The
University
Of
Sheffield.

Bio-inspired foveal and peripheral visual sensing for saliency-based decision making in robotics

A thesis submitted to the University of Sheffield in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Uziel Jaramillo Avila

Department of Automatic Control and Systems Engineering

April 2021

To my family

Declaration

I, Uziel Jaramillo Avila, declare that the work presented in this thesis is my own. All material in this thesis which is not of my own work, has been properly accredited and referenced.

Acknowledgements

"You are lucky to be here. You were incalculably lucky to be born, and incredibly lucky to be brought up by a nice family that helped you get educated and encouraged you to go to Uni ... Well done you, for dragging yourself up by the shoelaces, but you were lucky. You didn't create the bit of you that dragged you up. They're not even your shoelaces."

- Tim Minchin

To Michelle, Pris, Raziel, Eva, Sean Anderson, everyone at ACSE and everyone back home. You know who you are, and I hope you know what you mean to me. They are your shoelaces.

Thanks to CONACyT and The University of Sheffield for their huge support.

Abstract

Computer vision is an area of research that has grown at immense speed in the last few decades, tackling problems towards scene understanding from very diverse fronts, such as image classification, object detection, localization, mapping and tracking. It has also been long understood that there are very valuable lessons to learn from biology and to be applied to this research field, where the human visual system is very likely the most studied brain mechanism.

The eye foveation system is a very good example of such lessons, since both machines and animals often face a similar dilemma; to prioritize visual areas of interest to faster process information, given limited computing power and from a field of view that is too wide to be simultaneously attended. While extensive models of artificial foveation have been presented, the re-emerging area of machine learning with deep neural networks has opened the question into how these two approaches can contribute to each other. Novel deep learning models often rely on the availability of substantial computing power, but areas of application face strict constraints, a good example are unmanned aerial vehicles, which in order to be autonomous should lift and power all their computing equipment.

In this work it is studied how applying a foveation principle to down-scale images can be used to reduce the number of operations required for object detection, and compare its effect to normally down-sampled images, given the prevalent number of operations by Convolutional Neural Network (CNN) layers. Foveation requires prior knowledge of regions of interest to center the fovea, this point in question is addressed by a merging of bottom-up saliency and top-down feedback of objects that the CNN has been trained to detect. Albeit saliency models have also been studied extensively in the last couple of decades, most often comparing their performance to human observer datasets, the question remains open into how they fit in wider information processing paradigms and into functional representations of the human brain. It is proposed here an information flow scheme that encompasses these principles.

Finally, to give to the model the capacity to operate coherently in the time domain, it adapts a representation of a well-established theory of the decision-making process that takes place in the basal ganglia region of the brain. The behaviour of this representation is then tested against human observer's data in an omnidirectional field of view, where the importance of selecting the most contextually relevant region of interest in each time-step is highlighted.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"If the entire $160 \times 175^\circ$ *subtended by each retina were populated by photo-receptors at foveal density, the optic nerve would comprise on the order of one billion nerve fibers, compared to about one million fibers in humans."* [2]

1.1 Background and motivation

A lot of studies have been developed to understand and mimic the functioning of a wide range of visual information processing stages in the human brain. Rather than presenting a new interpretation or implementation of any of these, here the motivation is to explore the feasibility of an overall modular system that incorporates the main principles of such advances already present in the literature, in a way that proves beneficial in engineering terms, takes into account their compatibility and allows to revise any given module when a breakthrough is presented in the literature, or to compare different approaches with similar goals. With these goals, the aim is to present an overall functioning system loop of how visual information is processed in the human brain, how it drives eye movement decision making and this in turn affects the subsequent visual information that is captured.

Vision is arguably the most important sense for the survival of a lot of animals, one that produces a huge amount of sensory input, for which a large part of the human brain is dedicated to its working. Despite the immediate parallel between how our eyes acquire visual information, and how a camera can serve the same goal for a robot, there are some very basic differences in how they work; conventional digital photographs have an uniform density of pixels, the first and last pixel take the same memory space and are equally important. Video cameras capture a series of images at a constant time interval (frame rate). In contrast, our visual system starts discerning information that does not appear to be relevant to the current task even before capturing it; the eye has a very small area with a high density of photo-receptors, the fovea, which captures with high acuity a visual field area of around twice the size of a thumbnail at arm's length [3], and yet producing about half of the visual input to the brain [4].

1.2 Aims and objectives

The overall goal of this project is to conceive an information flow system for visual data, grounded current understanding of how this process takes place in the human brain in order to take advantage of some of the same underlying principles. Our main focus is in information selection, under the axiom that humans and autonomous robots face similar dilemmas while navigating an unstructured environment; visual data is inherently bulky, and our brain is not capable of processing the full scene in real time, facing an equivalent limitation to what most embedded computing equipment will face, using state of the art techniques such as Convolutional Neural Networks (CNN).

Rather than aiming at creating exclusively a CNN implementation that encompasses some of these principles, a more modular approach is taken here, by using and adapting available methods that have proven effective in their particular goals; saliency models and CNN object detection methods are normally evaluated independently by comparing them against datasets established as ground truth. By taking these implementations as a starting point, here is presented an information cascading approach, mapped to current understanding of brain functionality (Fig. 1.1), where it can select prominent regions of interest from a wide angle view, accumulate evidence over time for switching regions of focus and feeding back goal-oriented information into the system as evidence for the next time step.

1.3 Thesis outline and contributions

Chapter 2 of this thesis starts with a background and literature review of relevant work, mainly in the topics of foveation and its computational representation, saliency, object detection through deep neural networks and decision making in the basal ganglia. In Chapter 3 a relatively delimited problem is first studied, exploring the effect of foveation in object classification performance with Convolutional Neural Networks, covering our contributions presented in; [6]. Chapter 4 broadens into studying how saliency can be used to select regions to foveate on, as well as the effect of top-down feedback information, it encompasses the work

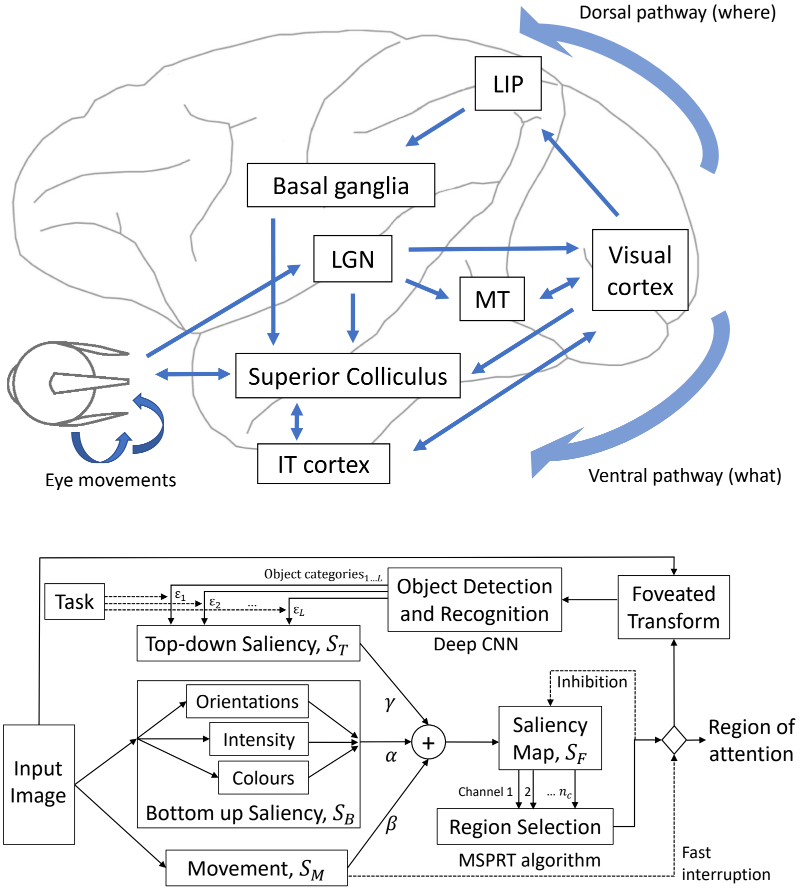


Figure 1.1: *Brain regions involved in visual processing emphasising the dorsal and ventral pathways: the where and what. It has been proposed that a bottom-up saliency map is formed in the primary visual cortex [5]. The basal ganglia is thought to accumulate evidence from different parts of the cortex, including those associated with eye movements, such as the lateral intraparietal area (LIP) and the frontal eye field (FEF).*

presented in ICAR [7]. In Chapter 5 a couple of additional measures to boost performance are studied; multiple foveas in each image and an explicit movement channel during training and testing for object detection. Chapter 6 expands the model into the time domain, thus requiring a strong decision making implementation. Here it is also modeled against data of human fixations in an omnidirectional field of view, it covers the work submitted to the Journal of Intelligent and Robotic Systems. Finally, Chapter 7 goes over the general conclusions and ideas for future work.

During the development of this thesis, the following contributions were made as peer reviewed publications:

a. A poster presentation in the 19th Towards Autonomous Robotic Systems

Conference (TAROS), entitled *"Top-Down Bottom-Up Visual Saliency for Mobile Robots Using Deep Neural Networks and Task-Independent Feature Maps"*, featuring the work done during the first year; [8].

b. Oral presentation in the 8th International Conference on Biomimetic and Biohybrid Systems with the title *"Foveated image processing for faster object detection and recognition in embedded systems using deep convolutional neural networks"*; [6].

c. Oral presentation in the 19th International Conference on Advanced Robotics (ICAR), entitled *"Visual saliency with foveated images for fast object detection and recognition in mobile robots using low-power embedded GPUs"*; [7]

d. Following the presentation in ICAR, an extended contribution was submitted by invitation to a special issue of the Journal of Intelligent and Robotic Systems (JINT). This work is entitled *"Robust top-down and bottom-up visual saliency for mobile robots using bio-inspired design principles"* and was submitted on July 2020, currently awaiting for feedback from reviewers.

