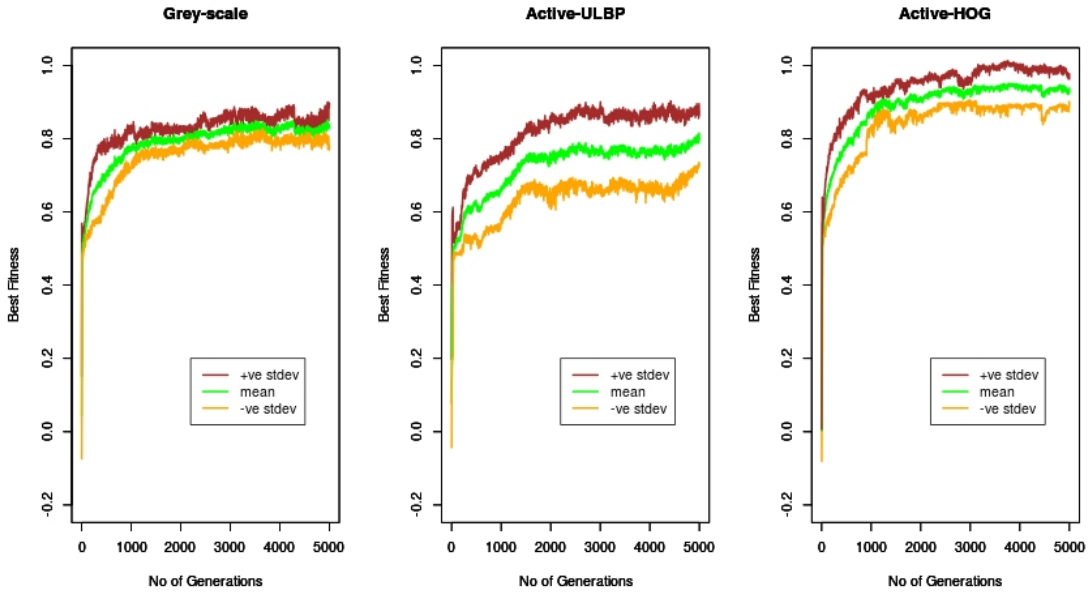

FIGURE 5.11: Shows the graph of the mean (average) of all best fitness in each generation of the 5000 generations for 6 evolutionary runs and their positive (+ve stdev) and negative (-ve stdev) standard deviation in each generation for the three methods of visual extraction

Active-HOG was generally higher than that of the other two methods in all generations of the evolutionary runs, while that of the grey-scale was a bit higher than that of the Active-ULBP. This suggest that Active-HOG fitness values over all generations in all evolutionary runs were generally higher than those of the other two methods. The grey-scale also generated higher fitness values in all runs than the Active-ULBP. Also, comparing the patterns of standard deviations, the grey-scale generally had a closer deviation to the mean than the other two methods, while that of Active-HOG was also closer than that of Active-ULBP. This shows that the variation in the individual fitness of each generation in all runs was smaller in the grey-scale method than the other two methods. The differences of individual fitness of Active-HOG were also smaller than the Active-ULBP in all runs.

5.3.2 Categorisation Performance

We assessed the performance of the system using the best evolved genotypes of 100 consecutive generations that had a relatively higher and more stable fitness pattern as compared to the other generations in all evolutionary runs. This differs from the 2D experiment described in Chapter 4, where we took a more systematic approach by re-evaluating the best genotypes of the last 1000 generations. The number of genotypes chosen for re-evaluation has been reduced in order to keep the re-evaluation time within reasonable limits, considering the high computational costs of the 3D experiments. Also

we did not limit ourselves to the re-evaluation of the genotypes of the last 100 generations, since in various runs these solutions turned out not to be among the most successful when compared to solutions that appeared in other evolutionary times.

TABLE 5.1: The average performance of the best performing re-evaluated genotype of **grey-scale averaging** in all trials of the testing stage.

	Percentage of Correct Categorisation (Highest in Bold)			
Current category	**sphere**	**cube**	**cone**	**torus**
sphere	**98.92**	0.00	0.00	1.08
cube	0.00	**100.00**	0.00	0.00
cone	0.00	0.00	**93.04**	6.96
torus	0.49	0.00	16.92	**83.08**

TABLE 5.2: The average performance of the best performing re-evaluated genotype of **Active-ULBP** in all trials of the testing stage.

	Percentage of Correct Categorisation (Highest in Bold)			
Current category	**sphere**	**cube**	**cone**	**torus**
sphere	**97.48**	0.4	2.12	0.00
cube	8.00	**92.00**	0.00	0.00
cone	0.00	0.00	**89.02**	10.98
torus	0.00	0.00	26.38	**73.62**

TABLE 5.3: The average performance of the best performing re-evaluated genotype of **Active-HOG** in all trials of the testing stage.

	Percentage of Correct Categorisation (Highest in Bold)			
Current category	**sphere**	**cube**	**cone**	**torus**
sphere	**98.00**	0.00	0.00	2.00
cube	0.00	**100.00**	0.00	0.00
cone	0.08	0.00	**99.92**	0.00
torus	0.00	0.00	0.00	**100.00**

The system was tested on the four categories of object used in the training by randomly scaling and rotating each object presented in a trial. The objects were randomly scaled within the range [-15%, 15%] relative to their original size and rotated in the range $[-10°, 10°]$ on the y axis, with a uniform distribution. A total of 200 trials were performed, with each object presented 50 times to the agent in all trials and the eye was initialised in each quadrant of the iCub gaze-space, but randomly located in each initialisation within a quadrant, and with the object within the eye view.

The categorisation performance was based on the percentage of times in which the categorisation unit corresponding to the correct category was the most activated in all trials. We discuss here the general trends in the categorisation performance for all the genotypes re-evaluated here. Tables 5.1, 5.2 and 5.3 show the confusion matrices of the

best performing re-evaluated genotypes for the grey-scale, Active-ULBP, and Active-HOG methods respectively. The three tables show that the current (correct) categories had the highest average categorisation performance in all trials for the categorisation tasks. One can also deduce that Active-HOG had the best performance, while grey-scale also outperformed Active-ULBP.

TABLE 5.4: The statistics of the best performing re-evaluated genotypes in all runs for each visual extraction methods.

Visual extraction methods	Max	Average	Worst	Stdev
Grey-scale averaging	93.76	74.47	66.19	±12.01
Active-ULBP	88.03	68.53	49.36	±13.32
Active-HOG	99.48	98.07	95.08	±1.9

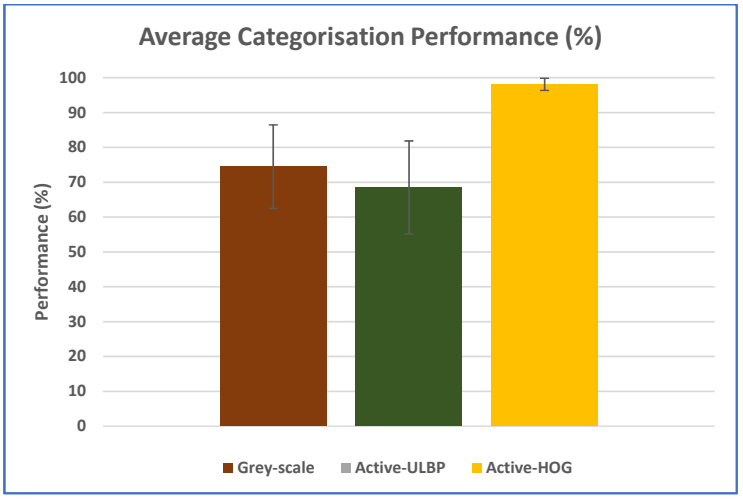

FIGURE 5.12: Bar-charts showing the average categorisation performance of the three methods of visual extraction in all runs

Overall performance of the genotypes re-evaluated for all evolutionary runs of the three visual extraction methods can also be estimated by observing Table 5.4. The metrics used are: **Max** indicates the best performance from all the genotypes re-evaluated in all runs; **Average** is the average of the best performance in each run; **Worst** is the worst of the best performances in each run; and **stdev** is the standard deviation of the best performance of all runs. From the table one can see that Active-HOG had the overall best performance of 99.48% as compared to that of grey-scale (93.76%) and Active-ULBP (88.03%). Active-HOG, also, had a higher average performance of 98.07% as compared to that of grey-scale (74.46%) and Active-ULBP (68.53%). It also had the best worst performance of 95.08% as compared to that of grey-scale (66.19%) and Active-ULBP (49.36%). Looking at the standard deviation values, the performance values achieved by Active-HOG for all best performing genotypes in all runs were less sparsely distributed

than those of Active-ULBP and grey-scale. The average categorisation performance of all the visual extraction methods in all runs are also shown in Fig. 5.12.

Finally, we can deduce from the summary of the performance results that the three methods of visual extraction of the active vision system have actually translated the skills learned during training to actual categorisation performance in the testing stage.

Statistical Analysis

As was done in Chapter 4, we used an extended version of a t-test to detect if significant differences exist between the averages of the three visual extraction methods. We used the extended version of the t-test also known as ANOVA which is commonly used in significance testing among three or more means (averages). We tested the significance of the differences of the averages with the (p- value$<$0.05) and a more demanding (p-value$<$0.01). Table 5.5 shows the statistical summary of the three visual extraction techniques that were used to calculate the results of the ANOVA. The first column, **Visual extraction methods** indicate the visual extraction techniques, the number of evolutionary runs is indicated by **Count**, the sum of the individual performance of the best performing re-evaluated genotypes from all runs of the three visual extraction methods is represented by **Sum**, while the **Average** and **Variance** indicate the averages and variance of the performance of the best performing re-evaluated genotypes from all runs of the three methods. In the second table for the ANOVA test, Table 5.6, the first column represents the **Source of variations** between and within the groups (visual extraction methods) of which the averages were compared, **SS** represents the sum of squares, **df** represents the degree of freedom, **MS** represents the group means squares, **F** is the F distribution value, **P-value** indicates the significance level of the differences of averages that were considered and **F crit** represents the F critical value.

The obtained p-value of 0.0004 as shown in Table 5.6 is less than our first and second significance level of 0.05 and 0.01 respectively and this indicates strong evidence against the null-hypothesis that the averages for the three visual extraction methods were equal, and therefore we reject the null hypothesis based on these two p-values (0.05 and 0.01). Since the null hypothesis was rejected, we carried a Bonferroni correction for the two significance levels of 0.05 and 0.01 to ensure that the overall significance level does not exceed these two values as the significance level of each individual t-test to be carried out. As shown in Table 5.7, the Bonferroni corrected value for a significance level of 0.05 is 0.0167, while that of 0.01 is 0.003. The result of the t-test of the averages of the three groups (visual extraction methods) using Bonferroni correction is shown in the Table 5.7. In the table, the first column indicates the compared paired groups, the

TABLE 5.5: Summary of the statistics of the best performing re-evaluated genotypes of the three visual extraction methods from 6 evolutionary runs that were used in the anova test

SUMMARY				
Visual extraction methods	***Count***	***Sum***	***Average***	***Variance***
Grey-scale averaging	6	446.79	74.47	144.31
Active-ULBP	6	411.17	68.53	177.40
Active-HOG	6	588.44	98.07	3.61

TABLE 5.6: The results of the anova test

ANOVA						
Source of Variation	***SS***	***df***	***MS***	***F***	***P-value***	***F crit***
Between Groups	2930.97	2	1465.49	13.51	0.0004	3.68
Within Groups	1626.59	15	108.44			
Total	4557.56	17				

TABLE 5.7: The significant test results using a paired t-test with test condition of (p-value<0.05) and (p-value<0.01)

			Signf. Level=0.05	***Signf. Level=0.01***
Compared Groups	***t-value***	***p-value***	***Bonf. Corr=0.0167***	***Bonf. Corr=0.003***
Active-HOG and Grey-scale	3.93	0.0014	Significant	Significant
Active-HOG and Active-ULBP	4.91	0.0002	Significant	Significant
Grey-scale and Active-ULBP	0.99	0.2371	Not Significant	Not Significant

second and third columns indicate the t-values and p-values of the compared averages of the paired groups, while the fourth and fifth column indicate the level of significance of paired group average comparison based on the Bonferroni corrected p-values.

Comparing the three groups in the table at the 0.05 significance level with the Bonferroni correction of 0.0167, the variation in averages of Active-HOG and grey-scale was significant, while those of Active-HOG and Active-ULBP were also significant. However, the variation in averages of grey-scale and Active-ULBP was not significant, which means that the resultant difference could have been by chance. Therefore, for the significant level of 0.05, we reject the null-hypothesis that the averages of the two groups Active-HOG and grey-scale, and Active-HOG and Active-ULBP were equal, while we fail to reject the null hypothesis for that of grey-scale and Active-ULBP.

Furthermore, considering the significance level of 0.01 with Bonferroni correction of 0.003, the difference in averages of Active-HOG and grey-scale was considered strongly significant, while those of Active-HOG and Active-ULBP were also strongly significant. The difference in averages of grey-scale and Active-ULBP, however, was not significant, which means that the resultant difference could have been by chance. Therefore, for the significance level of 0.01, we reject the null-hypothesis that the averages of the two groups

Active-HOG and grey-scale, and Active-HOG and Active-ULBP were equal, while we fail to reject the null hypothesis for those of grey-scale and Active-ULBP

5.3.3 Dynamics of Categorisation Process

In this section we investigate the process of object categorisation in the 3D environment. In particular, we examine:

(i) To what extent the sensory patterns provided by the three visual extraction methods and experienced by the agent during interaction with the objects have been able to provide the discriminative stimuli that facilitate the categorisation process.

(ii) To what extent the agents succeed in self-selecting the stimuli that are associated with a particular category.

Note: stimulus ambiguity may depend on the nature of the stimulus, the field of view of the agent and the eye location.

The classification outputs of our system depend on the visual information that was provided, apart from the copy of the outputs of categorisation and motor units at the previous time step. However, since our focus is mainly on the influence of visual features on the control of the active vision in order to improve learning for categorisation, we only investigate the visual sensory channel. In order to carry out this investigation, we extend the Modified Geometric Separability Index (MGSI) proposed in [23] and described in Chapter 4 to the 3D domain for object categorisation. We computed the MGSI of the best performing re-evaluated evolved genotypes for all three visual extraction methods in all evolutionary runs for 200 trials during which the agent experienced the stimuli from the four categories (i.e, sphere, cube, cone, and torus), where each object was uniformly and randomly scaled between [10%, -10%] to the original size and rotated within the range $[-10°, 10°]$ relative to the original orientation with 50 different initial eye positions. For each type of visual extraction method using the sensory patterns, the MGSI was computed for each of the 100 time steps of a trial (Fig. 5.13, Fig. 5.14, Fig. 5.15).

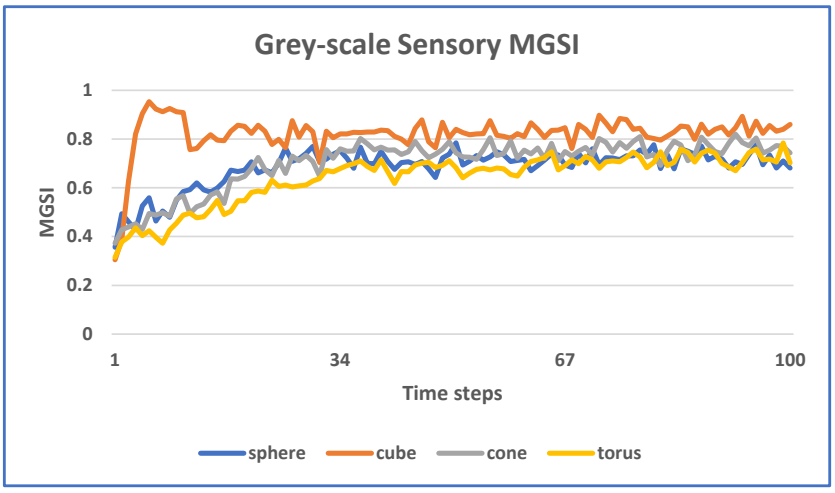

FIGURE 5.13: Modified Geometric Separability (MGSI) of the stimuli provided by grey-scale averaging

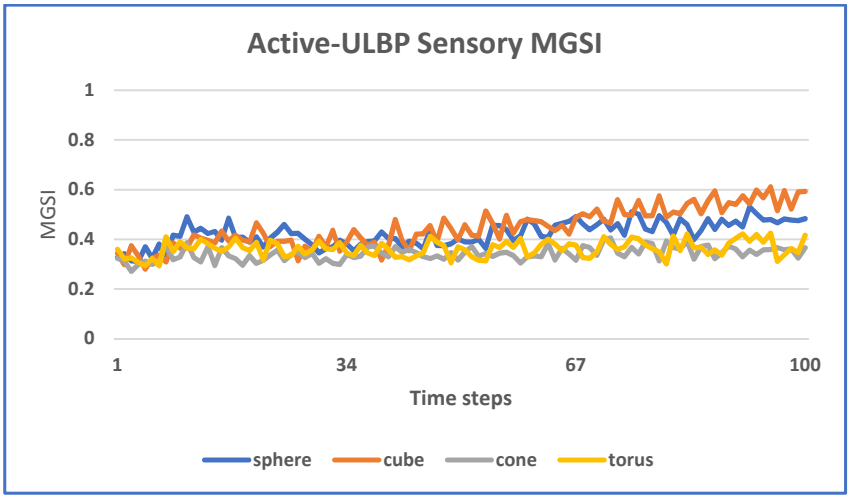

FIGURE 5.14: Modified Geometric Separability (MGSI) of the stimuli provided by the Active-ULBP method

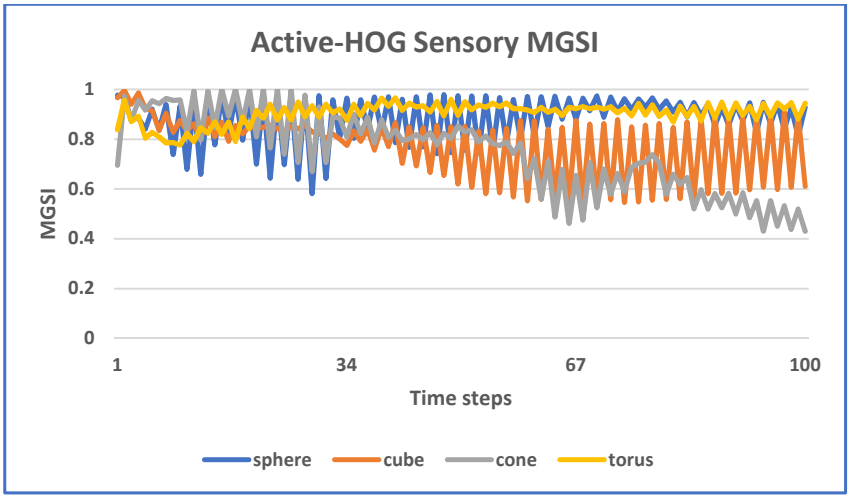

FIGURE 5.15: Modified Geometric Separability (MGSI) of the stimuli provided by the Active-HOG method

(i) The fact that the MGSI increased for grey-scale, showed that the system moved away from very ambiguous to more discriminative stimuli when using the grey-scale visual extraction method.

(ii) The fact that the MGSI only showed modest improvement for the Active-ULBP, i.e. mainly for the sphere and cube, showed that the system could only use slight intelligent coordinated motor behaviours to experience less ambiguous stimuli over time for these two objects.

(iii) Active-HOG generated less ambiguous stimuli than grey-scale and Active-ULBP. This means that it produced more discriminative stimuli in the input space than the other visual extraction methods. In fact, the Active-HOG MGSI reached 1.0 in some time steps. This means that Active-HOG sensory patterns experienced at this time steps were fully discriminative.

(iv) The fact that Active-HOG MGSI generally did not show improvement over time and even deteriorated in the case of cone object, showed that the system was not able to move to less ambiguous stimuli over time when the stimuli were generated by the Active-HOG visual extraction method.

(v) Active-HOG exhibited some kind of oscillatory behaviour in most time steps for all the objects. This might have been some kind of complex behaviour developed by the system as a result of the reduced ambiguity provided by the Active-HOG stimuli from the start, and, as such, there was not much need in this case to use the eye movements to reduce ambiguity.