Chapter 2

Background and literature review

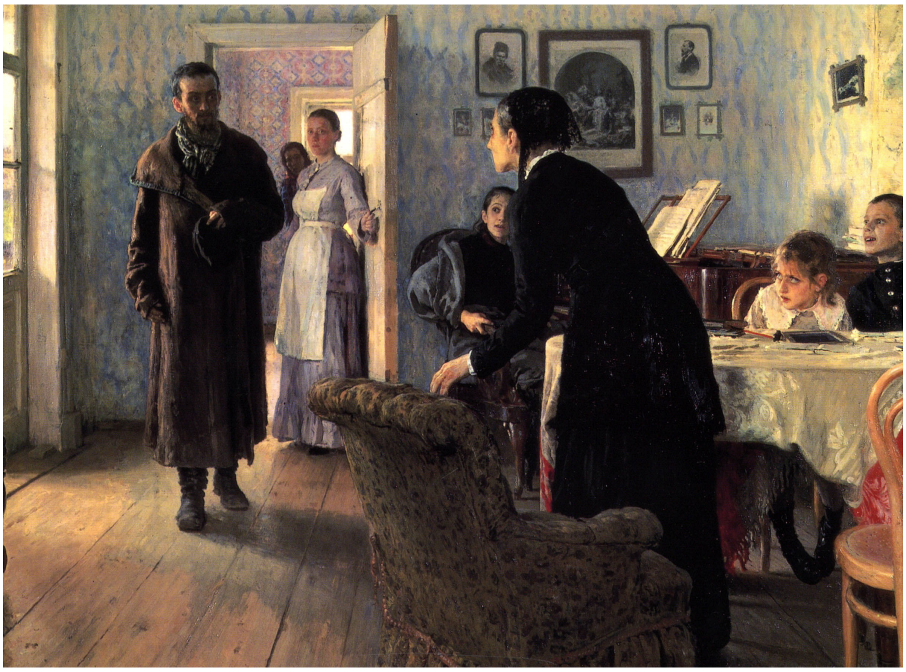


Figure 2.1: *Unexpected Visitors (c. 1886), by Ilya Repin. Tretyakov Gallery, Russia [9].*

2.1 The study of eye movements

Alfred L. Yarbus was a pioneer in the analysis of eye movements, during the 1950's and 1960's he created several studies of human behaviour. In a particularly prominent one, he asked participants to perform different observation tasks. For example, a subject was asked to observe the "Unexpected Visitors" painting (Fig. 2.1) while trying to gather the following information:

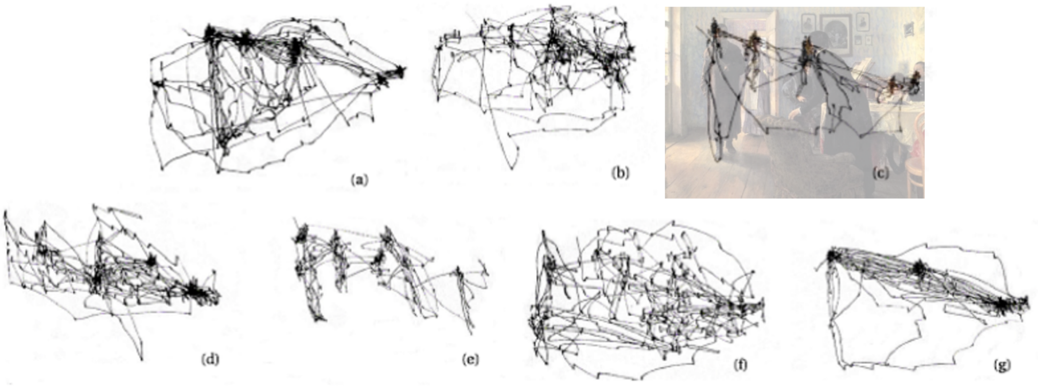


Figure 2.2: *Eye movements made by a single observer during 3 minutes, while examining the "Unexpected Visitors" painting with 7 different goals (listed in Section 2.1). Adapted from [10, Fig. 109]*

a. Free viewing

b. Material circumstances of the family?

c. Ages of the people?

d. What was the family doing before the arrival of the visitor?

e. Remember the clothes

f. Remember the positions of people and objects

g. How long has the visitor been away?

The tracked eye movements for one person performing this exercise is shown in Fig. 2.2. It is a good illustration of how the goal at hand influences the visual stimulus that becomes relevant. Studies of the interconnections of the human brain and its functioning have shown remarkable progress in the last few decades: With the help of new technologies such as functional Magnetic Resonance Imagining (fMRI), it is now possible to obtain data from normal functioning brain activity of subjects performing different tasks [11]. FMRI works by reading the blood flow in the brain, to which neural activity is related [12]. For these kind of tests, human subjects can perform conscious tasks, such as reading, moving a computer joystick, or remembering and visualizing memories, proving to be particularly useful for eye movement studies [13–15].

The human eye has a very small visual area with sharp vision, created by a high concentration of photoreceptor cells in the fovea, these cells detect light changes and are connected to the brain through the optic nerve [16]. The fovea accounts for around half of the photoreceptors, and the other half is distributed along the rest of the retina. Photoreceptors are classified as rod cells (or rodes) and cone cells in the visual system of mammals. Rodes are mostly distributed in the periphery of the retina, they are more sensitive to light than cones, but less so to color changes (they are solely attributed for allowing us to see at night, which also accounts for the effect that colors are more difficult to distinguish under poor lighting). On the other hand, cones are highly concentrated in the fovea, creating the effect of a small area of sharp vision [16]. Fig. 2.3 shows a diagram of the distribution of rods and cones in the human eye, illustrating the higher concentration on cones in the fovea.

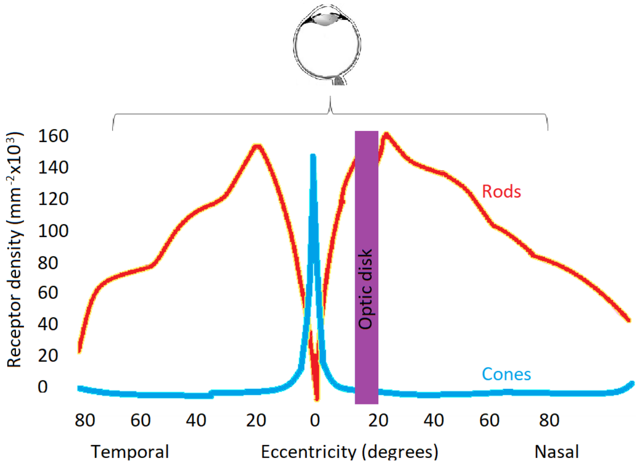


Figure 2.3: "Distribution of rods and cones in the human retina: cones are present at a low density throughout the retina, with a sharp peak in the center of the fovea" [17]. Redrawn from [17]

It is evident that these underlying principles can provide similar advantages on the fields of robotics and computer vision, different strategies have been used through the years to imitate and take advantage of them, broadly following one of these approaches;

- **Sub-sampling** an image to mimic the foveal resolution effect. Meaning a software approach to foveation, with a few examples illustrated in Subsection 2.3.
- **Using two cameras per "eye"**, to simulate the contrast between foveal and peripheral vision, making it a hardware solution. Some of these examples are described in Subsection 2.4.
- **Saliency maps**, based on the idea of studying and predicting where a human observer would focus his/her attention while looking at an image (where the

small high acuity vision would be pointing), given its contents and context. Further detailed in Subsection 2.5.

- **Dynamic Vision Sensors (DVS)** [18] are a hardware alternative to conventional video cameras, they attempt to closely mimic the mammalian vision system by not relying on a fixed frame rate. They consist of a CMOS integrated circuit that detects light changes in each "pixel" and transmit the information as soon as, and only if, a big enough change has occurred. They work at a remarkable speed (with a latency in the order of microseconds). They have received a lot of attention in the last decade but are still in an early development stage, with resolutions around 256 × 256 [19, 20]. A few models have been presented of visual attention using Dynamic Vision Sensors, such as [21–23], these tend to be a lot faster, but in limited implementations; for example, Galluppi et al. [24] present an attention selection system using a DVS with the advantage of being processed by a SpiNNaker board [25], making it time efficient and a biologically realistic system. An illustration of DVS data is shown in Fig. 2.4.

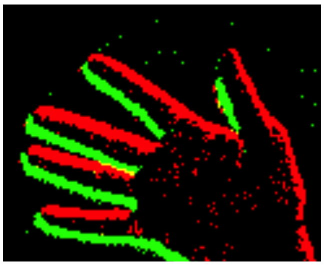


Figure 2.4: *Dynamic Vision Sensor sample information; DVS only produces output data for each pixel when it crosses a lighting change threshold, encompassing if it is either a positive or negative change and the location of the pixel.*

2.2 Foveal and peripheral vision

Human visual perception is dominated by the fovea, a small region of densely clustered photoreceptors in the retina, which accounts for just ∼2% of the visual field [26], but as much as ∼50% of the input to neurons in the primary visual cortex [27]. This amplification of the visual field in neural processing is the cortical magnification factor. In order to see with high acuity, humans actively redirect their fovea towards objects of interest.

The foveated image processing system in human vision contrasts strongly to how digital images are usually processed in computer vision, where large numbers of pixels are typically used to represent the entire field of view in dense, uniform sampling. Foveated transformation for digital image processing preserves high resolution in the foveal region, centred on an object of interest, whilst compressing the periphery, resulting in reduced image size but no reduction in the field of view.

The importance of peripheral vision is frequently neglected [28], which translates into it being overlooked in bio-inspired robotic implementations. Two important elements of peripheral vision are often not given enough emphasis [28]; (a) crowding (the difficulty to distinguish an element in a cluttered environment) is a bigger factor than loss of acuity for recognizing objects outside of the fovea, and (b) vision does not mainly consist of stitching together foveal views taken at different times. It is conceivable that these misconceptions cascade into engineering implementations of bio-inspired vision. For this reason it is worthwhile to define the rate at which performance decreases for different resolutions of peripheral vision in state-of-the-art location and classification system.

2.3 Foveation approaches in software

Fovea-like images have been explored in the context of robotics for a long time, under the assumption that the information placed in the fovea is the main point of interest. There are several different approaches to use in computer vision to obtain foveated images, such as the log-polar transform [29], the reciprocal wedge-transform [30], and Wavelet and Fast Fourier transformations [31], addressing the issue of non-linearity in log-polar transformation. Akbas and Eckstein [32] illustrated the advantages of foveated image processing with regard to improvements in computational efficiency in detection objects at different numbers of fixations, by using image templates and Gabor features of objects. The requirement of image templates makes it difficult to scale up.

In some examples of the versatility of foveation principles, Wei and Li [33] also base their work on foveated wavelet transforms, with the goal of motion estimation and object tracking, using multiple foveations to reconstruct conventional images. Geisler and Perry [34] use foveation to encode video files with a better compression rate, by taking advantage of their specific encoding formatting; waonhile Bailey and Bouganis [35] present an Field Programmable Gate Arrays (FPGA) implementation, enabling its use in real time. This study is also interesting in that it represents a middle point between software and hardware implementations.

2.4 Foveation approaches in hardware

Given the contrasting properties between photoreceptors present in the periphery (rods) and photoreceptors in the fovea (cones), described previously in this chapter, another interesting engineering approach to foveation is to use at least two separate physical sensors to independently act as fovea and periphery. This type of hardware implementation can be very useful for purpose-built robots, and bring different intuitions than a software based foveation effect. For example, Bjorkman and Kragic [36] use two pairs of cameras (one as fovea and one as periphery, per eye), where the foveal cameras can move (pan/tilt) directed by cues from the peripheral ones.

Using several cameras provides the benefit of depth information, making it better suited for grasping applications. Ude et al. [37] present a theoretical analysis for a similar setup, with the goal of maintaining the object of interest centered in the foveal camera, based on the information provided by the peripheral ones. Ude et al. [38] presents with more detail the object recognition system, the state-of-the-art of which has drastically changed in the last two decades. Nonetheless, the basic principle of using a narrow field of view area with high resolution to ease the identification of the object of interest undoubtedly still applies.

Shibata et al. [39] present a model for a similar type of robotic headset, in tasks such as smooth pursuit with a very biomimetic approach, using control strategies based on the Vestibulo-Ocular Reflex (VOR) and Opto-kinetic Response (OKR). This implementation on a humanoid system with 30 degrees of freedom illustrates how quickly the complexity of head/eye kinematics adds up. Ude [40] used a headset with two cameras per eye with an early model of foveal camera object identification, using support vector machines learning; Kragic and Bjorkman [41] place a similar vision head system [42] on top of a robotic arm to demonstrate object grasping and manipulation, and Craye et al. [43] present a more modern implementation of such systems, as proof on concept for an on-the-fly object-oriented saliency learning and estimation system, once again using the foveal stream to provide object identification information. A particularly more meticulous project, in terms of biomimicry, by Dean et al. [44] is shown in Fig. 2.5.

2.5 Saliency maps

Considering how fast the human visual system allows us to interact with the environment (e.g. scan it to locate a specific object, find food, or detect an immediate threat), a robust system is needed to regulate this behaviour, by answering the

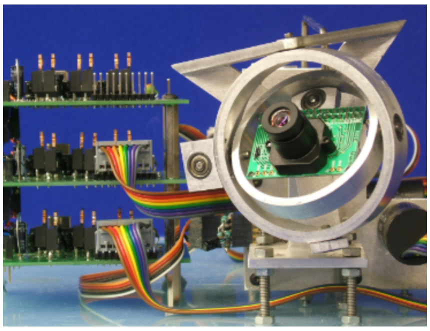


Figure 2.5: *The project "Functions of Distributed Plasticity in a Biologically-Inspired Adaptive Control Algorithm" [44] was notably focused on biomimicry, with the general goal of investigating image stabilization in a robotic eye [45], to determine if their model of cerebellar function worked in real-world conditions, instead of just simulation, also proposing a physical implementation of the mammalian vestibular system (shown, from [44]).*

question *"Where to look next?"*. The stimuli that drives us to look at something are often classified as either *bottom-up* or *top-down*, the former makes reference to purely visual stimuli (such as a bright color, an odd shape or a sudden movement), the latter is directed by the current task; while driving, it is paramount to be *looking at*[*] the road most of the time, even if it has an uniform and uninteresting shape.

In the last couple of decades, a lot of effort has also gone into the study of saliency maps, with the goal of ranking the conspicuity of the elements in a visual scene. Saliency maps have also gotten increasingly sophisticated with the availability of databases of real eye tracking information from human participants and better computing power, they are used both in engineering visual systems, and to try to understand and mimic exactly towards what a human observer would direct his/her attention[†]. An example of a saliency map is presented in Fig. 2.6.

Itti et al. [1] introduced one of the first prominent saliency model implementations around 20 years ago, derived from the visual attention theory presented by Koch and Ullman [46]. In broad terms, the model is based on extracting color, intensity and orientation cues from an image and iteratively comparing them against each other, as illustrated in Fig. 2.7.

Since then, a lot of models and adaptations have been presented, with both bottom-up and top-down influences (an overview of approaches is illustrated in Fig. 2.8). Borji and Itti [48] provide a recent comprehensive review and evaluation of existing approaches, the review by Bylinskii et al. [49] focuses on comparing

[*] By "looking at", it is implied that the central, high acuity vision is directed towards it.

[†] *Overt attention* requires eye movements to point them in the direction of the object of interest, while *Covert attention* is not directly associated with the object currently looked at.

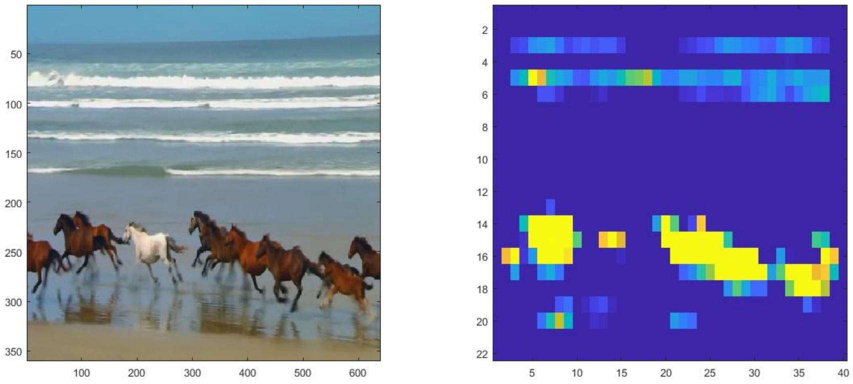


Figure 2.6: *Example of a saliency map, highlighting the elements of an image that would most likely attract the attention of an observer (based only on bottom-up cues). Generated using the Walther and Koch [47] Toolbox.*

models of human eye fixation and in finding shortcomings in current methods. For example, human observers tend to pay attention to parts of an image with text or signs, while current models struggle to mimic this behaviour. Also, the face of a person who is executing an action or seems to be the leader in a group of people are important social indicators [49].

These models have increasingly expanded; Navalpakkam and Itti [50] proposed a goal directed model where relations such as *"is a, includes, part of, contains, similar, related"* are used to determine if a salient object is relevant for a specific task, by checking against relations stored in memory. In a subsequent model, Navalpakkam and Itti [51] used feature maps of different learned objects as a top-down influence in the map; the most likely object detected is selected via an object hierarchy tree. They also explore having memory relationships between elements in a scene, such as a person and a plate of food, determining that the person is eating. An implementation that is a good reference point, is the model by Walther and Koch [47], where they focused on establishing the boundaries of an object deemed as salient.

In a subject more related to robotics, Forssén et al. [53] and Meger et al. [54] present two related publications on how visual saliency and object recognition can be implemented; they use a saliency map to find promising regions of interest and a peripheral camera to have a wide angle view of the scene and a foveal camera to be able to extract detailed characteristics of the objects of interest. Their main goal is to identify correctly as many object categories as possible; they mostly focus on detecting a potential object and moving the robot around it to take more images from different angles, to then identify it by using Scale Invariant Feature

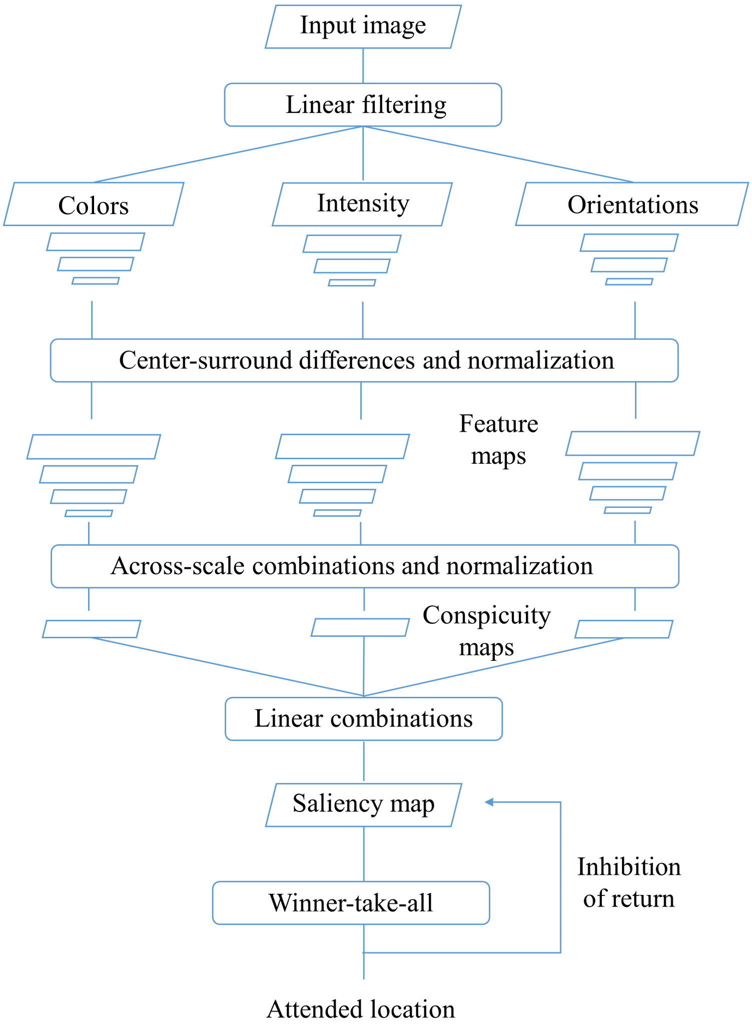


Figure 2.7: *Itti et al. [1] Saliency map model: color, intensity and orientation maps are sub-sampled in Gaussian pyramids and combined with each other until a Winner-takes-all location is assigned as the attended location. Redrawn from [1, Fig. 1]*