(vi) The MGSI never reached a value of 1 for the grey-scale and Active-ULBP. This means that the system never managed to generate unambiguous stimuli for these two visual extraction methods. However, the system was still able to achieve overall success rates.

(vii) For some objects, the system managed to generate less ambiguous patterns than for other objects. This means that the system produces more discriminative patterns for those objects than the others.

5.4 Discussion

We have extended the evolutionary active vision system to the 3D environment for object categorisation using our benchmark architecture (Mirolli et. al. [23]). We have chosen the iCub platform (iCub) because it will help to show the plausibility of our methods in complex artificial systems. We started this chapter by extending the evolutionary active

vision for object categorisation in the 3D environment using the grey-scale averaging method [23] as the visual extraction method. We further sought to improve system performance in the 3D environment for object categorisation using two pre-processing methods from computer vision, i.e. Uniform Local Binary Patterns [1] (as Active-ULBP) and Histogram of Oriented Gradients [2] (as Active-HOG). We discuss: (i) visual representation and active vision categorisation; and (ii) learning control of the active vision system.

Visual Representation and Active Vision Categorisation

As we previously did in the 2D environment, we investigated three visual extraction methods, that is (i) grey-scale averaging [23], (ii) Uniform Local Binary Patterns [1] (as Active-ULBP), and (iii) Histogram of Oriented Gradients [2] (as Active-HOG) as visual representations for the active vision in the 3D environment. The Active-HOG achieved higher average performance than the other two visual extraction methods. grey-scale also performed better on average than Active-ULBP.

Furthermore, the results of the statistical analysis using 0.05 and 0.01 significance levels showed that there was a significant difference between the average performance of Active-HOG when compared to the other two visual extraction methods. However, there was no significant difference between the average performance of grey-scale when compared to that of the Active-ULBP. This implies that using the two significant levels, the higher average performance of grey-scale over Active-ULBP might have been by chance.

The fact that Active-HOG performed better than grey-scale and Active-ULBP in the 3D object classification scenario may be due to the more structural nature of object categorisation problem. This boosts the credentials of HOG as an effective feature descriptor for applications that involve structures e.g. object detection [51] and human recognition [2]. The fact that Active-ULBP also demonstrated good performance provides further evidence of ULBP as an effective feature descriptor in many applications [45][203].

Finally, the performance of the visual extraction methods may further show that pre-processing methods in computer vision can have great applicability for visual representation in active vision systems.

Learning Control of the Active Vision System

Sensory-motor coordination helps biological agents to interact with their visual environment by intelligently using their motor mechanism to exploit regularities in this environment that enhance vision problems. This intelligent cooperation can be greatly

dependent on local visual information perceived each time which guides the active vision system to experience stimuli that enhance the vision task and at the same time avoid disruptive information. Replication of this process in artificial systems, such as in a robot may greatly improve the tackling of vision tasks, such as object categorisation. Since intelligent control of the motor mechanism can be influenced by the kind of visual information that is being perceived, it is therefore imperative to investigate how visual representation can contribute to learning in active vision systems.

From the perspective of the three visual extraction methods investigated in this chapter, only the grey-scale has been able to significantly use intelligent control of the active vision system to experience more discriminative stimuli over time, given the improvement of the Modified Geometric Separability Index (MGSI). Active-ULBP, by contrast, showed very little improvement over time. The fact that Active-HOG generated less ambiguous stimuli than the other methods, may be due to the gradient features provided by Active-HOG, and which might have also enhanced the recognition of the 3D structural problem. However, the stimuli provided by Active-HOG did not help the system to move to less ambiguous stimuli over time. This may be because of the unambiguous stimuli provided by Active-HOG from the start. They probably did not give the system much need to use eye movements as a strategy in solving the categorisation tasks since the behaviours of an agent are partially determined by the sensory stimuli experienced [105].

Also, the oscillatory behaviour developed by the system as a result of the stimuli provided to it by the Active-HOG was probably a strategy the system developed in order to continue to experience features with low ambiguity, and which led to good performance. However, we are not absolutely sure of this, and a future investigation may be needed to examine this further. The ability of the system to use the stimuli provided by the Active-HOG in solving the categorisation tasks nevertheless shows that HOG is an effective feature descriptor for structural applications since it helped to reduce ambiguity in the object categorisation task.

5.5 Chapter Summary

In this chapter we have extended an evolutionary active vision system for object categorisation from 2D to 3D with the grey-scale averaging visual representation method [23], and sought further improvement in performance with two pre-processing techniques in computer vsion, i.e. Uniform Local Binary Patterns [1] (as Active-ULBP) and Histogram of Oriented Gradients [2] (as Active-HOG).

The best grey-scale performance of all replications of evolutionary runs was 93.76% and the average performance was 74.47%. The best Active-HOG performance of all runs was 99.48% and the average performance was 98.07%. In the case of Active-ULBP, the best performance was 88.03% and the average performance was 68.53%.

Statistical analysis that compared the average performance of the three visual extraction methods showed that the higher average performance of Active-HOG over the other two methods was significant. On the other hand, the higher average performance of grey-scale over that of Active-ULBP was not significant. This means that the better performance of grey-scale in comparison to Active-ULBP might have occurred by chance.

Analysis based on a Modified Geometric Separability Index (MGSI) showed that the stimuli provided by grey-scale helped the system to experience more discriminative stimuli over time than the other two methods. Active-ULBP also showed very little improvement, while Active-HOG generally did not show any improvement, even though the stimuli provided by it were less ambiguous from the start than the other two methods.

Analysis also showed that since MGSI never reached a value of 1 for the grey-scale and Active-ULBP cases, and in very few time steps did so in the Active-HOG case, the categorisation process may also have involved some kind of integration of perceptual information over time.

The next chapter further extends the evolutionary active vision system for indoor and outdoor environment categorisation in 3D using the iCub robot platform.

Chapter 6

Experiment 3: Gaze Control in 3D Environment Categorisation

6.1 Introduction

In the last chapter, we extended our gaze control framework to object categorisation in the 3D environment. We also showed that pre-processing can be used to enhance active vision object categorisation in the 3D domain. In this chapter, we further extend our evolutionary active vision system with pre-processing into the problem of indoor and outdoor environment classification in 3D using the Humanoid (iCub) platform. The extension to this other classification domain is necessary mainly because it has a different problem structure and therefore different sensory-motor strategies are expected to be used in addressing it. Thus, it will give a greater need to use the eye for exploration when compared to object categorisation, and as such give a more objective and conclusive means in answering our research questions. In Section 6.2, we describe the general experimental set-up, and in Section 6.3 we provide the results. Section 6.4 gives a general discussion of the chapter and the results, and finally in Section 6.5 a summary of the chapter is given.

6.2 Experimental Set-Up

To investigate how a simulated agent (the iCub) can exploit its eye movement in the classification of indoor and outdoor environments, a simulated robot controller was trained using an evolutionary technique. We also investigated if classification performance could be improved with pre-processing techniques. The encoded free parameters of the evolutionary technique that regulate how the agent interacts with these two environments

(indoor and outdoor) were randomly varied, and variations were retained or discarded based on the agent's ability to perform the categorisation task.

6.2.1 The iCub agent and the environment

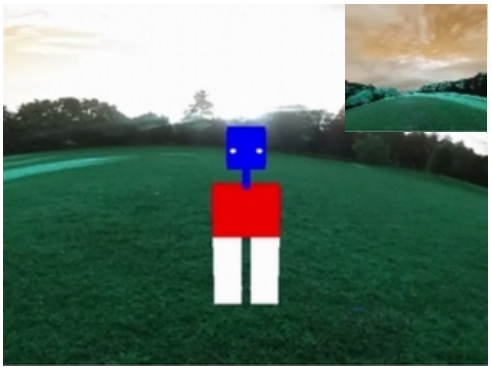

FIGURE 6.1: Shows the iCub in an outdoor environment. Inset on top right shows the environment from the iCub point of view.

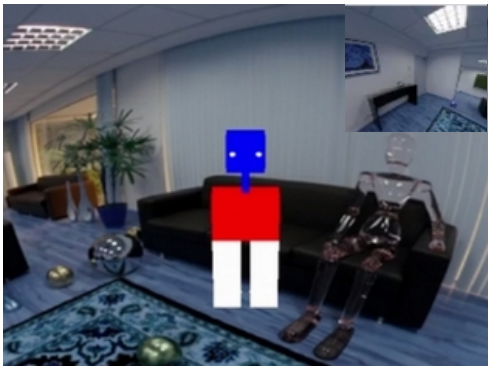

FIGURE 6.2: Shows the iCub in an indoor environment. Inset on top right shows the environment from the iCub point of view.

The experimental set-up involved a simulated Humanoid robot agent equipped with just a right-eye vision capability. The agent was situated in various 3D indoor and outdoor environments. The environments (indoor and outdoor) were represented with 20 texture images, which were downloaded from Google's image database (website)[213] using the keywords "indoor and outdoor panoramic texture images", "panoramic outdoor texture images"and "outdoor panoramic scene sphere texture map". The indoor environments were enclosed and objects were confined within the enclosed environment, while the outdoor environment were not enclosed and totally opened to the sky. The texture images were dynamically mapped to the interior of a 3D sphere containing the iCub (Fig. 6.1 and Fig. 6.2). The entire data-set of 20 texture images representing the environments were divided into 2-equal halves for training and validation sets for a 2-fold cross-validation (Appendix C, Fig. C.4 and Fig. C.5). The agent was situated in each environment one at a time and the environment randomly rotated within the range $[-40°, 40°]$ on the z axis with a uniform distribution and subsequently used its pan and tilt movement to explore the environment in each time step. The rotation of the environment ensured that the agent was always seeing different part of the environment in any given trial Fig. 6.3. The visual information perceived with the retina was processed with one of the visual extraction methods, i.e. grey-scale averaging, Active-ULBP or Active-HOG as described in Chapter 3.

It is very important to underscore here that the agent could not perceive the entire environment in each time step. Therefore, the environment was represented as texture

images mapped into the interior of a sphere, and the iCub agent was located inside it, with the virtual camera located in the eye position. For instance as shown in Fig. 6.1 and Fig. 6.2, the agent cannot see the front and back of the environment (image) at the same time, and its freedom of movement was limited to its eye. Therefore, what the agent perceived per time step was determined by eye orientation as a result of pan and tilt.

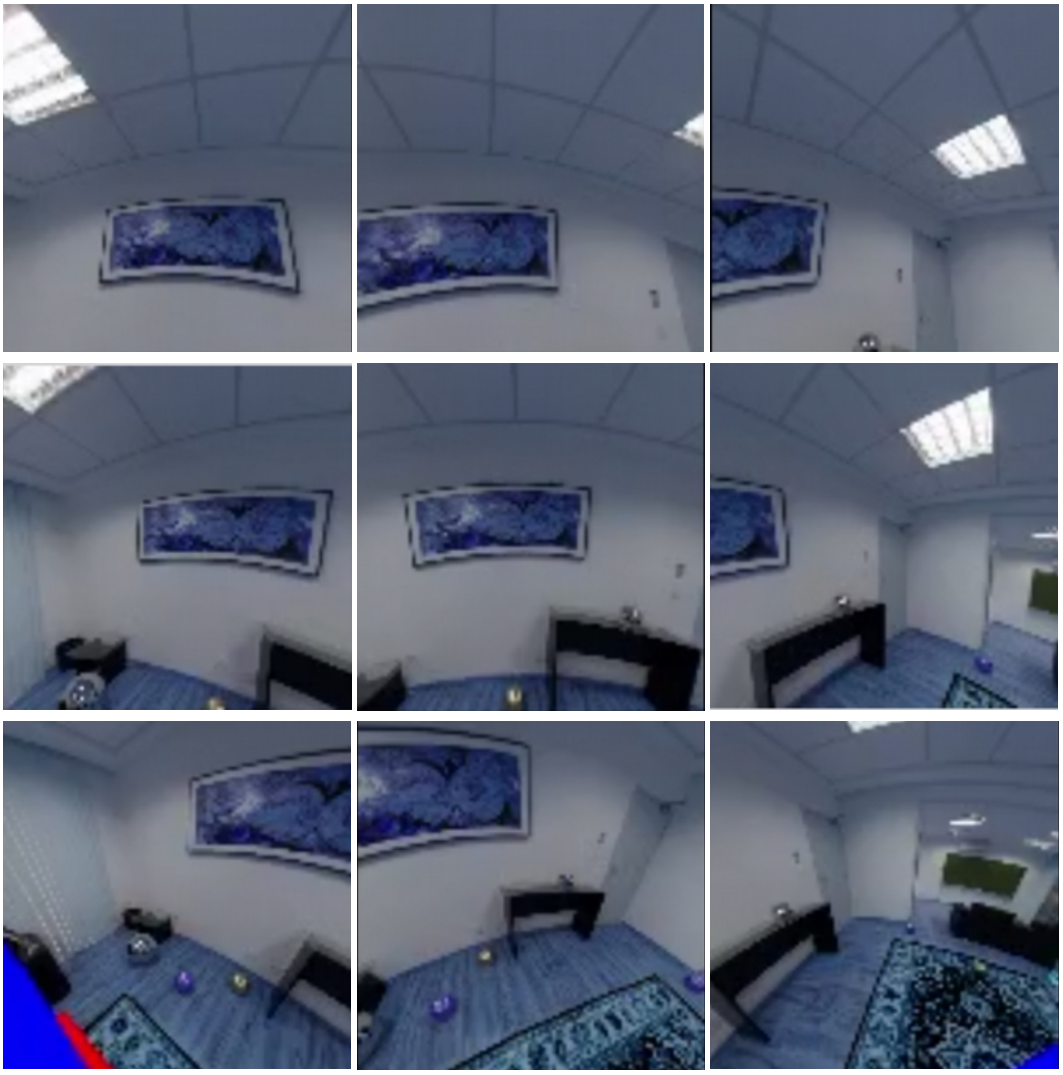

FIGURE 6.3: Shows the images of indoor environment in 9 different view directions of the simulated iCub robot.

6.2.2 The neural network controller

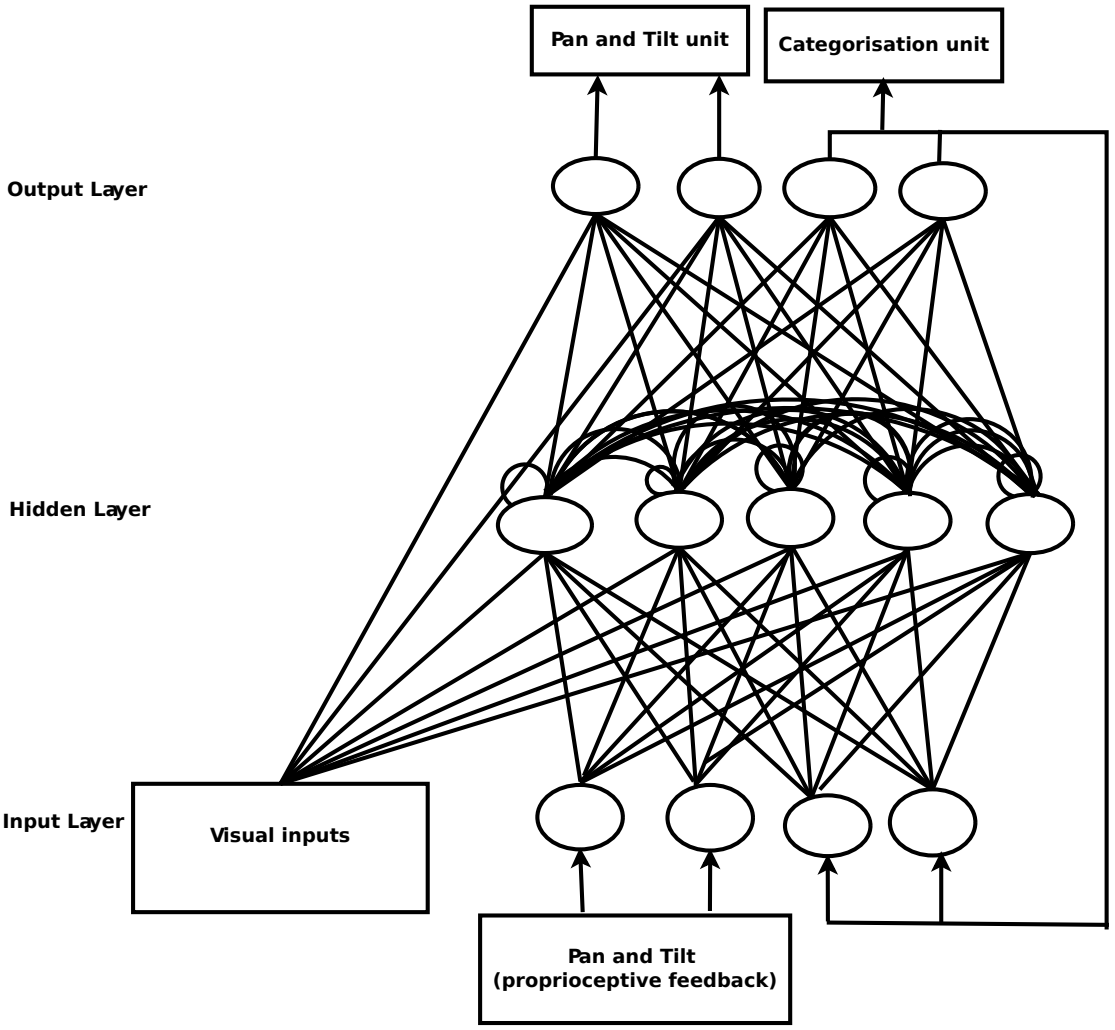

FIGURE 6.4: The architecture of the Continuous Recurrent Neural Network. On the input layer: the left block is made up of the visual inputs of one of the three visual extraction methods, the middle block of two input units encode the state of the proprioceptive inputs from pan and tilt, and the last two inputs encode the state of the categorisation output units at previous time step. The hidden layer has five hidden recurrence neurons, while the left and the right blocks of the output layer are the two units for the pan and tilt and five units of categorisation at time step t.

The simulated robot controller is a 3-layer continuous recurrent neural network inspired by [23] and with the updates equation as described in Chapter 3. It has one input-layer whose vector size is determined by the method chosen for visual feature processing. It also has one hidden layer of 5 recurrent neurons, and an output layer of 4 neurons. In the output layer, 2 of the neurons determine the eye movement, i.e the pan and tilt in the iCub visual scene, and the other 2 neurons are for labelling the categories (i.e. indoor or outdoor) per time step. The input layer consists of units which encode the current activation state of the neurons of the retina region, the copies of 2 classification

units at previous time step $t-1$, and the pan and tilt, normalised between 0 and 1 (as pan_{input} and $tilt_{input}$). A random value with a uniform distribution within the range of [-0.05,0.05] was added to the inputs of the visual stimuli processed by any of the visual extraction methods at each time step in order to simulate the effect of noise in the sensors.

6.2.3 The task and the evolutionary process

The agent was evaluated for 20 trials, with the iCub agent situated 10 times in each environment (indoor or outdoor) and each trial lasting 100 time steps (a presumably sufficient length of time for exploration in a trial). At the beginning of each trial: (i) the agent was situated in an environment (indoor or outdoor) randomly rotated in each trial; (ii) the state of the internal neurons of the agent's controller was initialised to 0.0; and (iii) the eye was initialised in each quadrant of the iCub gaze-space, although randomly located in each initialisation within a quadrant. Also, in each time step of a trial, the pan_{step} and $tilt_{step}$ values were calculated and their normalised updates were input as (pan_{input}, and $tilt_{input}$) as proprioceptive feedback along with the categorisation outputs at previous time step into the network (as described in Chapter 3). In each trial, the eye was left to freely explore the environment; however, in order to savve time and improve exploration, a trial was terminated when the eye (pan or tilt) reached the iCub pan limit ([-0.523616, 0.523616] radians) or tilt limit ([-0.663243, 0.314177] radians) for three consecutive time steps. The task of the agent was to correctly classify the environment (indoor or outdoor) during the second half of the trial, that is, when the agent had explored the environment for a sufficient length of time.