Transform (SIFT) features. Since they are publications from 2008, the state of the art in object categorization has changed a lot with the advancements to machine learning, but they serve as a good proof of concept of the mentioned tools. A similar approach is taken by Ekvall et al. [55], who mostly focus on implementing a Simultaneous Localization and Mapping (SLAM) algorithm, integrated with an object recognition one; so that a robot can navigate a new environment and map along the way the location of key objects once they are detected and identified. Rasolzadeh et al. [56] also use a saliency visual-attention system and the foveal and peripheral cameras principle (two peripheral cameras and two foveal cameras, allowing to perceive object depth) in order to detect objects for their manipulation

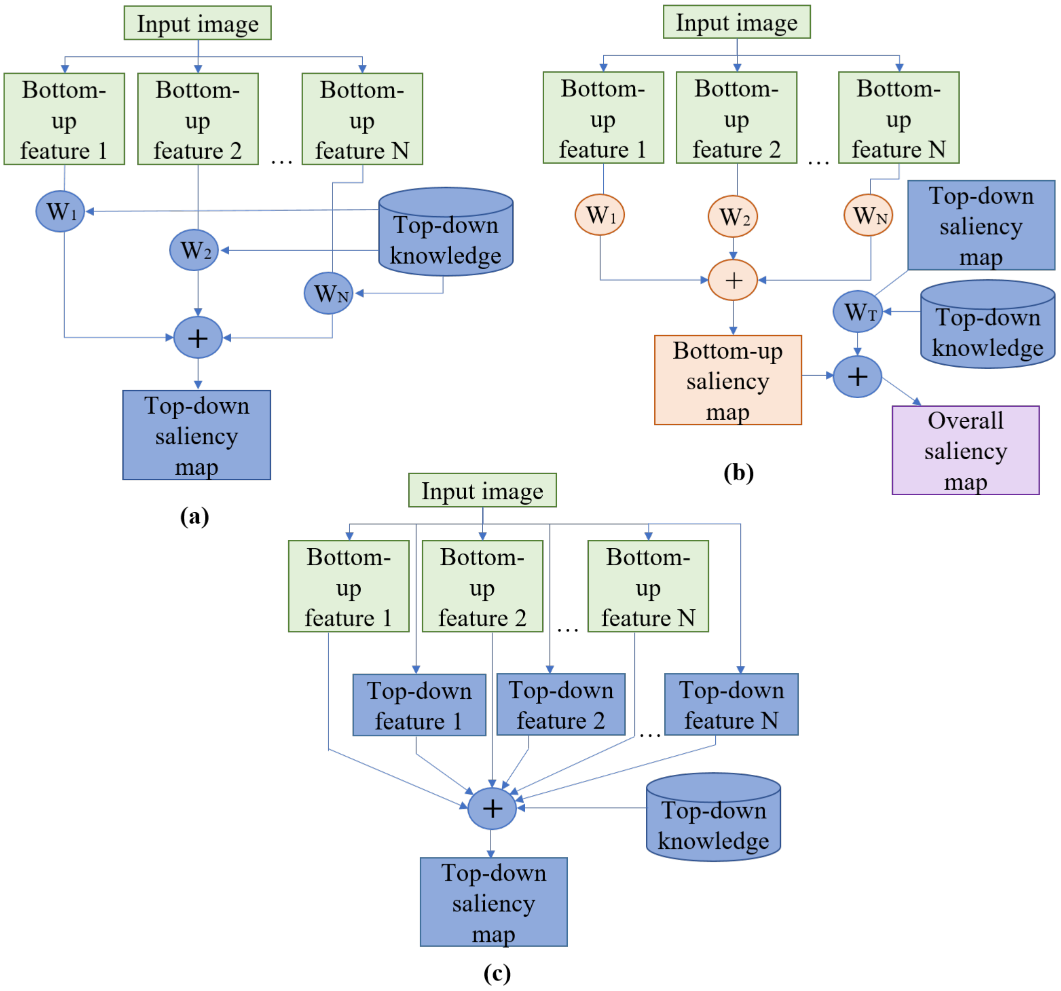


Figure 2.8: *Top-down and bottom-up information merging approaches:* ***(a)*** *Weight modulation of bottom-up features;* ***(b)*** *weighted combination of bottom-up and top-down saliency maps;* ***(c)*** *Joint learning of bottom-up and top-down features. Redrawn from [52, Fig. 9]*

with a robotic arm; picking up an object and moving it to a desired location.

2.6 Salient region and foveation connectivity

The connection between foveation and saliency models is a very logical one, but studies applying both principles in an engineering context are not too numerous. Itti [2] used saliency to blur segments of video files deemed less interesting, taking advantage of specific video format encoding in order to compress them (Similar to [34], but using saliency to have a more complete approach). In more contemporary approaches, foveal transformations have been applied to networks using layers of convolutional operations; Almeida et al. [57] work on the idea of using a forward network pass to obtain class labels and then gaining object proposal masks over a saliency map from a network backward pass. Recasens et al. [58] focus on using saliency as an intermediary layer in a CNN, by altering how to sample the input image from a trained saliency estimation. Foveation is then approached as a mass attraction towards the most relevant pixels.

However, those works require specific CNN retraining and do not address the relation between image size reduction and frame-rate speed-up, which is of critical importance for embedded systems. There is a current gap in studying the speed-up effect of foveated transforms on conventional CNNs used for detection and recognition.

2.7 Object detection with deep neural networks

The field of study of Machine Learning (ML) has advanced at great speed in the last couple of decades, specially in implementations based on the use of interconnected layers of Convolutional Neural Networks (CNNs). They are part of an approach also often referred as deep neural networks given that, during use time, only the input layer (e.g. an image) and the output layer (e.g. bounding boxes of objects intended to detect) are visible and easy to interpret by the user. This structure is also inspired, to a certain extent, in neuronal interconnections present in the human brain.

Such large volumes of new implementations of ML for computer vision tasks have been presented in the last few years that it is difficult to thoroughly encompass the state-of-the-art. Here are introduced some of the basic features and principles, and in subsequent chapters of the thesis describe the most relevant models that were used as they appear. Depending on the type of data, ML approaches are generally built using one of three relevant strategies;

a. **Supervised learning:** This type of algorithm is based on the premise that training data is available containing both inputs and their desired output, from where an optimization loss function can be learned that applies to a wider set of data. This strategy is popular in computer vision tasks with large image datasets, e.g. bounding boxes of objects of interest (cars, persons, etc.) labeled by humans and used as input in the algorithm, which in turn learns to generalize and propose similar boxes in images that it has not previously seen. They can be further divided in classification approaches (identifying the category at which an object pertains) and regression (finding the real value of a variable). Some of the most common models are Support Vector Machines (SVM) and linear regression.

b. **Unsupervised learning:** The desired output for the training data is not available, so the goal is to find patterns in the input data to make decisions or predictions of future inputs. It mainly takes place in one of two forms [59]; *(i) clustering according to its defining properties*, by finding for each data point the group where it is most likely to belong, and *(ii) dimensionality reduction*, for data with a large number of variables or observations, reducing it to a low dimensional space but retaining consequential properties.

c. **Reinforcement learning:** In this strategy, actions are taken by the software agent (i.e. computer program, such as a driver-less car) as a combination of exploring new options and exploiting current knowledge. Successful actions are rewarded and the goal of the agent is to maximize such rewards. Reinforcement learning can also be described by two central strategies of its own [60]; *(i)* to explore the problem space for behaviours that return good performance, and *(ii)* to make a statistical analysis to estimate the reward of each possible action.

In the computer vision research community, the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is often referred as a significant breakthrough, when a CNN model named Alexnet [61] won by a substantial amount, decreasing the classification error rate from around 26% to 15.4% and drawing a lot of attention towards CNNs in general, this model uses 224×224 pixels resolution in its input layer. Even in more recent implementations, such input resolution size is normal for such models; VGG19 [62], another highly influential model presented in the 2014 ILSVRC, uses the same input image size. Remarkably, the more recent Redmon and Farhadi [63] is trained at 448×448 pixels with more than 9000 hierarchical categories, it still can run in real time at an adjustable resolution, trading accuracy against speed. So even with this exceptional progress, it is understandable that such models are not designed for and would struggle with more wide-angle or panoramic views, which becomes increasingly important with the popularity of omnidirectional cameras (with an horizontal field of view of 360°) and aerial perspective videos.

While computing power has steadily become more available in the last few decades, remarkably following the prediction of Moore's law (every two years the number of transistors in an integrated circuit roughly doubles), the demand for such power has also increased, at least partially driven by the market desire for higher resolution photo, video, video-games, etc., requiring a lot of computer processing power. This is also evident in the high computing power demand of state of the art machine learning models and the availability of high resolution digital cameras, it is easy to visualize it in Fig. 2.9; at the turn of the century, VCD was a common format with a resolution of 352 × 240 pixels. In contrast, DCI 4K resolution (with a resolution of 4096 × 2160) has become much more common in the last couple of years.

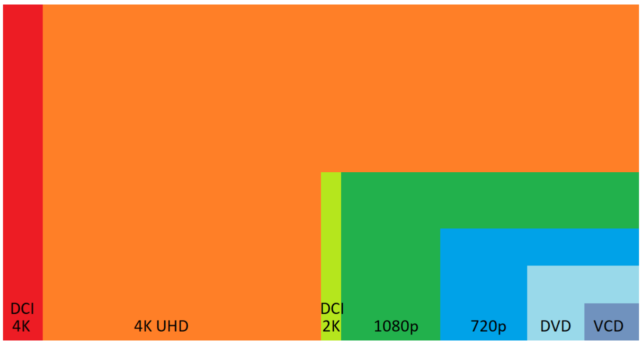


Figure 2.9: *Common digital video resolutions, making it easy to visualize the contrast in the amount of information that needs to be processed, from VCD* (352 × 240 *pixels) to 4K* (4096 × 2160 *pixels).*

Since resources for an embedded system tend to be much less than for a static workstation, a constant need for embedded systems is to minimise computational workload to speed up processing and reduce power consumption. There have been considerable efforts towards developing more compact CNNs for object detection and recognition in embedded systems, which tend to significantly improve frame-rate [64–66].

Computational load in CNN detection-recognition systems can also be reduced by down-scaling the resolution of the input, and of subsequent layers accordingly. The reduction in image size can lead to an increase in computational efficiency due to the reduced number of convolution operations in the CNN, but also tends to trade-off against a decrease in detection and recognition performance. Hence, the challenge is to retain detection and recognition performance whilst using small images. The solution investigated here is based on foveated image transformation, inspired by photoreceptor density in the human eye.

2.8 Overall brain vision system

With the overall goal of making a functional representation of the principal elements of decision making that guide eye movements to foveate in specific locations, and study how it can be useful in robotics, it is opted to create a schematic representation, where different state-of-the-art computer vision elements can be plugged-in and interchanged. Seeking to represent different brain regions based on what are understand to be their key functions, provides interesting insight into how they interact, and the capacity to partially upgrade the system if a breakthrough in the technology of one of them is achieved, also by increasing robustness by separating information channels, thus avoiding that a malfunction in one of them compromises the overall system.

Given the large number of models that are present in the literature aiming to establish a functionality scheme of the brain, it is difficult to present an unanimous agreement of its functionality. The "Two-streams hypothesis" [67] proposes that the brain has two distinct processing pathways for visual and auditory information; the dorsal and ventral streams. The former focuses on where objects are located in space and in action planning [68], often called "where" stream. While the latter is associated with object categorization, also known as "what" stream.

The area of visual cortex that first processes visual information is named V1 (or primary visual cortex), from where it is projected into the ventral and dorsal streams. It has been estimated that a bottom-up saliency map is formed in this area [5, 69, 70], based on *"pyramidal cells, inter-neurons, and horizontal intracortical connections"* [5]. The visual cortex is interconnected with several areas, such as the superior colliculus and the lateral intraparietal cortex (LIP). This last one is associated with evoking saccades on salient locations [68, 71].

One brain region to which is given special attention in this implementation is the basal ganglia, having being considered to accumulate evidence and disinhibiting motor actions [72, 73]. The basal ganglia accumulates evidence from different parts of the cortex, including those associated with eye movements, such as the lateral intraparietal area (LIP) and the frontal eye field (FEF). It is further explored in Section 2.9.

A scheme of brain inter-connectivity is presented in Fig. 2.10, based on current understanding of several studies primarily in the macaque brain, and its functionality in an engineering context. It is taken as a starting point the equivalence between eye movements and the digital foveation of a conventional image, with a similar goal of providing more detail level to a small region of the visual field for object identification. Some of the first visual processing elements in the brain are related to eccentricity, edges, single and double opponent cells [68, 74], similar to early stages of classical saliency maps (Fig. 2.7), which happen around the LGN and primary visual cortex. A more complete saliency map is formed further into the visual cortex. Having identified top salient locations is a key element to plan future eye movements, since they can not all be simultaneously followed, a reliable evidence accumulation is necessary, which also can not be based only in bottom-up cues.

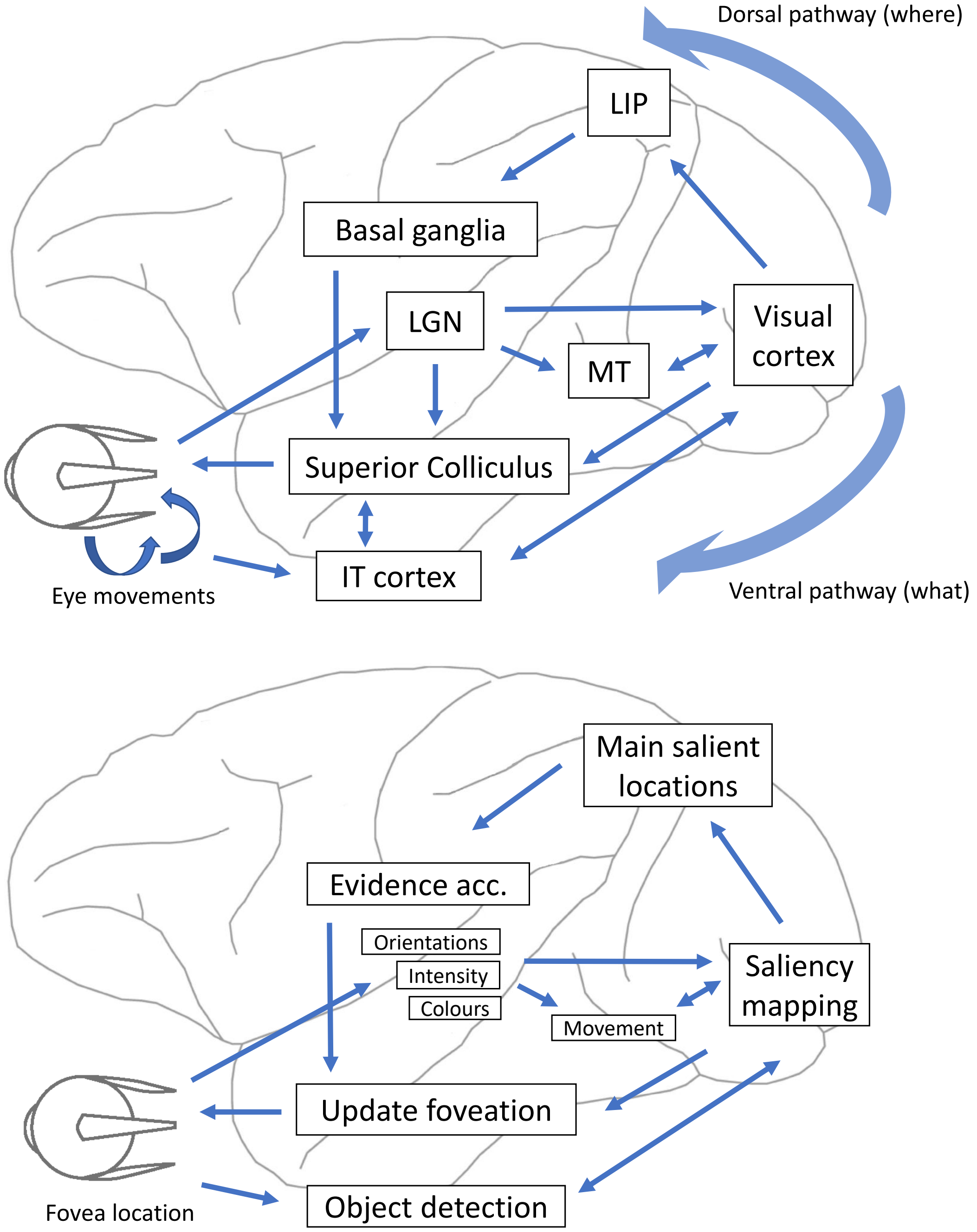


Figure 2.10: *Most of the visual information coming from the eyes retinas projects to an area called Lateral geniculate nucleus (LGN), and then to the visual cortex, from where it feeds the ventral and dorsal streams [68]. Strong centre-surround interactions have been observed in LGN neurons [74], similar to some of the first layers in saliency estimation models, such as the design by Itti et al. [1], illustrated in Fig. 2.7. Scheme drawn gathering information from [68, 72, 74, 75]*

Fig. 2.11 shows a summary of brain areas related to eye and head movements, and Table 2.1 enlist the main elements of the basal ganglia connectivity with the cerebral cortex. Both with the main goal of illustrating the connectivity between the superior colliculus and the basal ganglia. The former allows an animal to orient its eyes through saccades; rapid movements of both eyes to a specific point. The basal ganglia output projects into the superior colliculus and controls the selection of them by suppressing undesired ones [76].

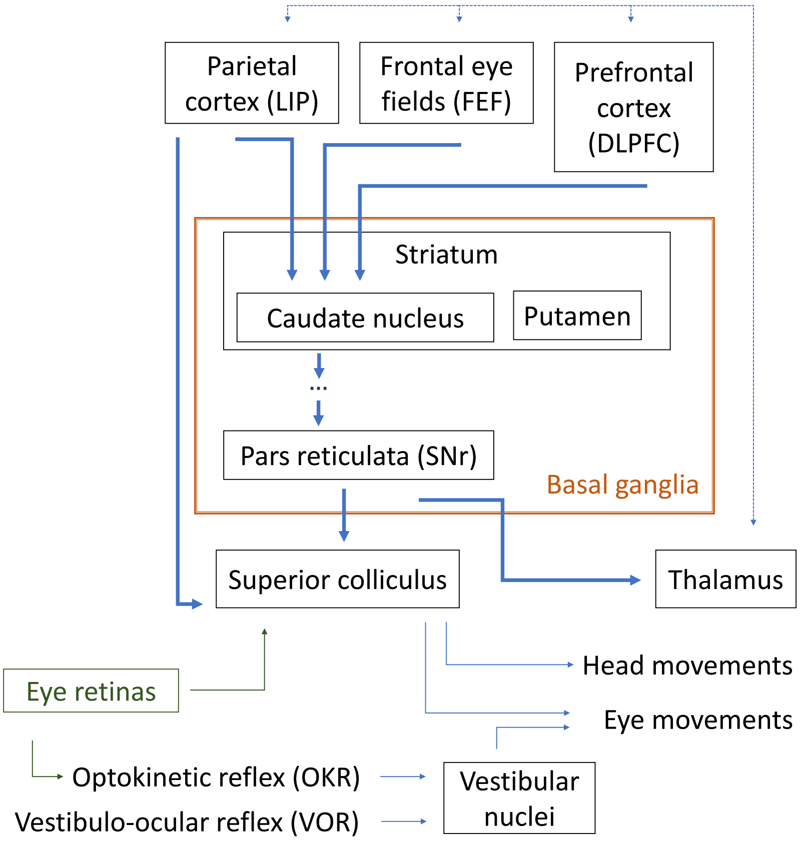


Figure 2.11: *Diagram of the inter-connectivity between the basal ganglia and the superior colliculus. The vestibulo-ocular reflex (VOR) is the response that generates eye movements to compensate for head movements, and thus providing a stabilized retinal input. The optokinetic response (OKR) corresponds to the tracking of a stimulus and corrective saccades when it leaves the visual field. The basal ganglia suppresses undesired saccades and allows appropriate ones to emerge. Assembled from [73, Fig. 1] and [75, Fig. 3].*

Eye movements in the form of saccades are directed by relatively separate cortical areas (such as FEF, LIP and SEF). The basal ganglia does not perform in a similar way, but rather inhibits or boosts such actions when appropriate [75].