The evolutionary run began with an initial population of 60 randomly-generated genotypes in the range $[0, 1]$. Each genotype encoded the free parameters for the corresponding neural controller, and included all the connection weights, gain factors, biases, and the time constants of the hidden neurons. For the generation of the phenotypes, weights and biases were linearly mapped in the range $[-10, 10]$ and $[-5, 5]$ respectively, while the time constants were mapped in $[-1, 2.2]$. Subsequent generations to the first were produced by a combination of selection with elitism, recombination and mutation. With each new generation "the elite", i.e. the genotype with the highest fitness value was copied from the previous to the new generation, while the worst 10 were dropped. The remaining 59 genotypes of the new generation were formed by randomly selecting two genotypes from the older generation using roulette wheel selection, and a new genotype was formed by combining the genetic material of these two old genotypes with a probability of 0.3 with cross-over point selected during the recombination. Mutation, which entails that a random Gaussian offset was applied to each real-valued component

encoded in the genotype, was done with a probability of 0.04. The mean was 0 and the standard deviation was 0.1. Note: the parameter values as specified above for the genotype/phenotype (controller) mapping and the genetic algorithm were adopted from Tuci [29].

6.2.4 Visual Extraction methods

We discuss the three visual extraction methods, i.e. grey-scale averaging [23], and our proposed Uniform Local Binary Patterns [1] (as Active-ULBP) and Histogram of Oriented Gradients [2] (as Active-HOG) methods as they were used in this experiment.

Also, as we mentioned in Chapter 5, in the 3D experiment we used a larger active window size of 100 x 100 pixels for the following reasons:

(i) The larger window size of 100 x 100 pixels as opposed to 50 x 50 pixels window size used in the 2D experiment, will give more visual information which may improve the object categorisation performance.

(ii) Preliminary experiments using 2 replications of the evolutionary run of 5000 generations for object categorisation for a 100 x 100 pixel window size showed significant improvement over a window size of 50 x 50 pixels. Also, initial work estimating object location for 1 replication of the evolutionary run in 5000 generations showed better performance using a window size of 100 x 100 pixels rather than a window size of 50 x 50 pixels.

6.2.5 Grey-scale averaging

FIGURE 6.5: Shows the grey-scale image patch of the area covered by the iCub retina at a time step t.

FIGURE 6.6: Shows the image of the grey-scale average values that was input to the network at time step t.

We used the grey-scale averaging method [23] to process the perceived area of the indoor or outdoor environment by the agent in every trial of the evolutionary run. The grey-scale average vector size of 25 of the perceived area of the environment by the agent at each time step formed the sensory inputs of the neural network (Fig. 6.5 and Fig. 6.6).

6.2.6 Active-Uniform Local Binary Patterns

We instantiated the Active-ULBP with a feature vector size of 59 for each of the four cells of the iCub receptive field covering the environment stimuli at each time step. The histograms for all four cells were concatenated to form a feature vector size of 236 (Fig. 6.7). The normalised concatenated feature vector was subsequently input into the neural network.

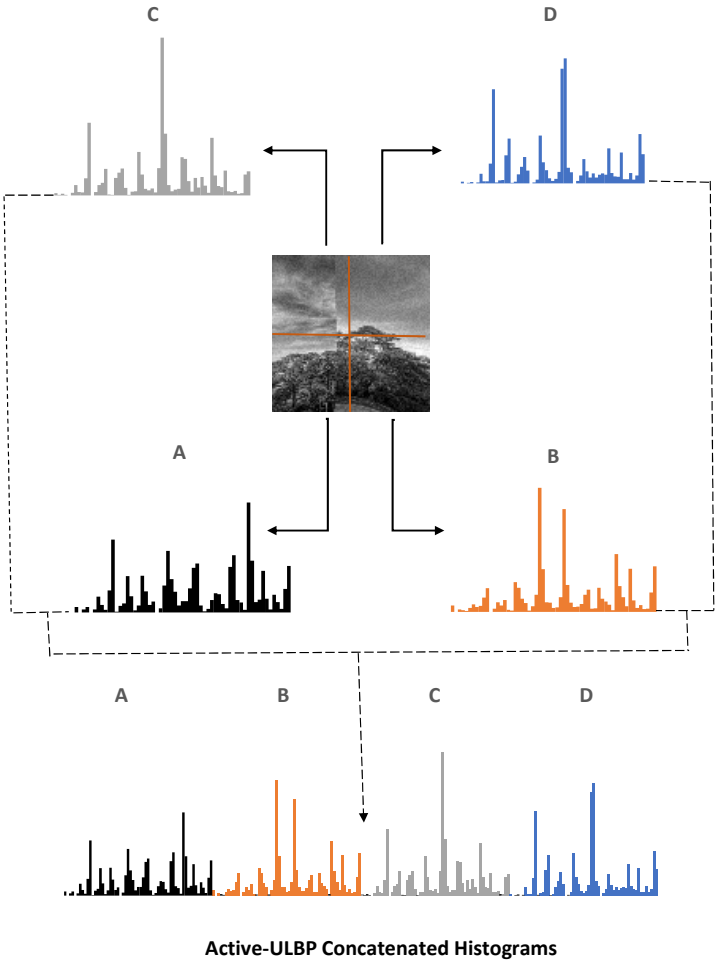

FIGURE 6.7: Shows the concatenated Active-ULBP histogram features that were normalised and input into the neural network at time step t.

6.2.7 Active-Histogram of Oriented Gradients

In each time step of a trial of an evolutionary run, the Active-HOG was used to process the environmental visual stimuli within the retina area of iCub vision, and the magnitude of the gradients in x and y direction of each pixel location were input into one of 9-histogram bins as explained in Chapter 3. The features extracted as gradient magnitudes in 9-histogram bins in each of the four cells of the retina were concatenated and normalised as an input vector to the neural controller (Fig. 6.8).

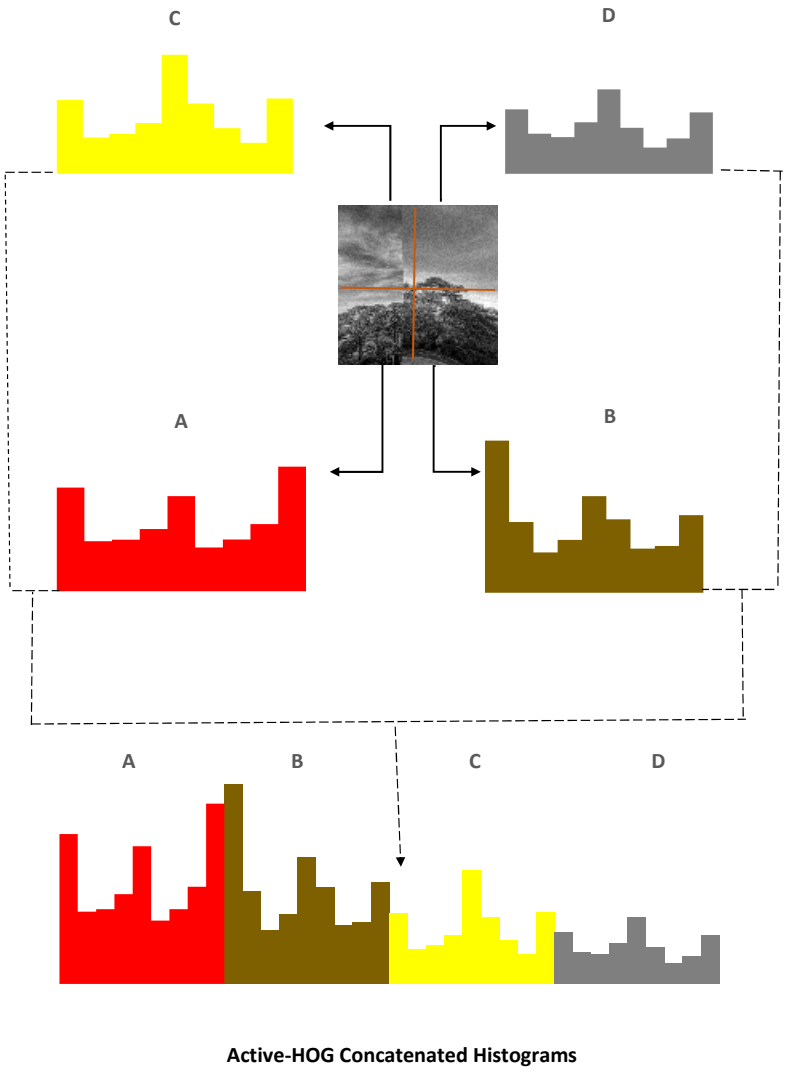

FIGURE 6.8: Shows the concatenated Active-HOG histogram features that were normalised and input into the neural network at time step t.

6.3 Results

In this section, we present the results and comparative analysis of the categorisation process for the three methods of visual extraction. The capability of the iCub agent

to correctly classify the category of an environment (indoor or outdoor) was assessed by the percentage of times in the second half of each trial that the categorisation unit corresponding to the current environment where the agent was situated, was the most activated.

6.3.1 Evolution

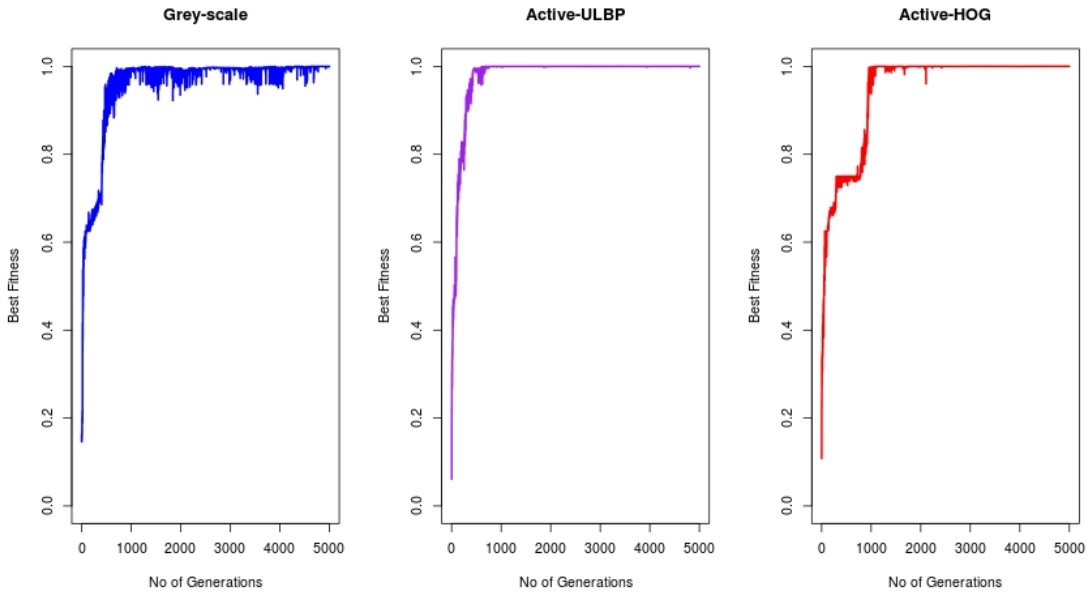

FIGURE 6.9: The best-fitness graphs of the best evolutionary runs of the three methods of visual extractions. **Left**: The best run of the grey-scale averaging method. **Middle**: The best run of the Active-ULBP method. **Right**: The best run of the Active-HOG method.

In the evolution of the active vision system, we performed 12 evolutionary runs for each of the visual extraction techniques (as shown in Appendix C, Fig. C.1, Fig. C.2 and Fig. C.3). The first 6 runs were for the first fold of the 2-fold cross validation, while the remaining 6 runs were for the second fold. Each evolutionary run had 5000 generations, with each genotype evaluated for 20 trials and 100 time steps in a trial. Fig. 6.9 shows the best fitness graphs of the best runs of the three visual extraction methods. Looking at the graphs for the three methods of the visual extraction, one can observe a common fitness pattern in which fitness growth reached close to the optimal value of 1.0 at the early stage of the evolution around 1000 generations. However, one will also notice that Active-ULBP and Active-HOG were more stable over the last generations than the grey-scale. Also, Fig. 6.10 shows the average (mean) of the best fitness in all generations of all evolutionary runs and their positive and negative standard deviation from the mean. The mean of the best fitness in all generations of all evolutionary runs for the three visual extraction methods had a common trend of close approximation to 1.0 from about 1000

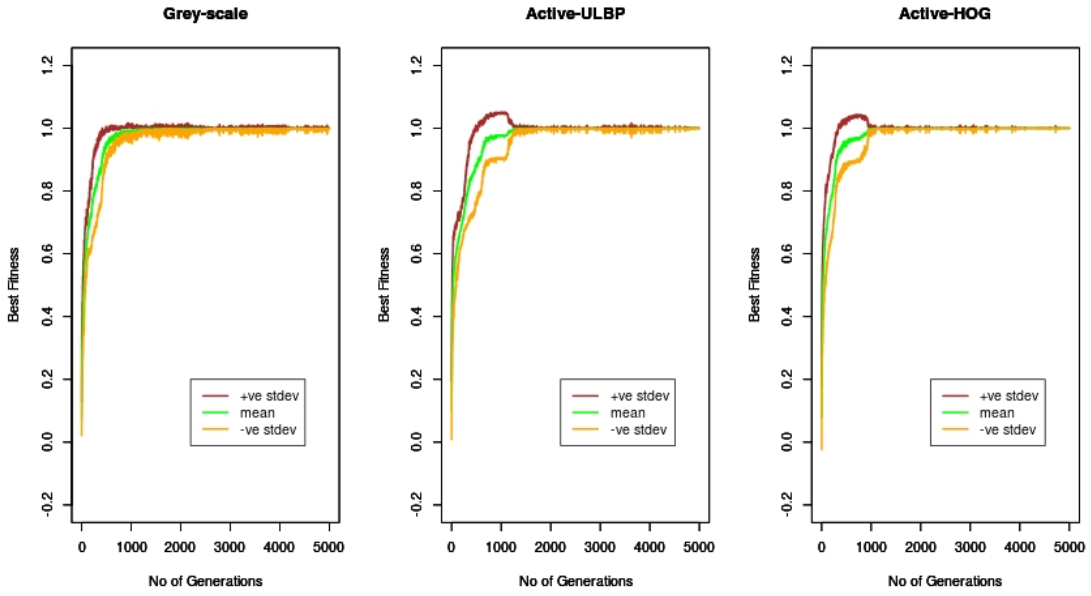

FIGURE 6.10: Shows the graph of the mean (average) of all best fitness in each generation of the 5000 generations for 12 evolutionary runs and their positive (+ve stdev) and negative (-ve stdev) standard deviation in each generation for the three methods of visual extraction.

generations onwards. However, looking at the standard deviation, the deviation from the mean seems to be more obvious from about 300 to 1000 generations for all three methods, with Active-ULBP having a larger deviation than the other two methods. The Active-HOG also had larger deviation within this generational period than the grey-scale.

The early optimal solutions of the three visual extraction methods as reflected in the training may be due to the small number of images that were used in order to reduce the time complexity of the evolutionary method. Therefore, the system might have formulated easy solutions to these problem because of the small number of images that were used, and also small number of trials that were performed. It probably developed some strategies of detecting a particular cue common to these environments (images), and the apparent close to optimal classification performance might have been by chance. Therefore, the importance of re-evaluation (testing) is to test the robustness of the model by introducing more variability into the system, for example: changing the initial position of the eye in each trial, rotations of the environment/stimuli and increasing the number of trials. This is not possible in the evolutionary runs because of computational cost. For this reason, the complexity of the problem was in the generalisation of the skills learned by the evolved genotypes to unseen images coupled with the additional variability and trials introduced in the testing. This is shown in the next section on categorisation performance.

6.3.2 Categorisation Performance

As was the case in the 3D object categorisation experiment in Chapter 5, we assessed the performance of the system using the best evolved genotypes of 100 consecutive generations that had a relatively higher and more stable fitness pattern as compared to the other generations in all evolutionary runs. This is unlike the 2D iCub-image categorisation experiment described in Chapter 4, where we took a more regular approach by re-evaluating the best genotypes of the last one thousand generations. We reduced the chosen number of genotypes for re-evaluation in order to keep the re-evaluation time within reasonable limits, considering the high computational costs of the 3D experiments. We also did not restrict ourselves to the re-evaluation of the genotypes of the last 100 generations, since in several runs these solutions turned out not to be among the most successful when compared to solutions arising from other evolutionary times. The 100 genotypes for each of the 12 evolutionary runs were tested on 10 unseen texture images mapped to a 3D sphere as a representation of indoor or outdoor environments. Also, the conditions set in identical fashion to the training, i.e. in each trial the eye was initialised in each quadrant of the iCub gaze-space, but randomly located in each initialisation within a quadrant, and the environment was randomly rotated in the range $[-40°, 40°]$ with a uniform distribution on the z axis.

A total of 200 trials were performed, i.e. in each trial the iCub agent was evaluated in each unseen indoor or outdoor environment with different initial random eye positions and the environment randomly rotated. We assessed the categorisation performance in the second half of each trial. The categorisation performance assessment of the active vision system was based on the percentage of times in which the categorisation unit corresponding to the current category in the second half of the trials was the most activated. Tables 6.1, 6.2, 6.3 shows the confusion matrices of the categorisation performance of the best performing re-evaluated genotypes from all the evolutionary runs for the three visual extraction methods. It can be seen from the tables that even though the correct categories had the highest average categorisation performance in all trials of the categorisation tasks, the performance were not close to optimum as was reflected in the evolution stage. However, Active-HOG still had a performance close to the optimum level and Active-ULBP also performed better than the grey-scale.

The summary of performance of the re-evaluated best genotypes from all evolutionary runs for the three methods of visual extractions are also shown in Table 6.4. The metrics used are as follows: **Max** represents the best performance from all re-evaluated genotypes in all runs; **Average** is the average of the best performance in each run; **Worst** is the worst of the best performances in each run; and **stdev** is the standard deviation of the best performance of all runs. From the table, one can see that Active-HOG had

the overall best performance of 99.15% as compared to those of Active-ULBP (91.48%) and grey-scale (88.31%). Active-HOG, also had highest average performance of 85.39% as compared to those of Active-ULBP (75.17%) and grey-scale (69.82%). Furthermore, Active-HOG had the best worst performance of 70.34% as compared to those of Active-ULBP (54.78%) and grey-scale (58.55%). However, the standard deviation values show that the distribution of performance for these three visual extraction methods had a similar pattern. In general, the average performance of all three methods shows that they performed well; however their performance was not close to the optimum in the testing stage as was reflected in the evolution stage (Fig. 6.11).

The difficulty encountered in the generalisation of the skills learned in the training to the testing data-set may be due to the following reasons: (i) even though the data-sets used were not many, it is very difficult to generalise the discriminatory labelling of environments as either indoor or outdoor because of the huge variability in these kinds of environment such as texture and structures; and (ii) the random rotation of the environment in each trial, coupled with the large number of trials that were performed.

In the next section, we discuss the statistical significance results of the three methods of visual extraction during the testing stage.

TABLE 6.1: The average performance of the best performing re-evaluated genotype of **grey-scale averaging** in all trials of the testing stage.

	Percentage of Correct Categorisation (Highest in Bold)	
Current category	**outdoor**	**indoor**
outdoor	**81.15**	18.84
indoor	4.55	**95.45**

TABLE 6.2: The average performance of the best performing re-evaluated genotype of **Active-ULBP** in all trials of the testing stage.

	Percentage of Correct Categorisation (Highest in Bold)	
Current category	**outdoor**	**indoor**
outdoor	**91.13**	8.87
indoor	8.17	**91.83**

TABLE 6.3: The average performance of the best performing re-evaluated genotype of **Active-HOG** in all trials of the testing stage.

	Percentage of Correct Categorisation (Highest in Bold)	
Current category	**outdoor**	**indoor**
outdoor	**98.62**	1.38
indoor	0.31	**99.69**

TABLE 6.4: Shows the summary of the statistics of the best performing re-evaluated genotypes in all runs for each visual extraction methods.

Visual extraction methods	Max	Average	Worst	Stdev
Grey-scale averaging	88.31	69.82	58.55	±9.74
Active-ULBP	91.48	75.17	54.78	±11.23
Active-HOG	99.15	85.39	70.34	±9.74

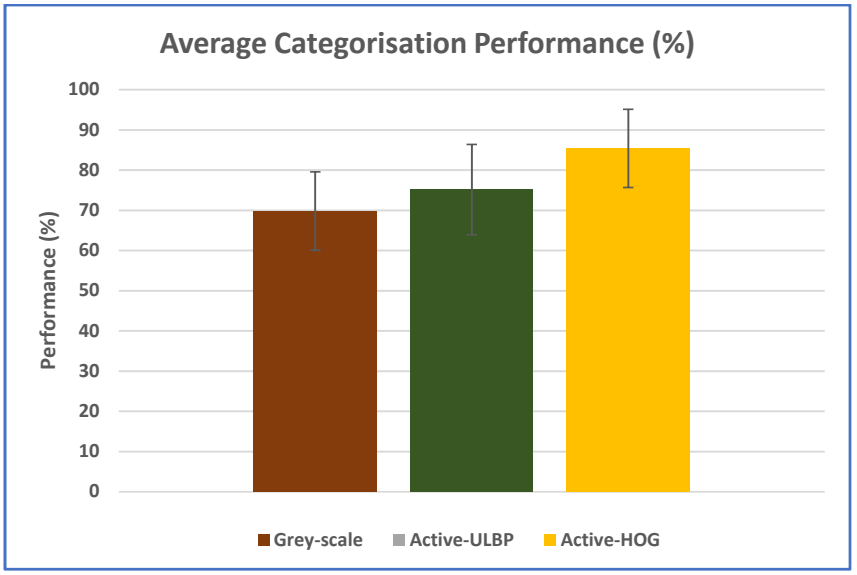

FIGURE 6.11: Bar-chart showing the average categorisation performance of the three methods of visual extraction in all runs.

Statistical Analysis

We tested if the averages of the three visual extraction methods were significantly different with an extended version of t-test, as we have done in the 2D and 3D objects categorisation experiments. We tested the significance of the differences of the averages with the (p-value<0.05) and a more stringent (p-value<0.01). The statistical summary of the visual extraction methods used to calculate results of the **anova** test are shown in Table 6.5 where: the **Visual extraction methods** column indicate the methods of visual extraction; the **Count** represent the number of evolutionary runs for each method; the **Sum** indicates the sum of the individual performances of the best performing genotypes of the three methods; and the **Average** and **Variance** indicate the averages and variance of the performance of the best performing genotypes of all runs for the three visual extraction techniques.

Likewise, for the ANOVA test, the columns of Table 6.6 are: **Source of variations** indicates the source of variations between and within the groups for which averages

were compared (i.e grey-scale averaging, Active-ULBP, Active-HOG); **SS** represents the sum of squares; **df** represents the degree of freedom; **MS** represents groups mean square; **F** is the F distribution value; **P-value** indicates the significance level of the difference in averages considered (i.e. for the three methods of visual extraction); and **F crit** represents the F critical value. The obtained p-value of 0.0027 in the table is less than the two significance levels of 0.05 and 0.01, and this indicates strong evidence against the null hypothesis that the averages for the three visual extraction methods were equal and therefore we reject the null hypothesis. We then carried a Bonferroni correction for the two significance levels of 0.05 and 0.01 to ensure that the overall significance level does not exceed these two values as the significance level of each individual t-test to be carried out. The obtained Bonferroni corrected p-values for 0.05 and 0.01 significance levels (i.e. 0.0167 and 0.003 respectively) were then used as the new significance levels for the results of a paired t-test comparison of the three visual extraction methods (Table 6.7). In the table, the first column indicates the paired groups that were compared, the second and third columns indicate the t-values and p-values of the means (averages) comparisons, while the fourth and fifth column indicate the significance levels based on the Bonferroni corrected p-values.

TABLE 6.5: Summary of the statistics of the best performing re-evaluated genotypes of the three visual extraction methods from 12 evolutionary runs that were used in the anova test.

SUMMARY				
Visual extraction methods	***Count***	***Sum***	***Average***	***Variance***
Grey-scale averaging	12	837.83	69.82	94.92
Active-ULBP	12	902.07	75.17	126.08
Active-HOG	12	1024.70	85.39	94.95

TABLE 6.6: The results of the anova test.

ANOVA						
Source of Variation	***SS***	***df***	***MS***	***F***	***P-value***	***F crit***
Between Groups	1502.35	2	751.17	7.13	0.0027	3.29
Within Groups	3475.31	33	105.31			
Total	4977.66	35				

TABLE 6.7: The significance test result using paired t-test with test conditions of (p-value<0.05) and (p-value<0.01).

			Signf. Level=0.05	***Signf. Level=0.01***
Compared Groups	***t-value***	***p-value***	***Bonf. Corr=0.0167***	***Bonf. Corr=0.003***
Active-HOG and Grey-scale	3.72	0.0010	Significant	Significant
Active-HOG and Active-ULBP	2.44	0.0237	Not Significant	Not Significant
Grey-scale and Active-ULBP	1.28	0.1742	Not Significant	Not Significant

Comparing the three groups in the table at the 0.05 significance level with a Bonferroni correction of 0.0167, the variation in averages of Active-HOG and grey-scale was statistically significant, while those of Active-HOG and Active-ULBP, and Active-ULBP and grey-scale were not statistically significant, which means that the resulting differences in their averages could have been by chance. Therefore, for the significance level of 0.05, we reject the null hypothesis that the averages of the groups Active-HOG and grey-scale were equal, while we fail to reject the null hypothesis for the other groups (i.e. Active-HOG and Active-ULBP, and Active-ULBP and grey-scale).

On the other hand for the strongly significant level of 0.01, the variation in averages of Active-HOG and grey-scale was statistically significant, while those of Active-HOG and Active-ULBP, and Active-ULBP and grey-scale were not statistically significant, that means that the resultant differences in their averages could have been by chance. Therefore, for the significance level of 0.01, we reject the null hypothesis that the averages of the groups Active-HOG and grey-scale were equal, while we fail to reject the null hypothesis for the other groups (i.e. Active-HOG and Active-ULBP, and Active-ULBP and grey-scale).

6.3.3 Dynamics of Categorisation Process

This section investigates the categorisation process in the 3D indoor and outdoor environment. In particular, we examine:

(i) To what extent the sensory patterns provided by the visual extraction methods and experienced by the agent during interaction with the indoor and outdoor environments have been able to provide the discriminative stimuli that facilitated the categorisation process.

(ii) To what extent the agent self-selection has succeeded in associating stimuli with a particular category.

Note: stimulus ambiguity may depend on the nature of the stimulus, the field of view of the iCub eye and the eye location.

The classification answers provided in the output units of our system are dependent on the visual information that was provided and the copy of the outputs of the categorisation and motor units at the previous time. However, since our focus is mainly on the influence of visual stimuli on control of the active vision in order to improve learning for categorisation, we only investigate the visual sensory channel. In order to do this investigation, we extend the Modified Geometric Separability Index (MGSI) proposed

in [23] and used in Chapters 4 and 5 to the indoor and outdoor environment categorisation. The MGSI of the best performing re-evaluated evolved genotypes of the three visual extraction methods of all evolutionary runs was computed for 200 trials during which the agent experienced 10 different indoor and outdoor environments, with each environment uniformly and randomly rotated within the range $[-40°, 40°]$ to the original orientation with 20 different initial eye positions. For each type of visual extraction method of the sensory patterns, the MGSI had been computed for each of the 100 time steps (Fig. 6.12, Fig. 6.13, Fig. 6.14).

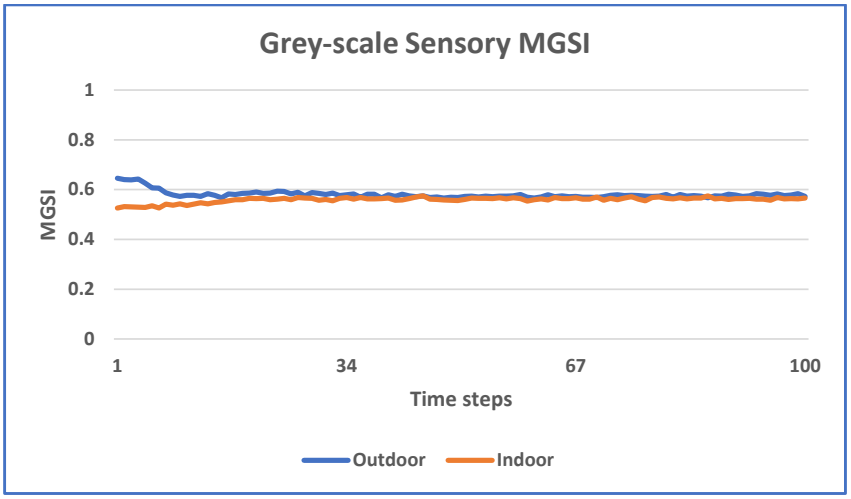

FIGURE 6.12: Modified Geometric Separability Index (MGSI) of the stimuli provided by grey-scale averaging.

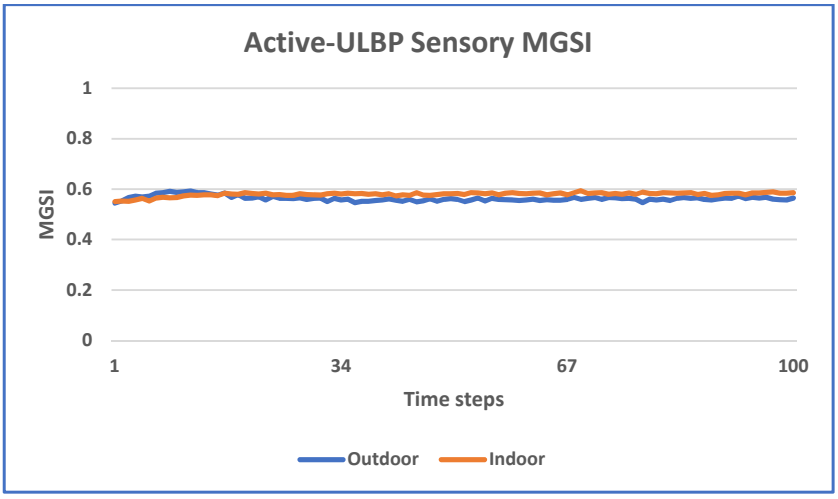

FIGURE 6.13: Modified Geometric Separability Index (MGSI) of the stimuli provided by the Active-ULBP method.

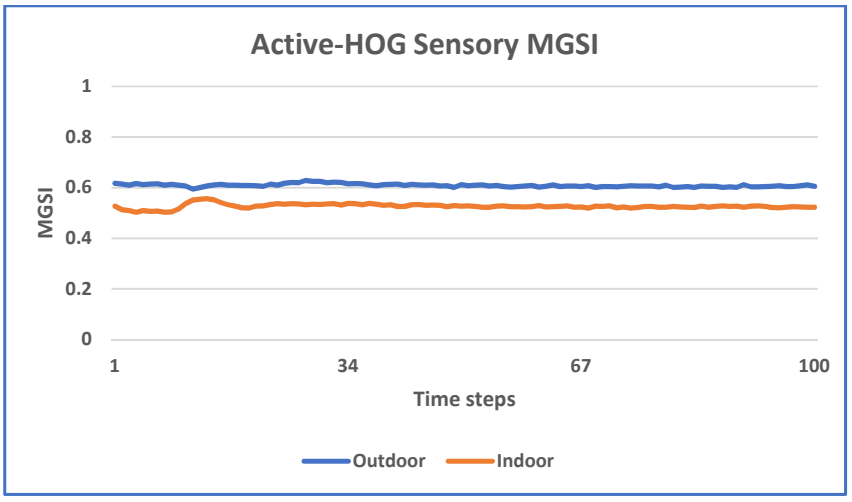

FIGURE 6.14: Modified Geometric Separability Index (MGSI) of the stimuli provided by the Active-HOG method.

The fact that the MGSI did not show much improvement either for all conditions (visual extraction methods) or the two environments (indoor and outdoor) showed that the system did not make much use of intelligent motor control in order to disambiguate the ambiguous visual information. This actually was not a problem given the performance of the three visual extraction techniques. The system must have relied heavily on the internal states of the controller for the integration of sequences of experienced sensory states over time.

6.4 Discussion

We have extended the evolutionary active vision system with pre-processing to 3D for indoor and outdoor environment categorisation using the iCub robot platform. This extension is important because it has a different problem structure with greater need for exploration.

We have used just 20 texture images to represent the indoor and outdoor environments because of the high computational costs incurred by the evolutionary method in a 3D context. However, we tried to compensate for the small data-set used with random rotation of the environment in different trials of the evolutionary runs; in this way, the agent would always see different views of the environment in different trials. However, the system seems to have found easy solutions to the problem as was evidenced by the early attainment of optimum fitness in the evolutionary runs. The complexity of the problem, however, was in the generalisation of the skills learned by the system in the training to the new set of environments (images) in re-evaluation (testing) stage,

with more variability introduced, such as the number of trials coupled with environment rotations and different initial eye positions of the iCub agent. This consequently lowered the performance as compared to what was reflected by the system (grey-scale, Active-ULBP and Active-HOG) in the evolutionary (training) stage, of which many trials were not possible because of the computational cost.

We further continue our discussion in two areas: (i) visual representation and active vision categorisation; and (ii) learning control of the active vision system.

Visual Representation and Active Vision Categorisation

We extended three visual extraction methods i.e. (i) grey-scale averaging [23], Uniform Local Binary Patterns [1] (as Active-ULBP) and Histogram of Oriented Gradients [2] (as Active-HOG), as visual representation of the active vision system for the environment categorisation problem. The average performance of Active-HOG was higher than the other two visual extraction methods, while Active-ULBP also outperformed on average the grey-scale method. However, statistical analysis results showed that the average performance of Active-HOG was only significantly better than that of the grey-scale. The Active-ULBP was also not significantly better than the grey-scale. This implies that the higher performance of Active-HOG relative to Active-ULBP and the better performance of Active-ULBP relative to grey-scale may have occurred by chance. The improvement shown by Active-ULBP in the environment categorisation problem may be due to the fact that ULBP is a good feature descriptor for detecting local binary texture patterns in texture images [45]. HOG may also work well for texture images, especially if there are a lot of structures in the images. Overall, the fact that the two pre-processing methods investigated (i.e ULBP and HOG) evinced good performance in the 3D indoor-outdoor environment categorisation shows the potential of these kinds of visual extraction methods as effective visual representation methods in active vision systems.

Learning Control of the Active Vision System

We have investigated the extent to which an active vision system has been able to use its intelligent control to detect the regularities that are peculiar to each environment. As we have seen for the three visual extraction methods, the MGSI for the two environments (indoor and outdoor) did not increase over time. The inability of the system to use intelligent sensory-motor coordination to experience regularities that are unique to the different environments may be due to the complexity of their visual stimuli, where it may be difficult for the agent to fully separate the unique stimuli that pertain to these two

environments in the input space. It is difficult to generalise the kind of features that make up indoor and outdoor environments, and the agent may in most cases experience similar stimuli for the two different environments during its interaction with them. Therefore, this may explain why there was not much improvement in the MGSI over-time. However, since the three visual extraction methods still performed well, the active vision system must have relied heavily on the integration of the sensory patterns over time by the internal dynamics of the controller.

6.5 Chapter Summary

In this chapter, we have extended the evolutionary active vision system for 3D indoor and outdoor environments categorisation using the grey-scale averaging visual representation method [23]. The best performance for the grey-scale averaging method was 88.31% and the average performance was 69.82%.

We also used the two pre-processing techniques in computer vision, i.e. Uniform Local Binary Patterns [1] (as Active-ULBP) and Histogram of Oriented Gradients [2] (as Active-HOG) in order to improve the performance of the active vision system. The best and average performance of Active-ULBP were 91.48% and 75.17% respectively, while those of the Active-HOG were 99.15% and 85.39%.

Statistical analysis investigation that compared the average performance of the three methods, shows that the performance of Active-HOG was only significantly better than that of grey-scale. Also, Active-ULBP was not significantly better than the grey-scale.

Analysis based on a Modified Geometric Separability Index (MGSI) shows that the categorisation tasks must have been dependant on the integration of the perceptual information over time, since the visual stimuli belonging to the indoor and outdoor environment were not clearly separated in the input space.

Chapter 7

Discussion and Conclusion

7.1 Introduction

This thesis investigated an evolutionary method of control of active vision for learning in categorisation. We tried to impose minimal assumptions on the active vision system in order to freely develop novel strategies for categorisation through dynamic interaction with the environment. We have therefore chosen an evolutionary method which leaves control of the active vision to the adaptation process of the evolutionary algorithm. We subsequently chose Mirolli et al. [23] as our benchmark architecture because of the following inherent properties of the system: (i) the biological plausibility of using a neural network as a controller,;(ii) the architecture is able to combine control with classification; (iii) the complexity of the categorisation task to be performed as compared to previous evolutionary active vision systems; and (iv) the system's inherent sensory-motor coordination property, and the ability to integrate sensory-motor information over time, which may be necessary for solving complex categorisation tasks. We sought to improve on their work with pre-processing techniques for visual extraction in the 2D environment, and subsequently extended it to the 3D domain. We demonstrated this in 2D for object categorisation of more complex images taken from the camera of the iCub and with object and environment categorisation in the 3D environment using the iCub platform. This chapter first gives general discussion of the thesis, and then concludes with answers to our research questions, and lists the key contributions of the PhD project. We subsequently outline some drawbacks of our method and possible directions of future research.

7.2 General Discussion

We started our research investigation in the 2D environment with three visual extraction methods i.e. grey-scale averaging [23], Uniform Local Binary Patterns [1] (as Active-ULBP) and Histogram of Oriented Gradients [2] (as Active-HOG). We proposed Active-ULBP and Active-HOG for our benchmark evolutionary active vision system, with a view to improving performance over the currently used grey-scale averaging. Overall, in all replications of the evolutionary run, Active-ULBP evinced the best average performance for 2D. Grey-scale also on average outperformed Active-HOG. The close to optimal performance results we obtained for the three visual extraction methods is highly commendable given the large variability in the data-sets. This is quite interesting, because it demonstrates the potential utility of pre-processing techniques for active vision systems.

Furthermore, we continued our investigation of the active vision system with preprocessing in the 3D environment. We chose the iCub platform because it allowed us to show the plausibility of our methods in complex robotic systems. We intentionally used only one eye with the iCub with fewer degrees of freedom as we felt this was sufficient to demonstrate the robustness of our system in complex categorisation tasks.

In the experiment of object categorisation in 3D, the first challenge was the randomly varied size and orientations in each trial, and the second challenge was the high ambiguity of the stimuli of the objects that were investigated (i.e, sphere, cube, cone, and torus). Despite, the complexity of the problem, the three visual extraction methods that were investigated performed handsomely. Active-HOG boasted the best average performance for 3D, while the grey-scale outshone Active-ULBP in average performance.

On the other hand, the complexity of the indoor and the outdoor environment classification may be due to the following reasons:

(i) In contrast to the object categorisation problem in which categorisation involves one category of object in each trial, environment categorisation can involve many objects within the same environment, which may or may not belong to shared category, and each of which may be in different spatial locations. Apart from this structural information, there is also textural information to be processed.

(ii) The system therefore may have to use the totality of contextual information within each environment to complete the discrimination task, coupled with random rotation in each trial.

In spite of the complexity of the problem, the active vision system also performed well over the course of testing all the visual extraction methods under investigation. Active-HOG evinced the best average performance, while Active-ULBP also achieved a better average performance than grey-scale.

Moreover, due tothe high computational cost of evolutionary system in 3D environments, we only used a small number of training and testing texture images. We tried to compensate for this with the additional variability introduced by random rotation of the environment in each trial, and also with many trials in the testing stage. This was to ensure that the iCub always saw a different view of the environment in each trial. It is very important to state here that we do not claim that our system can discriminate any kind of indoor and outdoor environment of all data-sets. However, we do say that the system was able to discriminate indoor from outdoor environments of the data-sets given based on the contextual information within the environment. Therefore, given more computational resources with more training data-sets, the system has a greater chance of generalising its skills to very large testing data-sets.