Multiple sensorial information; visual, auditory and somatosensory, converges into the superior colliculus, forming a spatial mapping [75, 76]. The middle tem-

Element	Function
Visual cortex	It is the part of the cerebral cortex (the largest and most anterior brain region) in charge of processing visual information [77].
Striatum	Part of the basal ganglia (BG) basic for motor and reward systems. It is the main input to the rest if the BG [78].
External globus pallidus (GPe)	It is the principal regulator of the BG system; the BG works by inhibiting movements that are not required by motor commands by inhibition of parts of the thalamus, adjustments are done via the GPe [79].
Internal globus pallidus (GPi)	One of the output of the BG (apart from the SNr), it inhibits parts of the thalamus to block undesirable movements [80].
Substantia nigra pars reticulata (SNr)	Sends output signals from the BG to the frontal and oculomotor cortex, for the control and stability of eye movements [81].
Subthalamic nucleus (STN)	Important modulator of the output activity of the BG [82].
Premotor cortex	It influences motor behaviour, control and planning using information from other brain regions to select the required movements [83].
Thalamus	Mass of gray matter that connects motor and sensory signals to the cerebral cortex [84].

Table 2.1: *Main elements of the basal ganglia connectivity with the cerebral cortex*

poral area (MT or V5) is related to optic flow and motion estimation [68, 85]. Although it is often associated largely with the dorsal ("where") pathway, it is interconnected with several areas and remains important at various stages of vision processing, including those normally associated with the ventral pathway [86]. A movement channel is added as a path of evidence accumulation, together with bottom-up saliency and top-down detections. Since *"timing analyses place MT/V5 at the first level of the visual hierarchy along with V1"* [86], it is placed it a similar level to where object detection is performed in this implementation.

2.9 Decision making and the basal ganglia

The basal ganglia is a part of the cerebrum composed of several clusters of neurons (or nuclei). It has been extensively studied and related to an important range of brain functions, such as action selection, exploratory behavior, motor preparation, sequence learning and reinforcement learning [87]. Since the motor system is the main output of the nervous system [88], a versatile action selection mechanism is needed. It has been proposed that the basal ganglia fulfills this role [72, 73], meaning that it is capable of activating a winning path of motor action from competing possible actions that require access to the motor resources [87]. In it, the substantia nigra pars reticulata (SNr) is a part of the basal ganglia that mainly serves as output and provides inhibition on presaccadic neurons of the superior colliculus, allowing saccades to take place [75].

A robust mechanism to select the statistically best next action is required for all types of engineering systems, particularly autonomous robots, since inaction or delaying a decision also incurs an implicit cost, optimal stopping is a key factor. *Sequential Probability Ratio Test* (SPRT), described in Section 2.10, proves to be a good approach in this sense; evidence is collected for different hypothesis until one of them crosses a set threshold. For example, Chung and Burdick [89] formulate a search task as a decision making one, using SPRT to verify that a target is present in a given segment of the search space. They show how SPRT can be a flexible decision making tool, where the critical criteria can be the rate of change, the confidence thresholds, the time limit for making a decision, among others.

This hypothesis has also been tested in robotic implementations a few times; Montes-González et al. [90] designed a model intended to illustrate how a clean and coherent switching between activities can be achieved, by mimicking a mouse enclosed in a small environment. It shifts between tasks like exploring the environment, grabbing food, taking it back to its niche, and staying close to the walls, while its "motivation" changes over time, as an illustration of how hunger and fear are also dependent on the time that the rodent has spent in the new environment.

In [91], the previous theory is further developed by presenting a more detailed model of the basal ganglia. It is a good illustration of how it is more effective in choosing a given action, in comparison to a Winner-Takes-All mechanism intended to always select the most salient input, since the former takes into account the history of past actions and anticipates the reward of future ones to select the more appropriate behavior at a given time. Prescott et al. [91] also make the point that the model helps to keep the winning actions active while they last by making them more salient, instead of "locking" them in the actuators or having a priority level for every action, this permits to have a more effective interruption of the current task if a sudden highly salient event occurs.

In another robotic implementation with similar perspective, Girard et al. [92, 93] used a Lego Mindstorm robot to test the basal ganglia model designed by

Gurney et al. [94, 95], again with the goal of testing the action switching behavior in a more complex environment, having the robot move around in a small enclosure in which certain actions allow it to find "food", recover energy, or spend the remaining energy finding the next food source, having the time that the robot "survives" in the environment depend on it. A key issue of this publication is comparing said basal ganglia model to a conventional Winner-Takes-All. In this instance, they did not find a huge difference in success between the two, except in the sense that the basal ganglia model is better at adjusting, depending on the environmental circumstances, to when it is more pertinent to constantly go back and forth between two salient actions (such as an animal eating while being vigilant in a dangerous area), or when focusing in a single task is safer or more energy efficient (e.g. looking at the road while driving).

2.10 Sequential Probability Ratio Test

Sequential Probability Ratio Test (SPRT) is a statistical tool for testing hypothesis without fixing a specific sample size. It was originally presented by Wald [96] and one of its advantages is being able to reach a decision (among the provided hypothesis) when enough evidence is accumulated, even if all the samples have not been taken into account; they are computed sequentially and each step a decision is taken among (for two hypothesis):

- Hypothesis H_A: There is enough evidence to claim that $\theta = \theta_A$
- Hypothesis H_B: where $\theta = \theta_B$ is selected and the process ends
- Ask for further evidence: Compute the next sample

Then the two type of errors that can occur are clear, [97]:

- α = P(Selecting $\theta = \theta_A$, when $\theta = \theta_B$ was correct)
- β = P(Selecting $\theta = \theta_B$, when $\theta = \theta_A$ was correct)

The log-likelihood ratio, for a given sample, is normally defined as;

$$\Lambda_n(X) = \log \frac{L(\theta = \theta_A | X_1, ..., X_n)}{L(\theta = \theta_B | X_1, ..., X_n)}$$

Given the desired error margins α and β, the desired boundaries can be established;

$$A = \log \frac{\beta}{1-\alpha} \quad \text{and} \quad B = \log \frac{1-\beta}{\alpha}$$

So that for the latest sample, if $\Lambda_n(X) \leq A$ we reach conclusion A. If $\Lambda_n(X) \geq B$ we conclude on B, and for the region were $A < \Lambda_n(X) < B$, it is deemed that there is not enough evidence yet.

Baum and Veeravalli [98] developed an expansion of the SPRT model to test multiple hypothesis (hence the name MSPRT). Here for M hypotheses, a stopping observation N_A is defined as the first sample (from $n \geq 1$) for which the posterior probability (for one or more samples k);

$$p_k(n) = P(\theta = \theta_k | X_1, ..., X_n) > \frac{1}{1 + A_k}$$

The winning hypothesis, $\theta = \theta_m$, is the one where $m = \arg\max_j p_j(N_A)$.

Bogacz and Gurney [72] pose the same problem as a decision threshold, which should provoke a decision to be made when the logarithm of the accumulated evidence for a given hypothesis surpasses it. If y_i is the evidence that supports $\theta = \theta_i$ (referred as salience in [72]) and $S(n)$ is the accumulated data of a specific hypothesis;

$$L(n) = \ln(p_i(n)) = y_i(n) - S(n) = y_i(n) - \ln \sum_{k=1}^{M} \exp(y_k(n))$$

We can arrive at this equation given the Bayes' theorem, since $p_i(n) = P(\theta = \theta_i | X_1, ..., X_n)$;

$$p_i(n) = \frac{P(X_1, ..., X_n | \theta = \theta_i) P(\theta = \theta_i)}{P(X_1, ..., X_n)}$$

Assuming that the different channels can not be true at the same time,

$$P(X_1, ..., X_n) = \sum_{k=1}^{M} P(X_1, ..., X_n \wedge \theta = \theta_k) = \sum_{k=1}^{M} P(X_1, ..., X_n | \theta = \theta_k) P(\theta = \theta_k)$$

$$p_i(n) = \frac{P(X_1, ..., X_n | \theta = \theta_i) P(\theta = \theta_i)}{\sum_{k=1}^{M} P(X_1, ..., X_n | \theta = \theta_k) P(\theta = \theta_k)} = \frac{P(X_1, ..., X_n | \theta = \theta_i)}{\sum_{k=1}^{M} P(X_1, ..., X_n | \theta = \theta_k)}$$

$$L(n) = \ln(p_i(n)) = \ln(P(X_1, ..., X_n | \theta = \theta_i)) - \ln(\sum_{k=1}^{M} P(X_1, ..., X_n | \theta = \theta_k))$$

Furthermore, Lepora [99] present how learning can be described as the adaptation of the threshold boundary in the SPRT model, using either a learning rule commonly used for Neural Networks (REINFORCE method) or Bayesian Optimization, with the goal of maximizing the reward function.

2.11 Summary and gaps in the literature

Huge efforts and advancements have been made to integrate robotic systems in our everyday lives. Since the world is an unstructured and unpredictable environment [100], algorithms need to be flexible and reliable. Since human observers are largely adaptable and flexible, robust research has also been dedicated to study how such systems work in the brain of humans and other animals.

There are interesting contrasts in the progress of some of the fields that have been briefly covered. For example, it has been appreciated since early days in computer vision that there must be a benefit on the structure into which human vision has evolved through a very long time period. As a result, very diverse studies of artificial foveation have consistently been presented. In contrast, artificial neural networks have seen "waves" of interest and success since their first proposition more than 70 years ago [101], with a very steep acceleration in the last decade, partially enabled by the ability of Graphic Processing Units (GPU) and their huge market demand. This last one have a (more lenient) base in the structure of the human brain, but have the constraint of being opaque in their functioning, opening the question of how these two approaches can benefit from each other. There is also a lacking of contemporary examples that place the two approaches together in a wider information flow context, while also opening the question of whether the performance measurement metrics most commonly used are actually the most insightful.

The performance of saliency models has often been measured against human fixation datasets as ground truth, mostly without including a time element, making them not best suited for applications in robotics, and with the open question of where they fit in a wider context of biological vision.

It is important to state that saliency maps are not the only feature extraction technique, but one that fits well with the modular representation of brain vision functions (Fig. 2.10). Considerable research in feature descriptors has been done in the last few decades with the goal of identifying key points or features of an image to categorize it, match it with different views or angles, or identifying its containing elements. Despite their intrinsic motivation being different, there is some overlap between these techniques and those underlying saliency maps; saliency maps attempt to find a small number of regions in an image that are deemed the most important, i.e. where a person would most likely shift their attention. Image descriptors such as SIFT tend to find key feature points in the order of thousands, distributed all around the image (e.g. a 500 × 500 pixel image might give around 2000 stable features [102]).

In general, there has being a renewed drive in understanding the brain connectivity in a bottom-up approach, with great advances in *(i)* Spiking Neural Networks, to mimic the exact curve of single neuron activations, and *(ii)* purpose-built hardware, e.g. SpiNNaker [25] and DVS [18]. But there is also the opportunity to gain new information from a top-down modular approach, making rough approximations to what tasks some brain regions are understood to do, and how suitable they are when computing power is substantially limited, in similar fashion to how vision processing is constrained in the human brain.

2.12 Discussion

Very diverse subjects have been introduced in this chapter, some of them relatively briefly, given the vastness of studies that exist in their corresponding research areas. It is easy to see how some principles of brain functionality, such as reproducing foveation in digital images, have invited research for a long time, even if the underlying technological constraint that they are aimed to tackle has greatly evolved over the last few decades. For example, image compression was motivated by transfer bandwidth limitation, e.g. [2, 34]. Now, the reemergence of neural networks in computer vision applications has pushed both software and hardware advances.

GPUs arose as a great fit for the application of object detection systems using deep CNNs. They are now a central point of interest, since modern neural network models can conceptually be made indeterminately large. A main restricting factor is the number of Floating Point Operations (FLOPs) that fit into a given GPU for it to run at an acceptable speed, which is even more relevant in embedded systems. In this chapter, the main concepts and state-of-the-art have been established to aim to tackle this relatively new problem using image foveation.

An interesting principle that has been highlighted is modularity. Given the complexity of the human brain, a lot of current understanding of it comes from studies of specific brain regions and functions, for example decision making. The ability to perform human vision studies in healthy participants and without any alteration to it is also what makes it one of the best understood brain systems. This modularity is also taken as an engineering principle, with several advantages presented through the thesis; *(i)* it allows for weak points of each module (saliency, foveation, object detection, decision making) to be counteracted by the bordering ones, *(ii)* it also facilitates plugging in and out new implementations of each of these modules (replacing an object detection CNN by another one) as the state-of-the-art advances or according to hardware limitations. *(iii)* Rather than doing one overall model, e.g. exclusively a deep neural network that encompasses some of the principles (like [57, 58]), better insight into each section of it can be obtained through modularity.

Chapter 3

Foveated vision for deep neural network object detection and recognition

3.1 Introduction

The aim of this chapter is to establish how detection, recognition and processing speed in a CNN are affected by reducing the input image size, by using a foveated transformation. The intention is to demonstrate that downsizing the convolutional layers in a CNN, enabled by foveating the input image, provides a considerable speed-up in processing while retaining high performance in detection and recognition in the foveal region, and reasonable performance in the periphery.

A key barrier to deploying the latest object recognition systems on embedded platforms is the computational burden placed on Graphic Processing Units (GPUs), resulting in low frame rates. Whilst more powerful computing becomes available over time, CNN models have also become persistently more complex. So the problem to allow state-of-the-art models to be run on embedded systems will continue to be present. The overall goal of this chapter is to study the extend of computational speed-up that can be obtained by down-scaling images, based on the foveal-peripheral transformation observed in the human retina.

As had been previously described, the fovea accounts for around half of the photo-receptors, and the other half is distributed along the rest of the retina, this presents a very contrasting process to how digital images are recorded, where pixels represent the colour information of fixed-size field of view regions.

The benefit of giving sensorial priority to regions more likely to contain contextually relevant information is evident and an idea that has been previously explored, this becomes even more relevant in embedded systems, such as robots, where computing power and energy are limited.

Even if such a foveal representation of digital images has been previously studied, with the fast-paced advances in machine learning and computer vision in the last few years. From the current state-of-the-art, it is not clear yet how feasible it is to use foveal images for detection and location tasks in convolutional neural networks, when the goal is to lessen the size of the network to maximize the rate at which images can be processed.

Considerable effort has been put into the development and improvement of neural networks, for tasks such as detection and localization, arguably more so after 2012, when the model AlexNet [61] achieved a substantial (around 10%) performance increase in the ImageNet Large Scale Visual Recognition Challenge [103]. Given the heavy computational requirements of current neural network algorithms, the frame processing speed is often a required compromise in low-energy embedded systems, like the Nvidia Jetson series.

3.2 Methods

3.2.1 Image Database: Microsoft COCO

The Microsoft COCO dataset [104] has become a standard benchmark for testing algorithms aimed at scene understanding and pixel-wise segmentation, providing a rich array of relatively context-free images. It was chosen for training and evaluating the object detection CNNs here, where the images were re-sampled at increasingly smaller sizes using a foveated transformation, and used to retrain the YoloV3 CNN for object detection and recognition [105]. This foveated approach was compared to linearly downsampled images to analyse the benefit of the foveated transform. To evaluate processing speed, YoloV3 was implemented on an NVIDIA Jetson TX2 GPU for embedded systems. In addition, as a comparison against a different type of object detection and recognition system, Faster R-CNN [106], was used with the foveated images to analyse performance and also compared to the system without being retrained in foveated images.

To make the retraining of the CNNs more manageable, a subset of the COCO dataset was used, considering the first 20 listed objects*. This dataset was chosen due to the aim of testing the hypothesis in cluttered scenarios in which the ground truth objects are not necessarily centred, having around 3.5 categories and 7.7 instances per image [104].

*person, bicycle, car, motorbike, aeroplane, bus, train, truck, boat, traffic light, fire hydrant, stop-sign, parking meter, bench, bird, cat, dog, horse, sheep, cow.

The foveated transform was applied separately to every object for each image in the COCO database. Therefore, each original image from the COCO database spawned multiple foveated versions of the image, each with the fovea centred on a different object. This increased the number of training images from around 82,000 to 306,000. To test and compare performance of uniform image sampling versus foveated image sampling, at different image sizes, the image sizes were varied from 416 × 416 to 96 × 96, at intervals of 32 pixels*. The upper limit of the image size, 416 × 416, corresponds to a typical size for running an object detection and recognition algorithm such as YoloV3.

3.2.2 Foveal-peripheral image resampling

Photoreceptor density and cortical magnification factors have been well studied in the biological domain [26, 27, 107, 108], which has also then been applied to design computational representations. From a computational point of view, several different methods have been developed to transform a standard digital image, with uniform sampling, into a foveated representation. These include the log-polar transform [29], the reciprocal wedge-transform [30] and Cartesian foveated geometry [31].

There is no general consensus in the literature on a best foveated image sampling approach, since it depends on the desired goal, e.g. biomimetic model representation or information reduction. Since here, the main interest is on resource optimization, the resulting images should be of a fixed size, to match that of the CNN. Previous methods generally create a "blurred" image of the same size to the original showing the effect of foveation, often in a non-rectangular fashion.

It also seems less logical to use blurring or leaving "blank" pixels to represent foveation, pixels that could be used to provide additional information as input to the CNN. The method used here was based on an approach of Cartesian log-spaced sampling, which captures the key feature by densely sampling the fovea and compressing the periphery. This method was found to be effective, and because it distorts the original uniformly-sampled image less than, e.g. a log-polar transform, it gives the additional key benefit of enabling the use of transfer learning to speed-up the training of the CNNs (i.e. initialising the CNN weights using a network pre-trained on uniformly sampled images).

The approach here proposed is based on the idea to re-sample any uniform digital image of size $N_x \times N_y$ pixels, to a new size of $n_x \times n_y$ pixels with log-spacing, so that for the upper right quadrant of the image with the fovea centred

*96 × 96, 128 × 128, 160 × 160, 192 × 192, 224 × 224, 256 × 256, 288 × 288, 320 × 320, 352 × 352, 384 × 384, 416 × 416 on (x_0, y_0) the sample locations are,

$$x_k = \exp(k\Delta_x) \quad \text{for } k = 0, \dots, n_x/2 \tag{3.1}$$

$$y_k = \exp(k\Delta_y) \quad \text{for } k = 0, \dots, n_y/2 \tag{3.2}$$

where

$$\Delta_x = 2n_x^{-1} \log(N_x/2) \tag{3.3}$$

$$\Delta_y = 2n_y^{-1} \log(N_y/2) \tag{3.4}$$

The mapping of the retina here presented is similar to the concept of *Message Sending Unit (MSU)* [108], making a parallel to pixels in digital images, together with their appraisal of MSUs in the different regions of the retina. Their estimation of the relation between data fields and eccentricity angle in the human eye is partially reproduced in Table 3.1.

Eccentricity	0.5	1	2	5	10	30	45	60	70	90
Data fields	256	552	848	1239	1534	2003	2176	2299	2365	2472

Table 3.1: *Relationship between the eccentricity angle in the eye and the number of data fields [108], where they represent retinal regions over which stimulus is collected in cell sub-assemblies from thousands of input fibers and overall properties are calculated over them. Data reproduced from [108].*

Given that the COCO dataset has images of different shapes and resolutions, foveated image resampling is performed using an iterative algorithm, to reach the desired exact pixel resolution. Whilst this slows the algorithm down, it is not required for an embedded system where input images are of constant size. It is only used here due to the nature of the varying image sizes in the COCO database. Otherwise, a look-up table can be created to store the pixel indices for every possible location of the fovea, which bypasses the need of calculating them online.