7.2.1 Visual representation in active vision categorisation

We have investigated three visual extraction methods, i.e. grey-scale averaging [23], Uniform Local Binary Patterns [1] (as Active-ULBP) and Histogram of Oriented Gradients [2] (as Active-HOG). We proposed Active-ULBP and Active-HOG in order to determine if pre-processing techniques in computer vision can yield better representation for active vision systems for improved performance.

In the 2D object categorisation experiment, Active-ULBP had the best average performance of 96.82% as compared to those of grey-scale (95.77%) and Active-HOG (92.87%). However, a comprehensive statistical analysis test shows that none of the three visual extraction methods performed "highly significantly" better than the others. This implies that the apparent differences in their averages might have arisen by chance.

In the 3D object categorisation experiment, the Active-HOG had the best average performance of 98.07% as compared to an average of 68.53% for Active-ULBP and an average of 74.47% for grey-scale. Also, further statistical analysis shows that the average performance of Active-HOG was both statistically significantly better than that of Active-ULBP and grey-scale, but the grey-scale was not significantly better than the Active-ULBP.

Also, in the 3D indoor-outdoor environment categorisation, Active-HOG showed the best average performance of 85.39% as compared to those of Active-ULBP (75.17%)

and grey-scale (69.82%). However, with this experiment, the results of a statistical analysis test shows that the Active-HOG performed only significantly better than the grey-scale. The average performances of Active-ULBP and grey-scale did not differ to any statistically significant degree.

The improvement in performance of Active-HOG in the 3D object categorisation may be due to the more structural nature of the object categorisation problem. Equally the good performance of Active-HOG also in indoor-outdoor environment categorisation may have been due to more structural information in the data-sets. Typically in most indoor and outdoor environments, the objects and structures are more conspicuous. For instance, a typical indoor environment may have conspicuous objects, such as tables, chairs, beds, and so on, while outdoor environments may have structures, such as houses, cars, trees and the like. On the other hand, the fact that Active-ULBP performed well in categorisation tasks irrespective of the environmental context (2D images or 3D indoor-outdoor) is evidence that ULBP is good feature descriptor for detecting local binary uniform patterns in texture images, and a good feature descriptor in many applications [47][214].

Overall, Active-HOG seems to be more robust in performance than the other two visual extraction methods for the following reasons:

(i) Based on the statistical analysis test in the 2D experiment (using p-value$<$0.05 and p-value$<$0.01), none of the visual extraction methods was "highly significantly" better than the others. This implies that the apparent differences in the average performance of the three methods might have been by chance, and given a new or larger data-set, any of the three methods might have performed better than the others.

(ii) However, since, Active-HOG performed better than the other two visual extraction methods in the 3D object categorisation, and the grey-scale in the indoor-outdoor environment classification, Active-HOG may have greater chance of achieving better results given new data sets in both 2D and 3D environments.

7.2.2 Learning for control in active vision categorisation performance

The categorisation performance of an active vision system may not depend as much on the complexity of the system design as on the extent to which the agent may use the dynamic interaction of the sensory-motor components to exploit regularities that pertain to the different categories in the sensor input-space. We investigated with the Modified Geometric Separability Index (MGSI) in order to analyse the extent to which the active vision system used its intelligent motor control to experience sensory stimuli

that could be unambiguously associated with a particular category for each of the three visual extraction methods in the input space.

In the 2D environment in particular, the MGSI results showed that all three visual extraction methods generated sensory patterns that allowed the system to move from very ambiguous to less ambiguous stimuli. Active-ULBP also provided less ambiguous stimuli than the other methods. However, grey-scale was a little bit more consistent over time than Active-ULBP.

In the 3D object categorisation, grey-scale was able to use sensory- motor coordination over time to experience more discriminative stimuli than the other two visual representation methods. Active-ULBP also showed some slight use of motor responses in moving to less ambiguous stimuli over time. However, even though Active-HOG generally had less ambiguous stimuli from the start, it was not to a great extent able to use eye movements to experience less ambiguous sensory stimuli. The low ambiguity of Active-HOG in most time steps may be due to the highly structural nature of the problem, and this may also have enhanced its recognition capability. That said, the inability to use sensory-motor coordination to experience less ambiguous stimuli over time, might have been due to the low ambiguity experienced by the system with Active-HOG stimuli from the outset. In this context, there was little need to make use of eye movements to reduce ambiguity over time. The behaviours generated by evolutionary active vision systems are partially determined by the nature of the stimuli that are experienced [105][6]. The oscillatory behaviours produced in most of the time steps were probably strategies the system developed in order to continue to experience highly discriminative features, which in turn led to good performance. However, we are not committed to this view and this may be a subject of future research.

On the whole, in both the 2D and 3D object categorisation, grey-scale used more eye movements than the other two methods to influence the performance of the active vision system. Active-ULBP also evinced more use of eye movements to reduce visual ambiguity in 2D than in 3D and outperformed Active-HOG in both environments.

On the other hand, in both indoor and outdoor environment categorisation experimental contexts, the active vision system seems to have relied heavily on the internal dynamics of the neural network controller. This was because there was only a slight improvement in the MGSI values for the three visual extraction methods over time. Since the performance of the three visual extraction methods was good, the system must have used the internal states to integrate the very ambiguous perceptual information over time. Moreover, the probable reason for the poor learning of the active vision system as compared to the object categorisation experiments may be due to the different context of categorisation. In the object categorisation experiments there was only one object to be

categorised in an image/environment, whereas in the indoor and outdoor environment categorisations there was more variability. For example, there were many structures, each of varying sizes and spatial locations. There were also other variables such as texture, and some of the variables may not be peculiar to a particular environment, which is to say that some structures are common to both indoor and outdoor environments. It may therefore be difficult for the system to discover regularities that are particular to an environment (indoor or outdoor) through dynamic sensory-motor interaction alone.

7.3 Conclusion

We started our work in the 2D environment using the Mirolli et al. [23] architecture as our benchmark, and for more complex images taken from the camera of a iCub robot. We further extended the model to the 3D environment using the iCub platform for object and indoor-outdoor environment categorisation tasks. Analysis based on the MGSI showed that our active vision system using grey-scale averaging visual representation was able to use a good deal of intelligent control of eye movements in solving both 2D and 3D object categorisation tasks. However, in the environment classification tasks, it seems to have relied more on the internal states of the system for the integration of perceptual information over time. By contrast, the pre-processing methods have been able to learn to control eye movements mainly in the 2D categorisation tasks, while only using a small degree of learning in the 3D object categorisation task with Active-ULBP. They also seem to rely mainly on the internal states of system in the environment categorisation tasks.

In general, the system was able to solve the categorisation problems through the dynamic interaction of sensory-motor components, and/or integration of perceptual information over time through the internal dynamics of the neural network controller. It should be noted that other analyses can be performed apart from the Modified Geometric Separability Index (MGSI) to understand more of the categorisation process. However, the focus of this PhD research is mainly on learning control of active vision for categorisation performance and not on underlying phenomena beyond the categorisation process. We only performed the MGSI to investigate the extent to which the sensory patterns of the different visual extraction methods contributed to learning, for performance in categorisation, given the strong coupling between perception and motor responses. Here, we re-visit our research questions and key contributions.

The research questions for this thesis are:

(i) Do evolutionary active vision systems for categorisation work in more complex scenes and environments?

It has been shown that evolutionary active vision system can work in complex scenes and environments. This was demonstrated by the extension of our bench-mark evolutionary active vision model (Mirolli et al. [23]) using the three visual extractions methods, i.e. the grey-scale averaging method [23], Uniform Local Binary Patterns [1] (as Active-ULBP) and Histogram of Oriented Gradients [2] (as Active-HOG) in 2D natural images and 3D environments for object categorisation, and indoor-outdoor environment categorisation. (Chapter 4, 5 and 6)

(ii) Can we make them work better with pre-processing techniques in computer vision?

It was shown with the improved performance (average and statistical significance) of Active-HOG over grey-scale in 3D for object and indoor-outdoor environment categorisation that an active vision categorisation performance can be enhanced through pre-processing (chapter 5 and 6).

7.3.1 Key Contributions

The following are the list of key contributions of this PhD research work:

(i) We extended the evolutionary active vision system for object categorisation using more complex (natural) images taken from the camera of the iCub robot. Our bench-mark Mirolli et al. [23], which to the best of our knowledge (in this flavor of active vision) has been used for largest number of categories to date, used hand written images (Chapter 4).

(ii) We extended the evolutionary active vision system for object categorisation in the 3D environment using the humanoid iCub robot platform. To the best of our knowledge no work has been done using evolutionary methods for object categorisation on this platform before (Chapter 5).

(iii) We further extended the evolutionary active vision system for indoor and outdoor environment classification in 3D using the humanoid robot (iCub) platform. To the best of our knowledge no work has been done with any computational model for distinguishing between indoor and outdoor environments on any humanoid robotic platform to date (Chapter 6).

(iv) We extended an active vision system with pre-processing using Uniform Local Binary Patterns [1] (as Active-ULBP) for 2D object categorisation (Chapter 4) and 3D object and environment categorisation (Chapters 5 and 6).

(v) We extended an active vision system with pre-processing using Histogram of Oriented Gradients [2] (as Active-HOG) for object categorisation in both 2D and 3D environments; and indoor-outdoor environment categorisation in 3D (Chapters 4, 5 and 6). We further showed improved performance with pre-processing with Active-HOG in the 3D object and indoor-outdoor environment categorisations over the grey-scale averaging method (Chapters 5 and 6).

7.4 Drawbacks

The evolutionary method approach of evolving controllers for active vision systems has shown promise but at the same time has these following drawbacks:

(i) Training time: the training time using this method may be very lengthy and this may render it impractical for some real life vision problems. This was the case for the environment categorisation problems in Chapter 6, where training with more images may give more generality to the system for testing with previously unseen images. In our case we had to give more variability (e.g. random rotation) to the environment in order to improve its generalisation capability.

(ii) The flexibility granted to determine eye-movement strategy may produce a solution of reduced generality. In the language of machine learning, the model here has a smaller learning bias than existing active vision models. This may result in overfitting, especially if the training set of images is not very large, as noticed in Chapter 6 of our experiment on environment categorisation. The extra degree of freedom given to the active vision system may introduce a greater risk of having a strategy that exploits spurious regularities in the training set of images. This may then result in a case where the system may perform very well in training but not as well in testing. As could be seen in the indoor and outdoor environment categorisation experiment, where the system was close to optimal performance in all the evolutionary runs, but did not perform to the same level in testing (except in the case of Active-HOG).

(iii) Since the system does not search the entire image, it runs a higher risk of missing the pattern of interest than is the case in passive vision systems. Therefore, the great challenge posed by active vision systems is finding intelligent eye movements that will compensate for this loss of general information by discovering regularities that will enhance a particular vision task.

7.5 Future Work

In the section, we discuss five main directions of possible future research. The first two mainly involve further understanding of the behaviour of the current system, while the remaining three are open new areas of research.

The first possible area of future research is to investigate the process of categorisation by the system. For instance, it would be interesting to investigate with an active vision system equipped with a reactive controller, i.e. a controller that does not have any form of internal states or memory. However, a reactive system may not work for the environment categorisation experiment, as was indicated in the MGSI because of its high dependent on controller internal states. Therefore, this future research may focus on the object categorisation experiment, especially 3D object categorisation, mainly to investigate the kind of behaviours exhibited by Active-HOG in the current system. In this reactive experiment all other conditions of the system will remain the same, such as the objects (stimuli) and other variabilities introduced into the system (scales, rotations and so on). An MGSI experiment may now be carried out to see if the behaviours shown by the system are similar to those of the current memory-using system. For instance, if Active-HOG gives similar oscillatory behaviour to the present system, it will be a further indication that the behaviour was a response to Active-HOG stimuli. However, since many transformations take place in the pre-processing methods, as compared to the grey-scale averaging method that uses raw image pixels, it may be difficult to deduce the exact cause of the behaviour.

The second area of research is to fix the eye movement of the present system that uses memory. This may be done at the re-evaluation stage with the best genotypes derived from the three methods of visual extraction (grey-scale, Active-ULBP and Active-HOG), and with similar re-evaluation conditions with the system that uses adaptive eye movements. In this experiment, if the performance still remains at a level comparable to the system that uses autonomous eye movement, it will be a further indication of systemic reliance on the internal states of the controller to complement sensory-motor coordination.

The third possible area of research is to increase the degree of freedoms in the iCub robot experiments. In this research, we have only used the right eye. It would be desirable to also include the left eye. This will give the iCub wider field of view and greater depth of perception in the 3D for the purposes of recognition. Also, additional degrees of freedom, in combination with proprioceptive information such as movement of the neck and head as additional parameters for the neural network controller may help to

resolve visual ambiguities in some 3D objects, where two different 3D objects may have the same 2D experience from a certain view point.

The fourth interesting area of research is to investigate other methods of visual representation for the active vision system. For example, one possible method would be a Gabor filter to extract features within the active window as input into the neural network. Scale and orientation invariant 2D Gabor filters have been widely used to model the behaviour of V1 simple cells as they exhibit similar behaviour to the impulse stimuli [215][216]. It would therefore be desirable to see how this can enhance visual discrimination in artificial systems. Another method of visual representation that can be investigated for an active vision system is a deep convolutional neural network. Convolutional neural networks have been shown to give state of the art performance in many object recognition and categorisation tasks ([217][218]). The model of the convolutional neural network has been inspired by the hierarchical architecture of the visual cortex in primates, in which complex functional responses generated by complex cells are created from more simplistic responses from simple cells. It should be noted to maintain consistency with our philosophy of an active vision system as we have done for the three pre-processing techniques investigated, the visual representation method (Gabor filter or convolutional neural network) would not be used to pre-process the entire image at once. The active vision system would determine the location in the visual scene (image) to be processed and the Gabor filter or convolutional neural network would be used to extract high level features for the neural network controller per time step.

Finally, it would be useful and informative to implement the active vision system in the actual robotic hardware platform in order to see if the system could replicate the same level of performance in the real system. Although, we had tried to simulate the conditions of the real world as much as possible, it is not automatic that the algorithms will perform as well in the real system.

7.6 Publications

The PhD project has yielded several publishable pieces of work, with the following conference papers already published:

(i) Olalekan Lanihun, Bernie Tiddeman, Elio Tuci, and Patricia Shaw. Enhancing active vision system categorisation capability through uniform local binary patterns. In Artificial Life and Intelligent Agents Symposium, pages 31–43. Springer, 2014.

(ii) Olalekan Lanihun, Bernie Tiddeman, Elio Tuci, and Patricia Shaw. Improving active vision system categorisation capability through histogram of oriented gradients. In Conference Towards Autonomous Robotic Systems, pages 143–148. Springer, 2015.

In addition to the two conference papers listed above, there is also a journal paper in preparation for submission.

Appendix A

Experiment 1: 2D Object Categorisation

A.1 Letter Categorisation Experiment

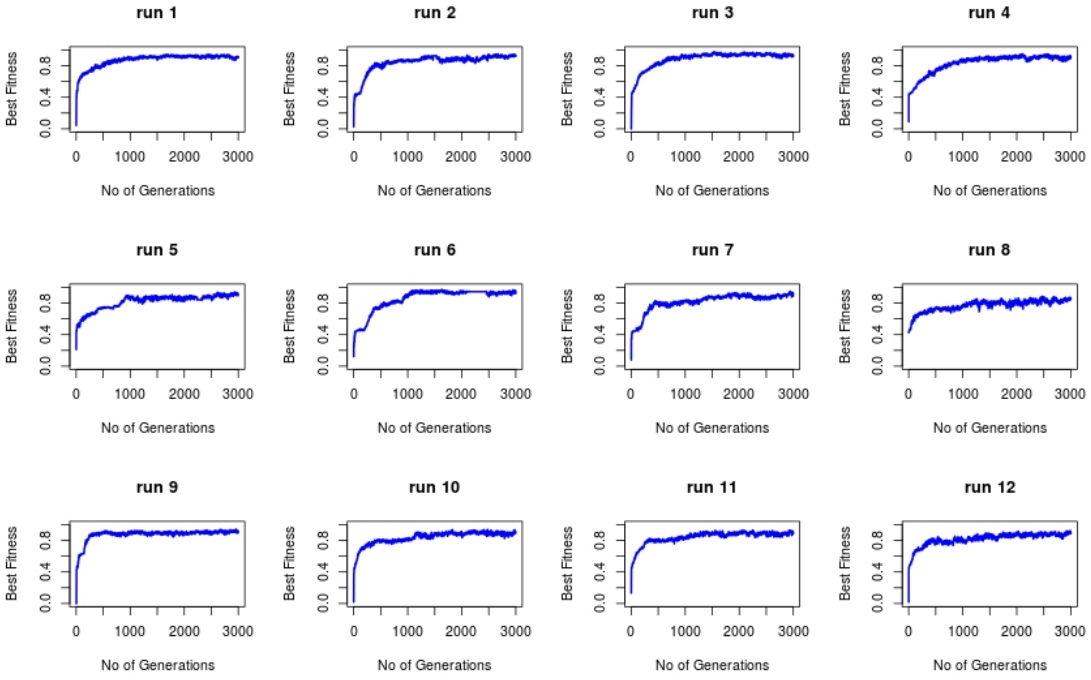

FIGURE A.1: **Grey-scale (Letters)**: The best fitness graphs for all the **evolutionary runs**

A.2 iCub Images Experiment

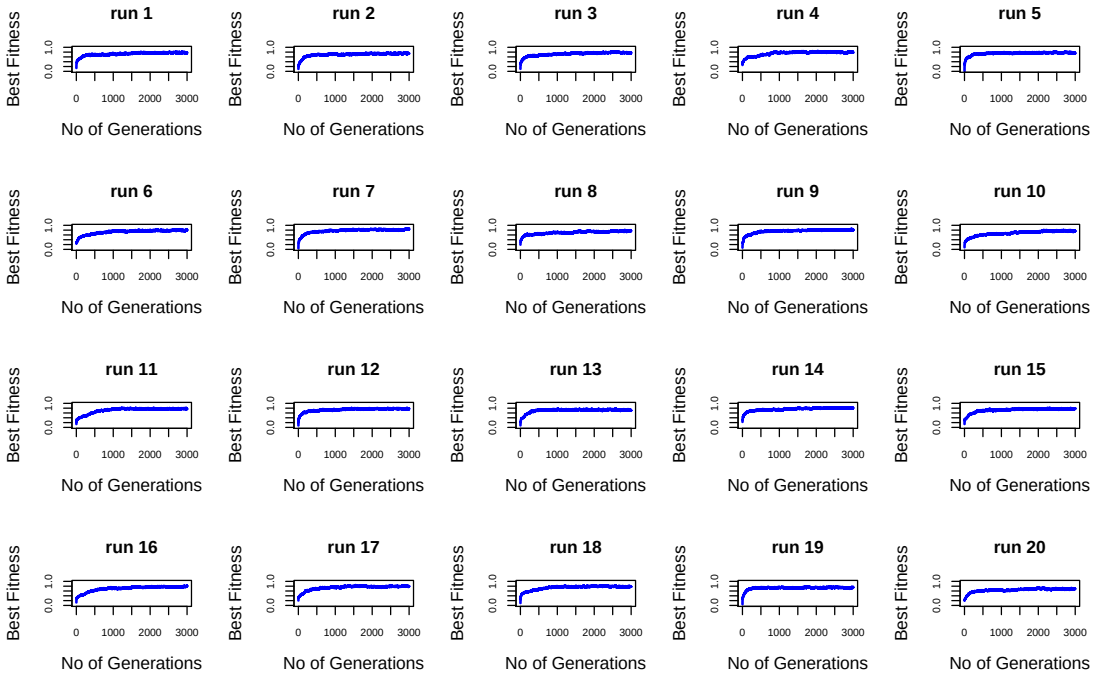

FIGURE A.2: **Grey-scale (iCub images)**:The best fitness graphs for all **evolutionary runs** in the 2-fold cross-validation

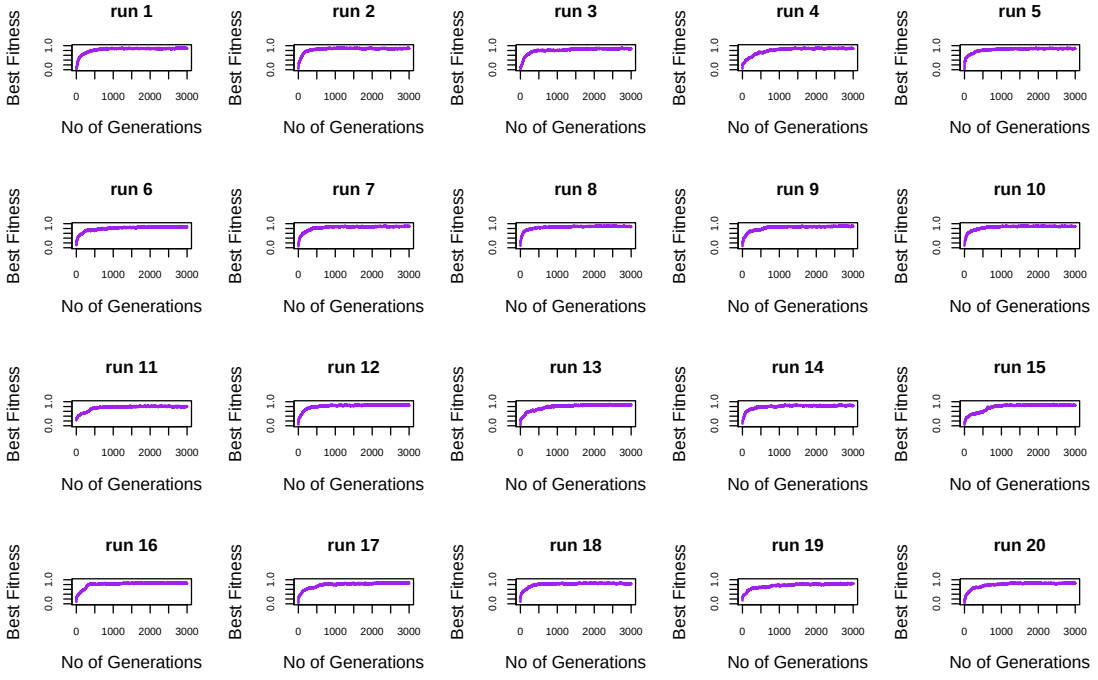

FIGURE A.3: **Active-ULBP**: The best fitness graphs for all **evolutionary runs** in the 2-fold cross-validation

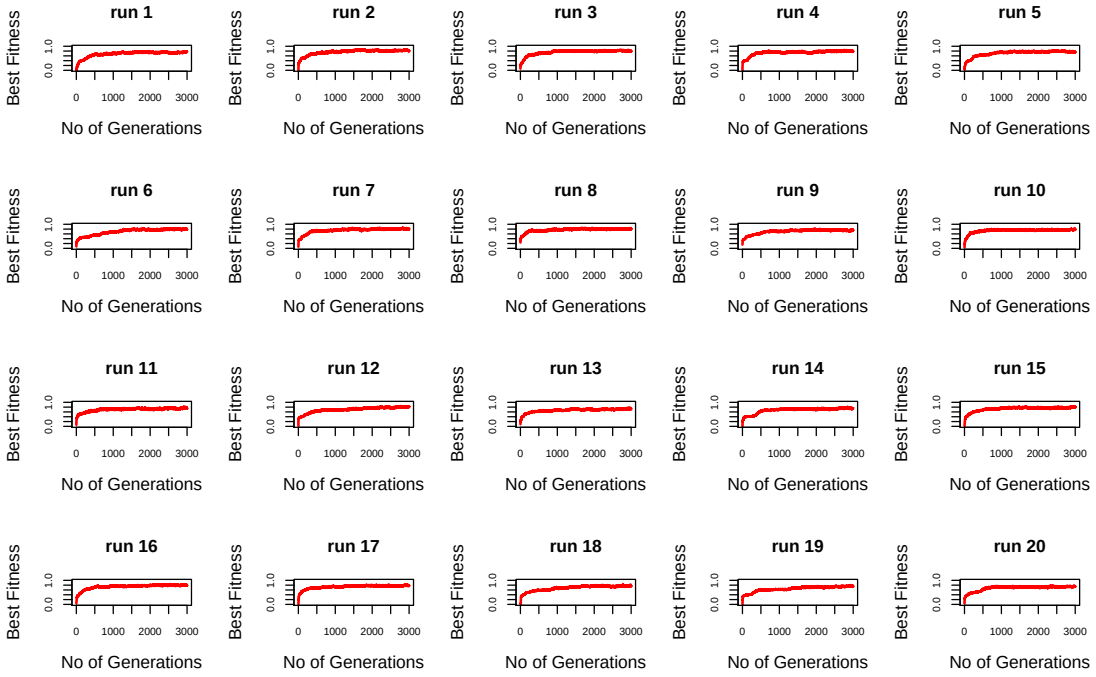

FIGURE A.4: **Active-HOG**: The best fitness graphs for all **evolutionary runs** in the 2-fold cross-validation

Appendix B

Experiment 2: 3D Object Categorisation

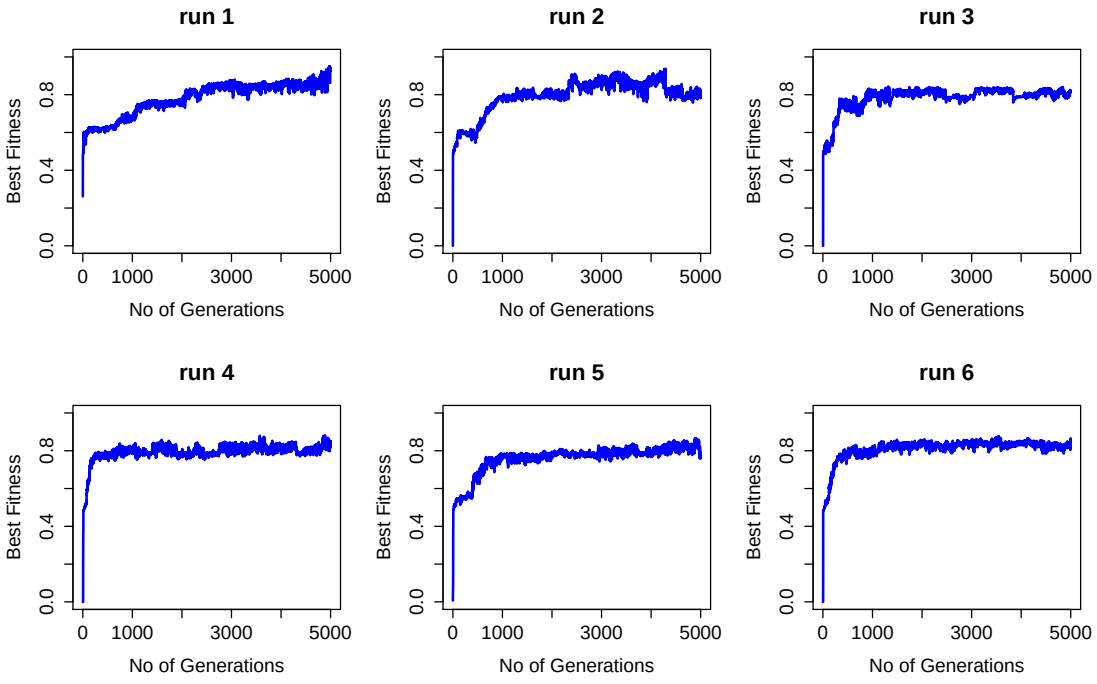

FIGURE B.1: **Grey-scale**: The best-fitness graphs of all **evolutionary runs**.

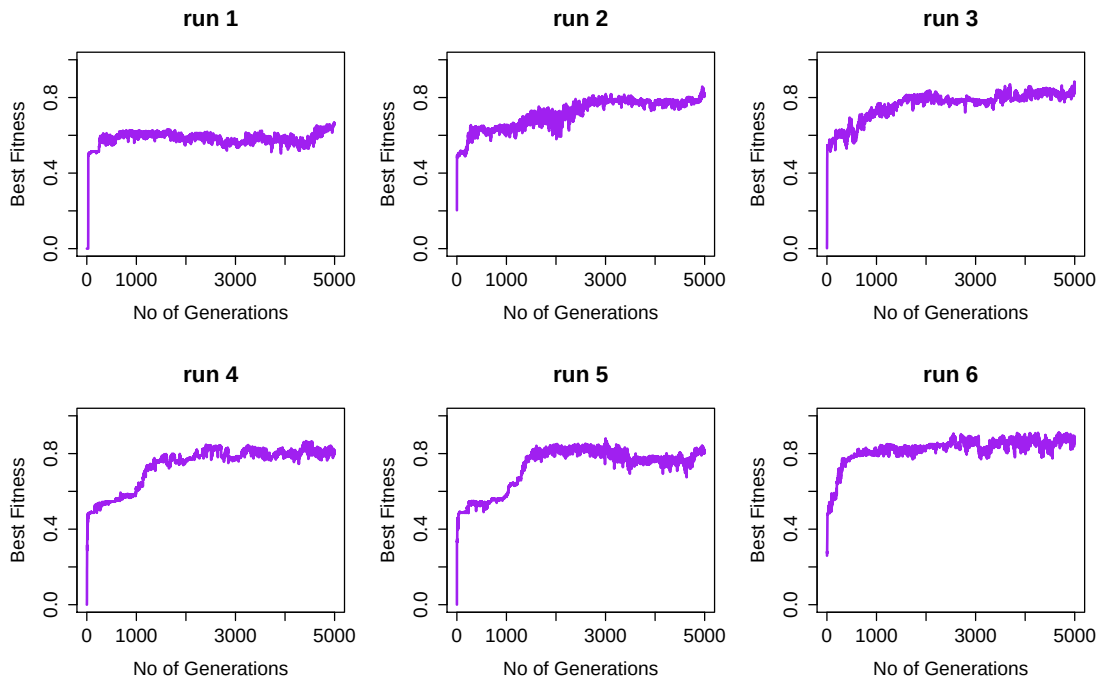

FIGURE B.2: **Active-ULBP**: The best-fitness graphs of all **evolutionary runs**.

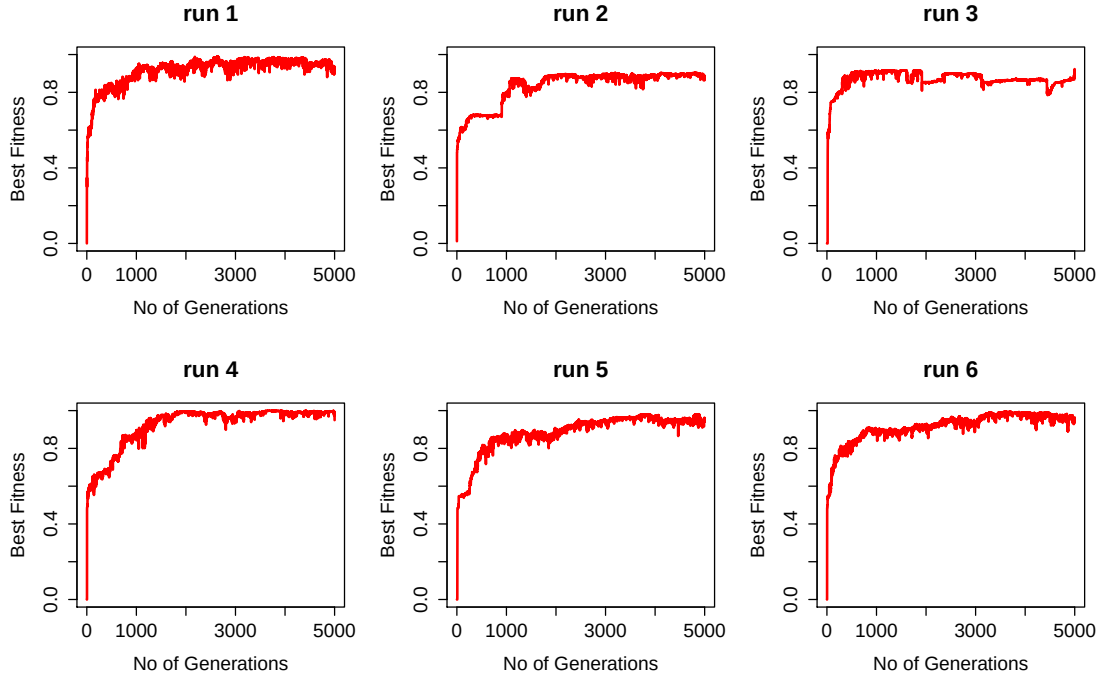

FIGURE B.3: **Active-HOG**: The best-fitness graphs of all **evolutionary runs**.

Appendix C

Experiment 3: 3D Environment Categorisation

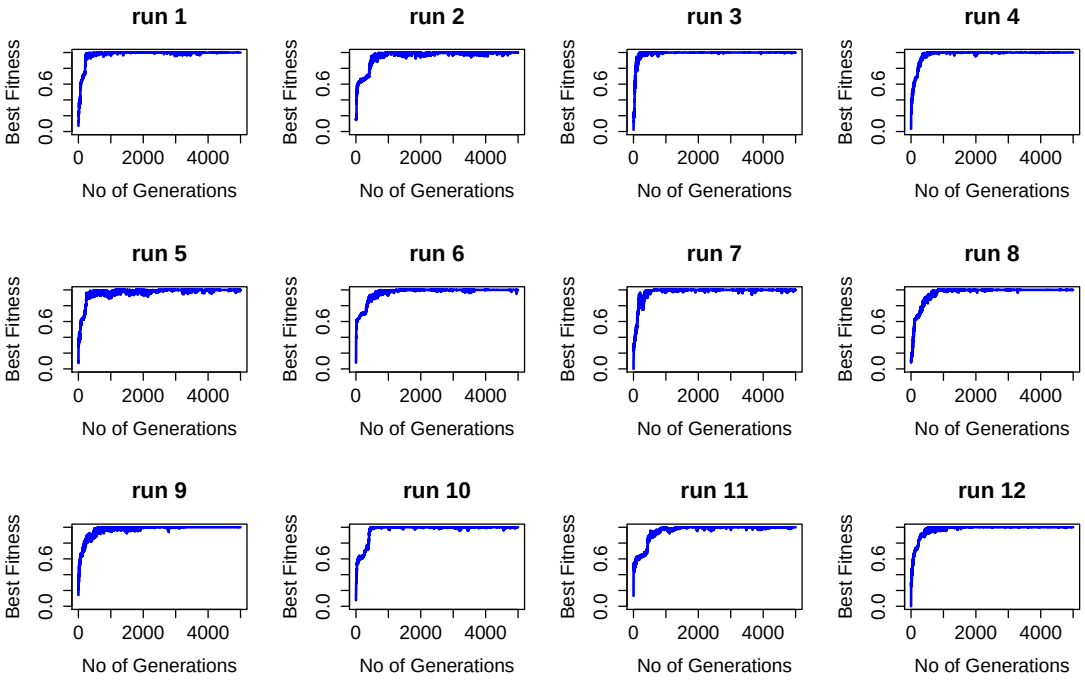

FIGURE C.1: **Grey-scale**: The best-fitness graphs of all **evolutionary runs**.

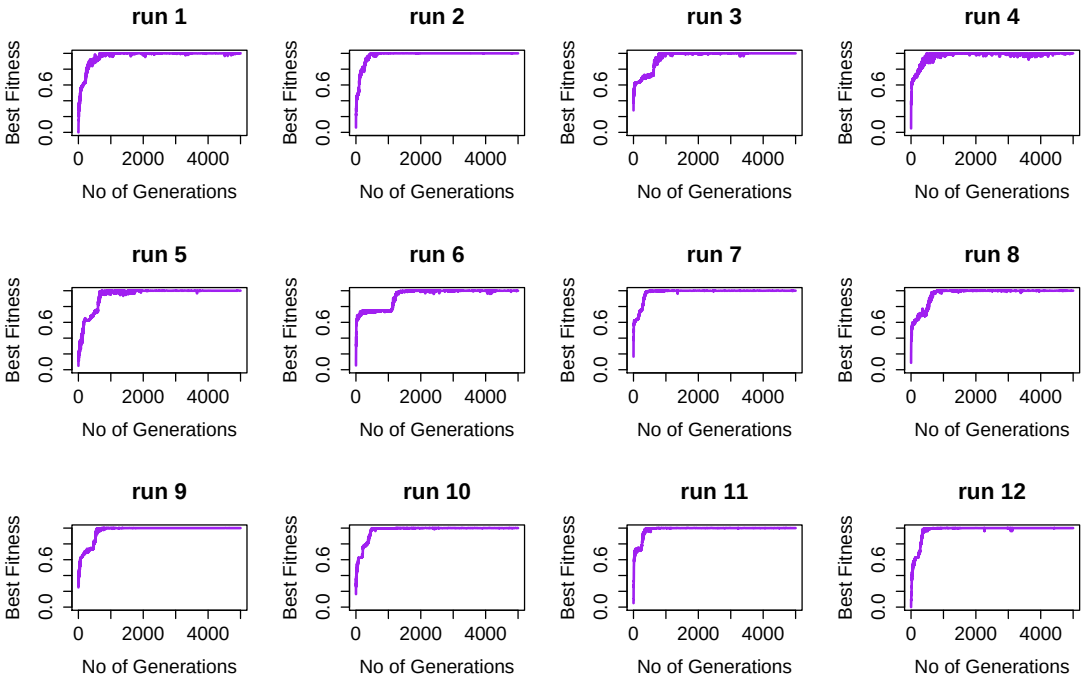

FIGURE C.2: **Active-ULBP**: The best-fitness graphs of all **evolutionary runs**.

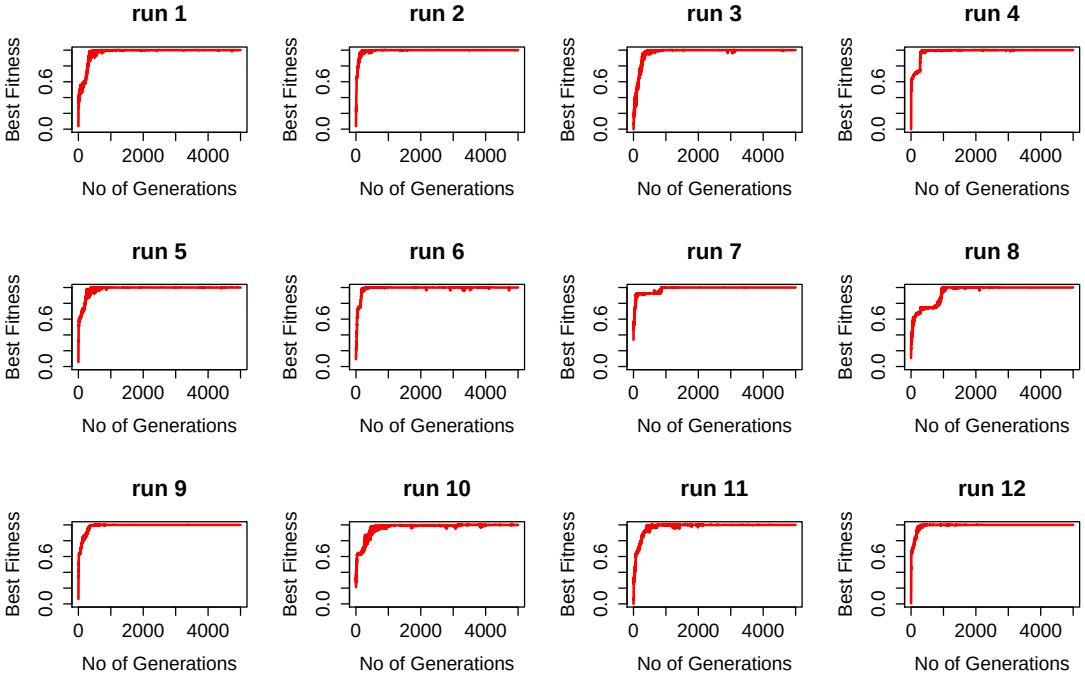

FIGURE C.3: **Active-HOG**: The best-fitness graphs of all **evolutionary runs**.

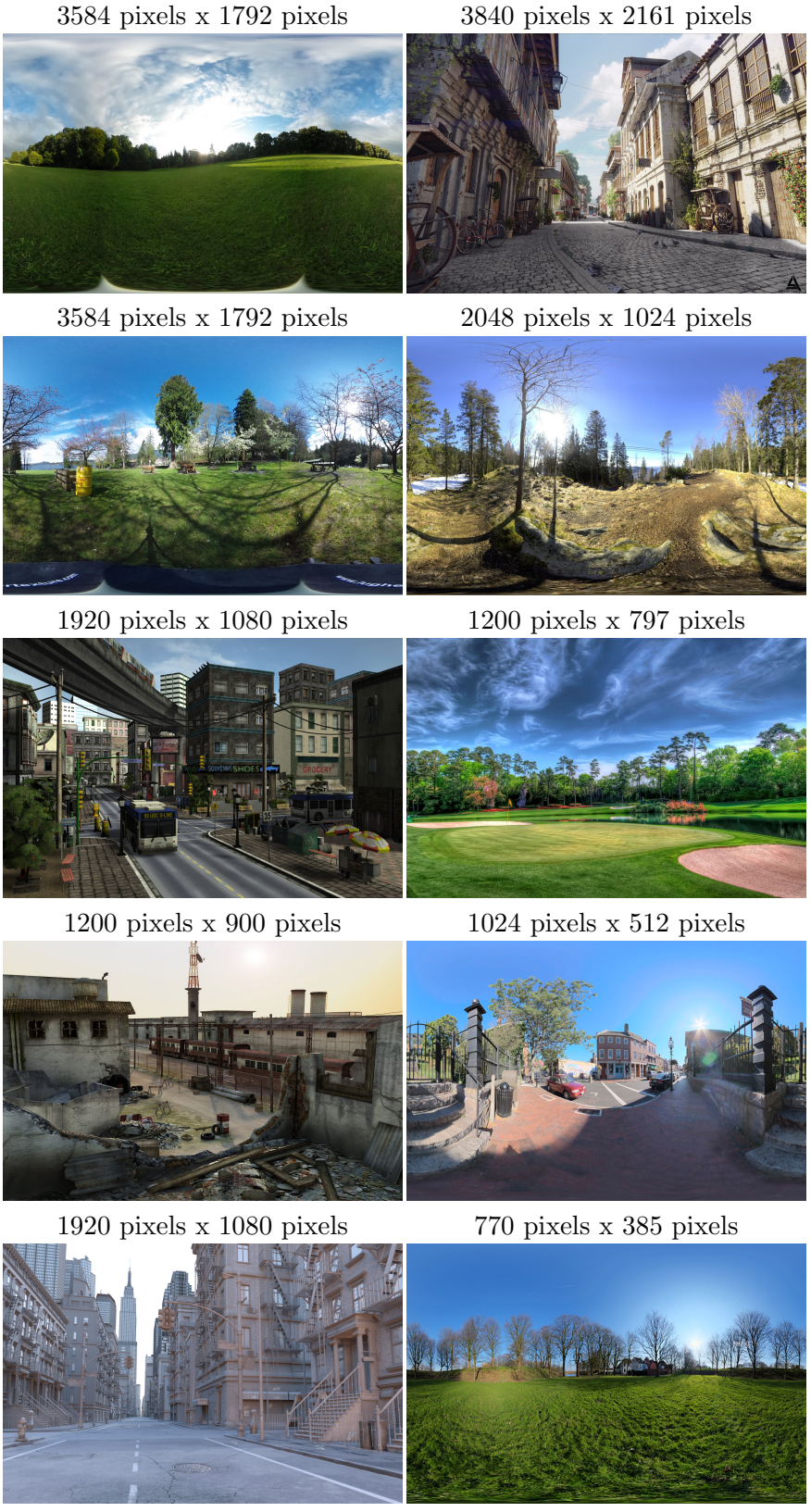

FIGURE C.4: Shows the images of outdoor environments used in Experiment 3 with image sizes in pixels (i.e. width x height)

.

FIGURE C.5: Shows the images of indoor environments used in Experiment 3 with image sizes in pixels (i.e. width x height)

Bibliography

[1] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[2] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[3] Wikipedia. Saccade. https://en.wikipedia.org/wiki/Saccade, 2017. Accessed: 2017-11-09.

[4] Jürgen Leitner, Simon Harding, Pramod Chandrashekhariah, Mikhail Frank, Alexander Förster, Jochen Triesch, and Jürgen Schmidhuber. Learning visual object detection and localisation using icvision. http://www.sciencedirect.com/science/article/pii/S2212683X13000443, 2017. Accessed: 2017-06-28.

[5] Wikipedia. Uv mapping. https://en.wikipedia.org/wiki/UV_mapping, 2017. Accessed: 2017-11-10.

[6] Stefano Nolfi. Categories formation in self-organizing embodied agents. *Handbook of categorization in cognitive science*, pages 869–889, 2005.

[7] Nicola Catenacci Volpi, Jean Charles Quinton, and Giovanni Pezzulo. How active perception and attractor dynamics shape perceptual categorization: A computational model. *Neural Networks*, 60:1–16, 2014.

[8] B Bridgeman. Eye movements. *The Encyclopedia of Human Behaviour*, 2:160–166, 2012.

[9] Dora Matzke, Sander Nieuwenhuis, Hedderik van Rijn, Heleen A Slagter, Maurits W van der Molen, and Eric-Jan Wagenmakers. The effect of horizontal eye movements on free recall: a preregistered adversarial collaboration. *Journal of Experimental Psychology: General*, 144(1):e1, 2015.

[10] Dario Floreano, Toshifumi Kato, Davide Marocco, and Sauser Eric. Coevolution of active vision and feature selection. *Biological cybernetics*, 90(3):218–228, 2004.

[11] Gentaro Morimoto and Takashi Ikegami. Evolution of plastic sensory-motor coupling and dynamic categorization. *Proceedings of Artificial Life IX*, pages 188–193, 2004.

[12] Stefano Nolfi and Davide Marocco. Evolving visually-guided robots able to discriminate between different landmarks. In *In From Animals to Animats 6. Proceedings of the sixth International Conference on Simulation of Adaptive Behavior SAB-00*. Citeseer, 2000.

[13] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):509–522, 2002.

[14] David G Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial intelligence*, 31(3):355–395, 1987.

[15] Tamar Avraham and Michael Lindenbaum. Esaliency (extended saliency): Meaningful attention using stochastic image modeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(4):693–708, 2010.

[16] Igor Kagan and Ziad M Hafed. Active vision: microsaccades direct the eye to where it matters most. *Current Biology*, 23(17):712–714, 2013.

[17] Martina Poletti and Michele Rucci. Active vision: Adapting how to look. *Current Biology*, 23(17):R718–R720, 2013.

[18] Edgar Osuna, Robert Freund, and Federico Girosit. Training support vector machines: an application to face detection. In *Computer vision and pattern recognition, 1997. Proceedings., 1997 IEEE computer society conference on*, pages 130–136. IEEE, 1997.

[19] Henry A Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE Transactions on pattern analysis and machine intelligence*, 20(1): 23–38, 1998.

[20] Michael Burl, Markus Weber, and Pietro Perona. A probabilistic approach to object recognition using local photometry and global geometry. *Computer VisionECCV98*, pages 628–641, 1998.

[21] S Nolfi. Adaptation as a more powerful tool than decomposition and integration: experimental evidences from evolutionary robotics. In *Fuzzy Systems Proceedings,*

1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, volume 1, pages 141–146. IEEE, 1998.

[22] John K Tsotsos. On the relative complexity of active vs. passive visual search. *International journal of computer vision*, 7(2):127–141, 1992.

[23] Marco Mirolli, Tomassino Ferrauto, and Stefano Nolfi. Categorization through evidence accumulation in an active vision system. *Connection Science*, 22(4):331–354, 2010.

[24] Toshifumi Kato and Dario Floreano. An evolutionary active-vision system. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 1, pages 107–114. IEEE, 2001.

[25] Guido Cornelis Henricus Eugene de Croon. *Adaptive Active Vision*. PhD thesis, Universiteit Maastricht, Gildeprint, The Netherlands, 3 2008.

[26] Joachim Denzler and Christopher M Brown. Information theoretic sensor data selection for active object recognition and state estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):145–157, 2002.

[27] Hermann Borotschnig, Lucas Paletta, and Axel Pinz. A comparison of probabilistic, possibilistic and evidence theoretic fusion schemes for active object recognition. *Computing*, 62(4):293–319, 1999.

[28] Hermann Borotschnig, Lucas Paletta, Manfred Prantl, and Axel Pinz. Appearance-based active object recognition. *Image and Vision Computing*, 18 (9):715–727, 2000.

[29] Elio Tuci. Evolutionary swarm robotics: genetic diversity, task-allocation and task-switching. In *International Conference on Swarm Intelligence*, pages 98–109. Springer, 2014.

[30] Davide Marocco and Dario Floreano. Active vision and feature selection in evolutionary behavioral systems. *From animals to animats*, 7:247–255, 2002.

[31] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[32] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[33] Mototaka Suzuki and Dario Floreano. Evolutionary active vision toward three dimensional landmark-navigation. In *International Conference on Simulation of Adaptive Behavior*, pages 263–273. Springer, 2006.

[34] Mototaka Suzuki and Dario Floreano. Active vision for neural development and landmark navigation. In *50th Anniversary Summit of Artificial Intelligence*, number LIS-CONF-2006-011, pages 247–248, 2006.

[35] Inman Harvey, Philip Husbands, and David Cliff. *Seeing the light: Artificial evolution, real vision.* School of Cognitive and Computing Sciences, University of Sussex Falmer, 1994.

[36] Oren Barkan, Jonathan Weill, Lior Wolf, and Hagai Aronowitz. Fast high dimensional vector multiplication face recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1960–1967, 2013.

[37] Svein Magnussen. Low-level memory processes in vision. *Trends in neurosciences*, 23(6):247–251, 2000.

[38] Olivier Le Meur, Patrick Le Callet, Dominique Barba, Dominique Thoreau, and Edouard Francois. From low-level perception to high-level perception: a coherent approach for visual attention modeling. In *Electronic Imaging 2004*, pages 284–295. International Society for Optics and Photonics, 2004.

[39] Emanuel Diamant. Unveiling the mystery of visual information processing in human brain. *Brain research*, 1225:171–178, 2008.

[40] Massimiliano Schembri and Marta Olivetti Belardinelli. Evolved simulated agents exhibit size constancy abilities in solving an online size discrimination task. In *EAPCogSci*, 2015.

[41] Martin Szummer and Rosalind W Picard. Indoor-outdoor image classification. In *Content-Based Access of Image and Video Database, 1998. Proceedings., 1998 IEEE International Workshop on*, pages 42–51. IEEE, 1998.

[42] Jiebo Luo and Andreas Savakis. Indoor vs outdoor classification of consumer photographs using low-level and semantic features. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 2, pages 745–748. IEEE, 2001.

[43] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Scene classification via plsa. *Computer Vision–ECCV 2006*, pages 517–530, 2006.

[44] Marco Pirrone. *Active Vision and Visual Attention for Indoor Environment Classification.* PhD thesis, Universita degli Studi di Roma La Sapienza, 2003.

[45] Matti Pietikäinen, Abdenour Hadid, Guoying Zhao, and Timo Ahonen. Local binary patterns for still images. In *Computer vision using local binary patterns*, pages 13–47. Springer, 2011.

[46] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. *Computer vision-eccv 2004*, pages 469–481, 2004.

[47] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12):2037–2041, 2006.

[48] Daniel S Margulies, Satrajit S Ghosh, Alexandros Goulas, Marcel Falkiewicz, Julia M Huntenburg, Georg Langs, Gleb Bezgin, Simon B Eickhoff, F Xavier Castellanos, Michael Petrides, et al. Situating the default-mode network along a principal gradient of macroscale cortical organization. *Proceedings of the National Academy of Sciences*, 113(44):12574–12579, 2016.

[49] Hou Beiping and Zhu Wen. Fast human detection using motion detection and histogram of oriented gradients. *JCP*, 6(8):1597–1604, 2011.

[50] Ekaterina Zaytseva, Santi Seguí, and Jordi Vitria. Sketchable histograms of oriented gradients for object detection. In *Iberoamerican Congress on Pattern Recognition*, pages 374–381. Springer, 2012.

[51] Stefanos Stefanou and Antonis Argyros. Efficient scale and rotation invariant object detection based on hogs and evolutionary optimization techniques. *Advances in Visual Computing*, pages 220–229, 2012.

[52] J Kevin O'Regan and Alva Noë. A sensorimotor account of vision and visual consciousness. *Behavioral and brain sciences*, 24(05):939–973, 2001.

[53] Vittorio Gallese and George Lakoff. The brain's concepts: The role of the sensory-motor system in conceptual knowledge. *Cognitive neuropsychology*, 22(3-4):455–479, 2005.

[54] Alessandro DellAnna, Alfredo Paternoster, et al. Phenomenal consciousness and the sensorimotor approach. a critical account. *Open Journal of Philosophy*, 3(04): 435, 2013.

[55] Alena Stasenko, Frank E Garcea, and Bradford Z Mahon. What happens to the motor theory of perception when the motor system is damaged? *Language and cognition*, 5(2-3):225–238, 2013.

[56] Caroline Whyatt and Cathy Craig. Sensory-motor problems in autism. *Frontiers in integrative neuroscience*, 7:51, 2013.

[57] Martin Peniak, Davide Marocco, Salomon Ramirez-Contla, and Angelo Cangelosi. Active vision for navigating unknown environments: An evolutionary robotics approach for space research. In *ESA Special Publication*, volume 673, 2009.

[58] Jose Nunez-Varela and Jeremy L Wyatt. Models of gaze control for manipulation tasks. *ACM Transactions on Applied Perception (TAP)*, 10(4):20, 2013.

[59] Laurent Itti and Pierre Baldi. A principled approach to detecting surprising events in video. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 631–637. IEEE, 2005.

[60] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[61] Renwu Gao, Faisal Shafait, Seiichi Uchida, and Yaokai Feng. A hierarchical visual saliency model for character detection in natural scenes. In *International Workshop on Camera-Based Document Analysis and Recognition*, pages 18–29. Springer, 2013.

[62] Guangyu Zhong, Risheng Liu, Junjie Cao, and Zhixun Su. A generalized nonlocal mean framework with object-level cues for saliency detection. *The Visual Computer*, 32(5):611–623, 2016.

[63] Saumya Jetley, Naila Murray, and Eleonora Vig. End-to-end saliency mapping via probability distribution prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5753–5761, 2016.

[64] Wei Zhang, Ali Borji, Zhou Wang, Patrick Le Callet, and Hantao Liu. The application of visual saliency models in objective image quality assessment: A statistical evaluation. *IEEE transactions on neural networks and learning systems*, 27(6): 1266–1278, 2016.

[65] Feng Qi, Debin Zhao, Shaohui Liu, and Xiaopeng Fan. 3d visual saliency detection model with generated disparity map. *Multimedia Tools and Applications*, 76(2): 3087–3103, 2017.

[66] Federico Perazzi, Philipp Krahenbuhl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 733–740. IEEE, 2012.

[67] Antonio Torralba. Modeling global scene factors in attention. *JOSA A*, 20(7): 1407–1418, 2003.

[68] Lingyun Zhang, Matthew H Tong, Tim K Marks, Honghao Shan, and Garrison W Cottrell. Sun: A bayesian framework for saliency using natural statistics. *Journal of vision*, 8(7):32–32, 2008.

[69] Ali Borji, Dicky N Sihite, and Laurent Itti. An object-based bayesian framework for top-down visual attention. In *AAAI*, pages 1529–1535, 2012.

[70] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[71] Demetri Terzopoulos and Tamer F Rabie. Animat vision: Active vision in artificial animals. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 801–808. IEEE, 1995.

[72] Silviu Minut and Sridhar Mahadevan. A reinforcement learning model of selective visual attention. In *Proceedings of the fifth international conference on Autonomous agents*, pages 457–464. ACM, 2001.

[73] Teresa A Vidal-Calleja, Alberto Sanfeliu, and Juan Andrade-Cetto. Action selection for single-camera slam. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(6):1567–1581, 2010.

[74] Pablo Guerrero, Javier Ruiz-Del-Solar, Miguel Romero, and Sergio Angulo. Task-oriented probabilistic active vision. *International Journal of Humanoid Robotics*, 7(3):451–476, 2010.

[75] Amaury Dame and Eric Marchand. Using mutual information for appearance-based visual path following. *Robotics and Autonomous Systems*, 61(3):259–270, 2013.

[76] Andrew J Davison. Active search for real-time vision. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 66–73. IEEE, 2005.

[77] Ming Liu, Francis Colas, and Roland Siegwart. Regional topological segmentation based on mutual information graphs. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3269–3274. IEEE, 2011.

[78] Josep M Porta, Bas Terwijn, and B Krose. Efficient entropy-based action selection for appearance-based robot localization. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 2842–2847. IEEE, 2003.

[79] Rudolph Triebel, Hugo Grimmett, Rohan Paul, and Ingmar Posner. Introspective active learning for scalable semantic mapping. In *Workshop. Robotics Science and Systems (RSS)*, pages 809–816, 2013.

[80] Mark Peters and Arcot Sowmya. Active vision and adaptive learning. In *Proceedings of the 15th. Conference on Intelligent Robots and Computer Vision*, volume 2904, pages 413–424, 1996.

[81] Alois Unterholzner and Hans-Joachim Wuensche. Hybrid adaptive control of an active multi-focal vision system. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 534–539. IEEE, 2010.

[82] GCHE de Croon, Ida G Sprinkhuizen-Kuyper, and Eric O Postma. Comparing active vision models. *Image and Vision Computing*, 27(4):374–384, 2009.

[83] Marco F Huber, Tobias Dencker, Masoud Roschani, and Jürgen Beyerer. Bayesian active object recognition via gaussian process regression. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 1718–1725. IEEE, 2012.

[84] Andreas Seekircher, Tim Laue, and Thomas Röfer. Entropy-based active vision for a humanoid soccer robot. *RoboCup 2010: Robot Soccer World Cup XIV*, pages 1–12, 2011.

[85] Alex Holub, Pietro Perona, and Michael C Burl. Entropy-based active learning for object recognition. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[86] Vignesh Ramanathan and Axel Pinz. Active object categorization on a humanoid robot. *VISAPP*, 11:235–241, 2011.

[87] Björn Browatzki, Vadim Tikhanoff, Giorgio Metta, Heinrich H Bülthoff, and Christian Wallraven. Active object recognition on a humanoid robot. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2021–2028. IEEE, 2012.

[88] Tal Arbel and Frank P Ferrie. Entropy-based gaze planning. *Image and vision computing*, 19(11):779–786, 2001.

[89] Tal Arbel and Frank P Ferrie. Interactive visual dialog. *Image and Vision Computing*, 20(9):639–646, 2002.

[90] Thomas Leopold, Gabriele Kern-Isberner, and Gabriele Peters. Combining reinforcement learning and belief revision-a learning system for active vision. In *BMVC*, pages 1–10, 2008.

[91] Francesco Pugliese. Development of categorisation abilities in evolving embodied agents: A study of internal representations with external social inputs. In *Evolution, Complexity and Artificial Life*, pages 123–134. Springer, 2014.

[92] Stefano Nolfi and Domenico Parisi. Evolving non-trivial behaviors on real robots: an autonomous robot that picks up objects. *Topics in artificial intelligence*, pages 243–254, 1995.

[93] Elio Tuci, Gianluca Massera, and Stefano Nolfi. Active categorical perception of object shapes in a simulated anthropomorphic robotic arm. *IEEE transactions on evolutionary computation*, 14(6):885–899, 2010.

[94] Lee Altenberg et al. The evolution of evolvability in genetic programming. *Advances in genetic programming*, 3:47–74, 1994.

[95] Efrén Mezura-Montes and Carlos A Coello Coello. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary computation*, 9(1):1–17, 2005.

[96] Alan Schultz and JOHN GREFENSTETTE. Using a genetic algorithm to learn behaviors for autonomous vehicles. In *Guidance, Navigation and Control Conference*, page 4463, 1992.

[97] Dave Cliff, Phil Husbands, and Inman Harvey. Explorations in evolutionary robotics. *Adaptive behavior*, 2(1):73–110, 1993.

[98] Angelo Cangelosi and Domenico Parisi. How nouns and verbs differentially affect the behavior of artificial organisms. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society, London: LEA*, pages 170–175, 2001.

[99] Angelo Cangelosi. The sensorimotor bases of linguistic structure: Experiments with grounded adaptive agents. In *Proceedings of the eighth international conference on the simulation of adaptive behaviour: from animals to animats*, volume 8, pages 487–496, 2004.

[100] DA Sofge, Mitchell A Potter, Magdalena D Bugajska, and Alan C Schultz. Challenges and opportunities of evolutionary robotics. *arXiv preprint arXiv:0706.0457*, 2007.

[101] Craig W Reynolds. Evolution of obstacle avoidance behavior: using noise to promote robust solutions. *Advances in genetic programming*, 1:221–241, 1994.

[102] Nick Jakobi, Phil Husbands, and Inman Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. *Advances in artificial life*, pages 704–720, 1995.

[103] Olalekan Lanihun, Bernie Tiddeman, Elio Tuci, and Patricia Shaw. Improving active vision system categorization capability through histogram of oriented

gradients. In *Conference Towards Autonomous Robotic Systems*, pages 143–148. Springer, 2015.

[104] Stefano Nolfi. Power and the limits of reactive agents. *Neurocomputing*, 42(1): 119–145, 2002.

[105] GCHE de Croon, S Nolfi, and EO Postma. Towards pro-active embodied agents: on the importance of neural mechanisms suitable to process time information. *Complex Engineered Systems*, pages 338–363, 2006.

[106] Mototaka Suzuki. Visuo-motor coordination in bipedal humanoid robot walking. In *Future Generation Communication and Networking Symposia, 2008. FGCNS'08. Second International Conference on*, volume 3, pages 207–208. IEEE, 2008.

[107] Elio Tuci, Stefano Nolfi, Marco Mirolli, Tomassino Ferrauto, Gianluca Massera, et al. Two examples of active categorisation processes distributed over time. In *Proceedings of the Ninth International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, pages 49–56, 2009.

[108] Olalekan Lanihun, Bernie Tiddeman, Elio Tuci, and Patricia Shaw. Enhancing active vision system categorization capability through uniform local binary patterns. In *Artificial Life and Intelligent Agents Symposium*, pages 31–43. Springer, 2014.

[109] Charles E Schroeder and John Foxe. Multisensory contributions to low-level,unisensoryprocessing. *Current opinion in neurobiology*, 15(4):454–458, 2005.

[110] Navid Serrano, Andreas Savakis, and A Luo. A computationally efficient approach to indoor/outdoor scene classification. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 4, pages 146–149. IEEE, 2002.

[111] Axel Pinz. Object categorization. *Foundations and Trends® in Computer Graphics and Vision*, 1(4):255–353, 2005.

[112] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on statistical learning in computer vision, ECCV*, volume 2, page 7, 2004.

[113] John Winn, Antonio Criminisi, and Thomas Minka. Object categorization by learned universal visual dictionary. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1800–1807. IEEE, 2005.

[114] Markus Weber, Max Welling, and Pietro Perona. Towards automatic discovery of object categories. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 101–108. IEEE, 2000.

[115] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.

[116] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[117] Lamberto Ballan, Marco Bertini, Alberto Del Bimbo, Andrea M Serain, Giuseppe Serra, and Benito F Zaccone. Combining generative and discriminative models for classifying social images from 101 object categories. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1731–1734. IEEE, 2012.

[118] Jiann-Ming Wu and Zheng-Han Lin. Learning generative models of natural images. *Neural Networks*, 15(3):337–347, 2002.

[119] Ravi Ramamoorthi and James Arvo. Creating generative models from range images. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 195–204. ACM Press/Addison-Wesley Publishing Co., 1999.

[120] Ruiping Wang, Huimin Guo, Larry S Davis, and Qionghai Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2496–2503. IEEE, 2012.

[121] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1458–1465. IEEE, 2005.

[122] Gaurav Sharma, Frédéric Jurie, and Cordelia Schmid. Discriminative spatial saliency for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3506–3513. IEEE, 2012.

[123] Zhuowen Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1589–1596. IEEE, 2005.

[124] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[125] Andreas Opelt, Michael Fussenegger, Axel Pinz, and Peter Auer. Weak hypotheses and boosting for generic object detection and recognition. *Computer vision-ECCV 2004*, pages 71–84, 2004.

[126] Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Gaussian processes for object categorization. *International journal of computer vision*, 88 (2):169–188, 2010.

[127] Amin Shah-Hosseini and Gerald M Knapp. Semantic image retrieval based on probabilistic latent semantic analysis. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 703–706. ACM, 2006.

[128] Dan Oneata. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty*, pages 1–7, 1999.

[129] Xiaogang Wang and Eric Grimson. Spatial latent dirichlet allocation. In *Advances in neural information processing systems*, pages 1577–1584, 2008.

[130] James Philbin, Josef Sivic, and Andrew Zisserman. Geometric latent dirichlet allocation on a matching graph for large-scale image datasets. *International journal of computer vision*, 95(2):138–153, 2011.

[131] Teofilo De Campos, Gabriela Csurka, and Florent Perronnin. Images as sets of locally weighted features. *Computer Vision and Image Understanding*, 116(1): 68–85, 2012.

[132] Yong Jae Lee and Kristen Grauman. Object-graphs for context-aware visual category discovery. *IEEE transactions on pattern analysis and machine intelligence*, 34(2):346–358, 2012.

[133] Shahab Ensafi, Shijian Lu, Ashraf A Kassim, and Chew Lim Tan. A bag of words based approach for classification of hep-2 cell images. In *Pattern Recognition Techniques for Indirect Immunofluorescence Images (I3A), 2014 1st Workshop on*, pages 29–32. IEEE, 2014.

[134] Giulio Iannello, Leonardo Onofri, and Paolo Soda. A bag of visual words approach for centromere and cytoplasmic staining pattern classification on hep-2 images. In *Computer-based medical systems (CBMS), 2012 25th international symposium on*, pages 1–6. IEEE, 2012.

[135] Nazli Deniz Cagatay and Mihai Datcu. Bag-of-visual-words model for classification of interferometric sar images. In *EUSAR 2016: 11th European Conference on Synthetic Aperture Radar, Proceedings of*, pages 1–4. VDE, 2016.

[136] Piotr Koniusz, Fei Yan, Philippe-Henri Gosselin, and Krystian Mikolajczyk. Higher-order occurrence pooling for bags-of-words: Visual concept detection. *IEEE transactions on pattern analysis and machine intelligence*, 39(2):313–326, 2017.

[137] Hugo Jair Escalante, Víctor Ponce-López, Sergio Escalera, Xavier Baró, Alicia Morales-Reyes, and José Martínez-Carranza. Evolving weighting schemes for the bag of visual words. *Neural Computing and Applications*, 28(5):925–939, 2017.

[138] Robert Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, page 2. IEEE, 2003.

[139] Mathieu Aubry, Daniel Maturana, Alexei A Efros, Bryan C Russell, and Josef Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3762–3769, 2014.

[140] David J Crandall and Daniel P Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *European Conference on Computer Vision*, pages 16–29. Springer, 2006.

[141] David Crandall, Pedro Felzenszwalb, and Daniel Huttenlocher. Spatial priors for part-based recognition using statistical models. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 10–17. IEEE, 2005.

[142] Patrick Ott and Mark Everingham. Shared parts for deformable part-based models. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1513–1520. IEEE, 2011.

[143] Megha Pandey and Svetlana Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1307–1314. IEEE, 2011.

[144] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer vision and Image understanding*, 106(1):59–70, 2007.

[145] Juan Carlos Niebles and Li Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[146] Christian A Mueller, Kaustubh Pathak, and Andreas Birk. Object shape categorization in rgbd images using hierarchical graph constellation models based on unsupervisedly learned shape parts described by a set of shape specificity levels. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3053–3060. IEEE, 2014.

[147] Paul Viola and Michael Jones. Robust real-time object detection. *International Journal of Computer Vision*, 4(34–47), 2001.

[148] Antonio Torralba, Kevin P Murphy, and William T Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, page 2. IEEE, 2004.

[149] Thrasyvoulos N Pappas. An adaptive clustering algorithm for image segmentation. *IEEE Transactions on signal processing*, 40(4):901–914, 1992.

[150] Husni A Al-Muhtaseb, Sabri A Mahmoud, and Rami S Qahwaji. Recognition of off-line printed arabic text using hidden markov models. *Signal processing*, 88(12): 2902–2912, 2008.

[151] Abdeljalil Gattal and Youcef Chibani. Segmentation and recognition strategy of handwritten connected digits based on the oriented sliding window. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 297–301. IEEE, 2012.

[152] Duangmanee Putthividhy, Hagai T Attias, and Srikantan S Nagarajan. Topic regression multi-modal latent dirichlet allocation for image annotation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3408–3415. IEEE, 2010.

[153] Robert Fergus, M Weber, and Pietro Perona. Efficient methods for object recognition using the constellation model. *California Inst. Technol., Tech. Rep*, page 54, 2001.

[154] Yasunori Kamiya, Tomokazu Takahashi, Ichiro Ide, and Hiroshi Murase. A multimodal constellation model for object category recognition. *Advances in multimedia modeling*, pages 310–321, 2009.

[155] Baofeng Guo, Yuesong Lin, Dongliang Peng, and Anke Xue. Band selection for hyperspectral image classification by a sliding window model. In *Seventh International Symposium on Multispectral Image Processing and Pattern Recognition (MIPPR2011)*, pages 80061–80061. International Society for Optics and Photonics, 2011.

[156] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

[157] Jinyi Zou, Wei Li, Chen Chen, and Qian Du. Scene classification using local and global features with collaborative representation fusion. *Information Sciences*, 348: 209–226, 2016.

[158] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Scene classification using a hybrid generative/discriminative approach. *IEEE transactions on pattern analysis and machine intelligence*, 30(4):712–727, 2008.

[159] Aymen Shabou and Hervé LeBorgne. Locality-constrained and spatially regularized coding for scene categorization. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3618–3625. IEEE, 2012.

[160] Valérie Goffaux, Corentin Jacques, André Mouraux, Aude Oliva, Philippe Schyns, and Bruno Rossion. Diagnostic colours contribute to the early stages of scene categorization: Behavioural and neurophysiological evidence. *Visual Cognition*, 12(6):878–892, 2005.

[161] Jan C Van Gemert, J Geusebroek, Cor J Veenman, Cees GM Snoek, and Arnold WM Smeulders. Robust scene categorization by learning image statistics in context. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 105–105. IEEE, 2006.

[162] Jianxin Wu and Jim M Rehg. Centrist: A visual descriptor for scene categorization. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1489–1501, 2011.

[163] Jan C Van Gemert, Jan-Mark Geusebroek, Cor J Veenman, and Arnold WM Smeulders. Kernel codebooks for scene categorization. In *European conference on computer vision*, pages 696–709. Springer, 2008.

[164] H Madokoro, A Yamanashi, and K Sato. Unsupervised semantic indoor scene classification for robot vision based on context of features using gist and hsv-sift. *Pattern Recognition in Physics*, 1(1):93–103, 2013.

[165] Darius Burschka and Gregory Hager. Scene classification from dense disparity maps in indoor environments. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 708–712. IEEE, 2002.

[166] Agnes Swadzba and Sven Wachsmuth. Indoor scene classification using combined 3d and gist features. In *Asian conference on computer vision*, pages 201–215. Springer, 2010.

[167] Zhibin Niu, Yue Zhou, and Kun Shi. A hybrid image representation for indoor scene classification. In *Image and Vision Computing New Zealand (IVCNZ), 2010 25th International Conference of*, pages 1–7. IEEE, 2010.

[168] Munawar Hayat, Salman H Khan, Mohammed Bennamoun, and Senjian An. A spatial layout and scale invariant feature representation for indoor scene classification. *IEEE Transactions on Image Processing*, 25(10):4829–4841, 2016.

[169] Lei Shi, Sarath Kodagoda, and Ravindra Ranasinghe. Fast indoor scene classification using 3d point clouds. In *Australasian Conference on Robotics and Automation*. The ACRA 2011 Organising Committee, 2011.

[170] Stevica S Cvetkovic, Saša V Nikolić, and Slobodan Ilic. Effective combining of color and texture descriptors for indoor-outdoor image classification. *Facta Universitatis, Series: Electronics and Energetics*, 27(3):399–410, 2014.

[171] Yang Liu and Xueqing Li. Indoor-outdoor image classification using mid-level cues. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, pages 1–5. IEEE, 2013.

[172] Waleed Tahir, Aamir Majeed, and Tauseef Rehman. Indoor/outdoor image classification using gist image features and neural network classifiers. In *High-Capacity Optical Networks and Enabling/Emerging Technologies (HONET), 2015 12th International Conference on*, pages 1–5. IEEE, 2015.

[173] Chen Chen, Yuzhuo Ren, and C-C Jay Kuo. Large-scale indoor/outdoor image classification via expert decision fusion (edf). In *Asian Conference on Computer Vision*, pages 426–442. Springer, 2014.

[174] Christina Pavlopoulou and X Yu Stella. Indoor-outdoor classification with human accuracies: Image or edge gist? In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 41–47. IEEE, 2010.

[175] Zhijie Zhao, Haitao Wang, Xuesong Jin, Huadong Sun, and Qian Wu. Indoor and outdoor scene classification method based on fourier transform. *International Journal of Hybrid Information Technology*, 7(5):341–350, 2014.

[176] Rui Wu, Zhipeng Ye, Peng Liu, Xianglong Tang, and Wei Zhao. Knowledge as action: A cognitive framework for indoor scene classification. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 3141–3144. IEEE, 2015.

[177] Rong Wang, Zhiliang Wang, et al. Indoor scene classification based on the bag-of-words model of local feature information gain. *IEICE TRANSACTIONS on Information and Systems*, 96(4):984–987, 2013.

[178] Matthew Traherne and Sameer Singh. An integrated approach to automatic indoor outdoor scene classification in digital images. In *IDEAL*, pages 511–516. Springer, 2004.

[179] Simone Bianco, Gianluigi Ciocca, Claudio Cusano, and Raimondo Schettini. Improving color constancy using indoor–outdoor image classification. *IEEE Transactions on image processing*, 17(12):2381–2392, 2008.

[180] A Nadian Ghomsheh and Alireza Talebpour. A new method for indoor-outdoor image classification using color correlated temperature. *Int. J. Image Process*, 6 (3):167–181, 2012.

[181] Weiping Wang, Qiang Chang, Qun Li, Zesen Shi, and Wei Chen. Indoor-outdoor detection using a smart phone sensor. *Sensors*, 16(10):1563, 2016.

[182] Valentin Radu, Panagiota Katsikouli, Rik Sarkar, and Mahesh K Marina. A semi-supervised learning approach for robust indoor-outdoor detection with smartphones. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 280–294. ACM, 2014.

[183] Jack Collier and Alejandro Ramirez-Serrano. Environment classification for indoor/outdoor robotic mapping. In *Computer and Robot Vision, 2009. CRV'09. Canadian Conference on*, pages 276–283. IEEE, 2009.

[184] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

[185] Aditya Vailaya, Mário AT Figueiredo, Anil K Jain, and Hong-Jiang Zhang. Image classification for content-based indexing. *IEEE transactions on image processing*, 10(1):117–130, 2001.

[186] Annalisa Barla, Francesca Odone, and Alessandro Verri. Histogram intersection kernel for image classification. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 3, pages III–513. IEEE, 2003.

[187] Claudio Mattiussi and Dario Floreano. Evolution of analog networks using local string alignment on highly reorganizable genomes. In *Evolvable Hardware, 2004. Proceedings. 2004 NASA/DoD Conference on*, pages 30–37. IEEE, 2004.

[188] Kenneth O Stanley and Risto Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21(6): 63–100, 2004.

[189] Dario Floreano, Peter Dürr, and Claudio Mattiussi. Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1(1):47–62, 2008.

[190] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

[191] Harold Szu and Ralph Hartley. Fast simulated annealing. *Physics letters A*, 122 (3-4):157–162, 1987.

[192] Scott Kirkpatrick, C Daniel Gelatt, Mario P Vecchi, et al. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[193] Jun Ye. Single valued neutrosophic cross-entropy for multicriteria decision making problems. *Applied Mathematical Modelling*, 38(3):1170–1175, 2014.

[194] Meimei Xia and Zeshui Xu. Entropy/cross entropy-based group decision making under intuitionistic fuzzy environment. *Information Fusion*, 13(1):31–47, 2012.

[195] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

[196] Che Lin, Venugopal V Veeravalli, and Sean P Meyn. A random search framework for convergence analysis of distributed beamforming with feedback. *IEEE Transactions on Information Theory*, 56(12):6133–6141, 2010.

[197] John H Holland. Genetic algorithms. *Scientific american*, 267(1):66–72, 1992.

[198] Colin Reeves. Genetic algorithms. In *Handbook of metaheuristics*, pages 55–82. Springer, 2003.

[199] Kalyanmoy Deb. An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering*, 186(2):311–338, 2000.

[200] Faustino J Gomez and Jürgen Schmidhuber. Co-evolving recurrent neurons learn deep memory pomdps. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pages 491–498. ACM, 2005.

[201] Li Wang and Dong-Chen He. Texture classification using texture spectrum. *Pattern Recognition*, 23(8):905–910, 1990.

[202] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[203] Juan E Tapia, Claudio A Perez, and Kevin W Bowyer. Gender classification from iris images using fusion of uniform local binary patterns. In *ECCV Workshops (2)*, pages 751–763, 2014.

[204] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[205] William T Freeman and Michal Roth. Orientation histograms for hand gesture recognition. In *International workshop on automatic face and gesture recognition*, volume 12, pages 296–301, 1995.

[206] William T Freeman, Ken-ichi Tanaka, Jun Ohta, and Kazuo Kyuma. Computer vision for computer games. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, pages 100–105. IEEE, 1996.

[207] Serge Belongie, Jitendra Malik, and Jan Puzicha. Matching shapes. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 454–461. IEEE, 2001.

[208] Ricardo Beira, Manuel Lopes, Miguel Praça, José Santos-Victor, Alexandre Bernardino, Giorgio Metta, Francesco Becchi, and Roque Saltarén. Design of the robot-cub (icub) head. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 94–100. IEEE, 2006.

[209] Nikolaos G Tsagarakis, Giorgio Metta, Giulio Sandini, David Vernon, Ricardo Beira, Francesco Becchi, Ludovic Righetti, Jose Santos-Victor, Auke Jan Ijspeert, Maria Chiara Carrozza, et al. icub: the design and realization of an open humanoid platform for cognitive and neuroscience research. *Advanced Robotics*, 21(10):1151–1175, 2007.

[210] Elio Tuci. The simple icub simulator used in the phd thesis. Department of Computer Science, School of Science and Technolog, Middlesex University, Room T112 Town Hall Building, Hendon Campus, The Burroughs, London, NW4 4BT, United Kingdom, 2016.

[211] Edgar A DeYoe and David C Van Essen. Concurrent processing streams in monkey visual cortex. *Trends in neurosciences*, 11(5):219–226, 1988.

[212] Chris Thornton. Separability is a learners best friend. In *4th Neural Computation and Psychology Workshop, London, 9–11 April 1997*, pages 40–46. Springer, 1998.

[213] Google. Google images. https://www.google.co.uk/imghp?hl=entab=wi, 2017.

[214] Chao Zhu, Charles-Edmond Bichot, and Liming Chen. Multi-scale color local binary patterns for visual object classes recognition. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3065–3068. IEEE, 2010.

[215] Yong Cheol Peter Cho, Sungmin Bae, Yongseok Jin, Kevin M Irick, and Vijaykrishnan Narayanan. Exploring gabor filter implementations for visual cortex modeling on fpga. In *Field Programmable Logic and Applications (FPL), 2011 International Conference on*, pages 311–316. IEEE, 2011.

[216] Judson P Jones and Larry A Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of neurophysiology*, 58(6):1233–1258, 1987.

[217] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[218] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3367–3375, 2015.