The proposed algorithm starts from the centre of the foveated object in the image and generates a logarithmically spaced vector towards each of the four borders of the image, initialized with a conservative number of elements and increasing until it reaches the target size. This step is required since the logarithmically spaced vector needs to be rounded to integer numbers, to reflect the index of rows and columns of the images, while also eliminating the duplicate indices (which accounts for the uncertainty of the number of iterations). The algorithm is described in pseudo-code in Fig. 3.1 and exemplified in Fig. 3.2.

Algorithm 1: Foveated Image Re-sampling

Input: Uniformly sampled image size, N; Centre of fovea, c; Desired foveated image size, n

Output: Vector of sample indices S

```
while length($S$) $< n$ do
    # Sample columns to the left of the fovea
    $L = \text{ceil}(((n + j)c) + c + 1)$
    $\Delta_L = (\log c)/(L - 1)$
    for $k = 0, ..., L - 1$ do
        $x_{L,k} = \text{ceil}(-\exp(k\Delta_L))$
    end for
    $x_L = \text{unique}(x_L)$

    # Sample columns to the right of the fovea
    $R = \text{ceil}((n + c)(1 - c) + c - 1)$
    $\Delta_R = (\log(N - c))/(R - 1)$
    for $k = 0, ..., R - 1$ do
        $x_{R,k} = \text{ceil}(\exp(k\Delta_R))$
    end for
    $x_R = \text{unique}(x_R)$

    # Concatenate $x_L$ and $x_R$
    S = unique($[x_L, x_R]$)

    # Increment counter
    $j = j + 1$
end while
```

Figure 3.1: *Algorithm for generating sample indices in the x- or y-directions to transform an arbitrarily sized, uniformly sampled, digital image to a foveated one. The function* ceil *rounds-up to an integer and the function* unique *ensures non-repeated indices. This algorithm is used separately for both the x- and y-directions in the image to select row and column samples.*

3.2.3 Object recognition

Given the time and computational requirements for training a network in a vast dataset such as COCO, it proves prohibitively long to exhaustively test on all new architectures that are added to the literature, specially at the speed in which it evolves. Two implementations representative from the state-of-the-art are selected for used given their success, specially in terms of computational speed, and their general strategies; while Faster R-CNN [106] has a dedicated convolutional Region Proposal Network (RPN), YoloV3 [105] is a regression model which penalizes the

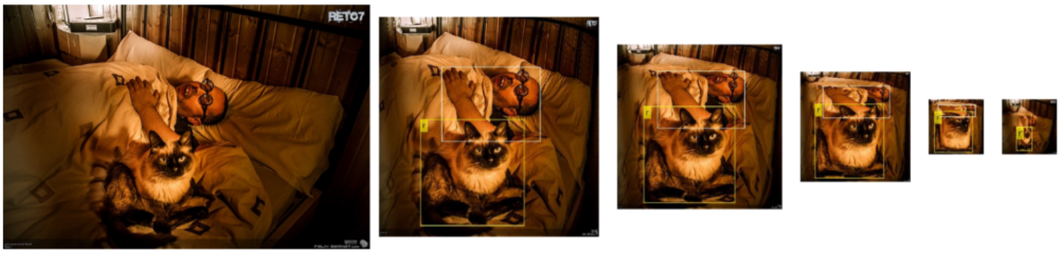


Figure 3.2: Example of typical image from the COCO dataset used for validation, from left to right, at its original resolution of 640 × 426 *pixels, at* 384 × 384*,* 288 × 288*,* 192 × 192*, and* 96 × 96*, and finally the same image normally downsampled to* 96 × 96 *pixels for comparison.*

sum of errors for inaccurate bounding boxes, the of the cell, object confidence and class prediction. These two models [105, 106] are briefly introduced in the following two subsections:

3.2.3.1 You Only Look Once, version 3 (YoloV3)

The *You Only Look Once, Version 3 (YoloV3)* [105] object detection system includes 75 convolutional layers designed with successive blocks, where each block is composed of a 1 × 1 convolutional layer, followed by a 3 × 3 convolutional layer, and a residual layer. Blocks are repeated numerous times with occasional shortcut connections, followed by average pooling then a fully connected layer with softmax output, as illustrated in Fig. 3.3 (a). With the *loss function*:

$$\begin{aligned} L = {} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_{i}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \end{aligned} \tag{3.5}$$

Where $\mathbb{1}_{i}^{obj}$ checks if the object is in cell i and $\mathbb{1}_{ij}^{obj}$ if the j bounding box corresponds to the predictor cell i [109]. The first pair of sums analyse the dimensions of the object and the bounding box in terms of its (x, y), width and height (w, h), while the following ones do it in accordance to its class label. The loss function for each of these variables is defined as the sum-of-squared error. The class label prediction is done for the objects contained in each bounding box using multi-label classification, which is trained using a binary cross-entropy loss function. YoloV3 uses dimension clusters as anchor boxes to predict the object bounding boxes along with the class label [105].

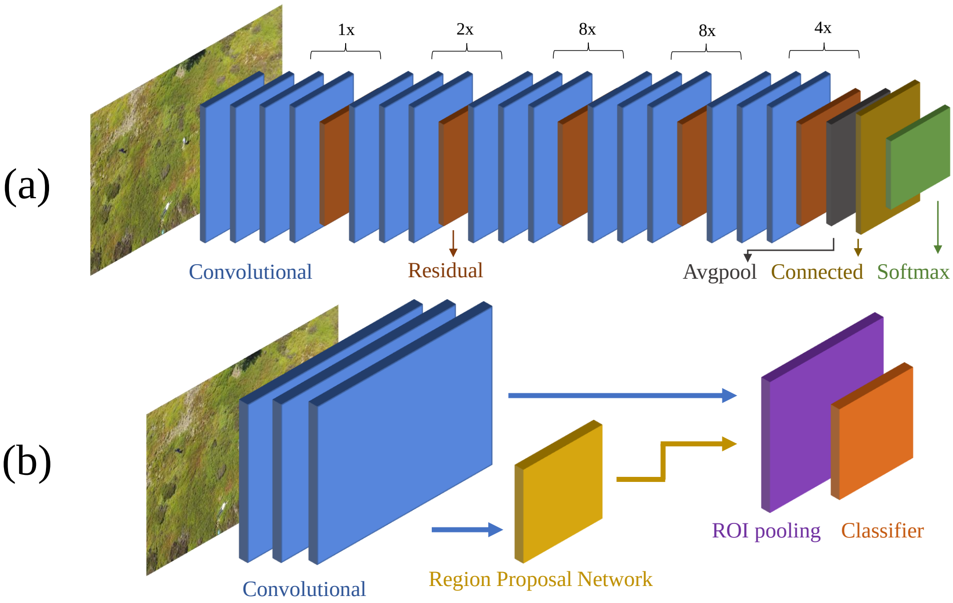


Figure 3.3: ***(a)*** *YoloV3, a fully convolutional model with 106 layers, that makes single shot detections at 3 different scales,* ***(b)*** *Faster-RCNN, a classification model using anchors and a dedicated network for region proposals.*

Training was performed on an NVIDIA GeForce GTX 1070 GPU (with 1,920 Pascal CUDA cores), using 306,000 images (from the COCO database), with a batch size of 32, and subdivision of 16, given the computing capacity of the GPU. Other Training parameters were used as defined by the original authors [105]; stochastic gradient descent with momentum was used as the training algorithm, with learning rate of 0.001, momentum of 0.9 and weight decay of 0.0005. Network weights were initialised using the pre-trained network obtained from [110].

Given the size of the dataset, training time was left for as long as possible, overfitting being less of a concern and monitoring the loss function over time. The training process was iterated for 200,000 steps ($\sim 9,500$ iterations per epoch, i.e. ~ 20 epochs), repeating the entire process for the 11 image sizes, where each CNN was restructured to match the size of the input images. Testing was done on a reserved validation data set of 6000 images. Frame-rate was evaluated by processing all test images and averaging the result on an NVIDIA Jetson TX2 board, with a 256-core Pascal GPU, using CUDA Toolkit 8.0 and cuDNN 5.1, with the images saved on internal memory but ignoring the time required to load them.

3.2.3.2 Faster R-CNN

Faster R-CNN [106] is an object detection and recognition system that uses a region proposal network (RPN) to generate regions for object detection and recognition by subsequent convolutional layers [106]. Since YoloV3 and Faster R-CNN are based on different principles in creating region proposal boxes, it is interesting to see how they react to foveated images; in YoloV3, the input is divided into a grid cell, where each cell is responsible for the object centred in it [109], while Faster R-CNN used the RPN based on fully convolutional layers, which are shared with the object detection network [106], also meaning that the RPN can be trained for each application. Training is done with the *loss function*:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*) \tag{3.6}$$

Where p_i is the probability of there being an object in anchor i, while p_i^* is a binary ground truth of it being the case or not [106]. The first half of the equation is the classification loss of *object/no object*. The second half, the regression loss, is only activated for positive anchors (given $p_i^* = 1$).

A general scheme of Faster R-CNN is drawn in Fig. 3.3 (b), a considerable focus of its innovation was the way in which region proposals are generated with, by default, 9 anchor boxes for every position of the sliding window, which has shown good results in the used datasets. The Region Proposal Network is trained with a loss function evaluating the probability of a box either being an object or not [106]. It is particularly efficient because the RPN shares convolutional features with the detection/recognition CNN. In this section, Faster-RCNN is used trained in the 80 object categories of the COCO dataset, at high resolution, as a comparison to YoloV3 for processing foveated images. The specific implementation used is the current version is as developed by the original authors [111].

3.3 Analysis and Evaluation

To evaluate performance and the effect of image and network size, conventional precision and recall metrics [112] are used,

$$\text{Precision} = \frac{\sum_{i=1}^{n}\text{TP}_i}{\sum_{i=1}^{n}(\text{TP}_i+\text{FP}_i)} \quad \text{and} \quad \text{Recall} = \frac{\sum_{i=1}^{n}\text{TP}_i}{\sum_{i=1}^{n}(\text{TP}_i+\text{FN}_i)} \tag{3.7}$$

where TP are the true positives, FP are false positives, and FN are the false negatives. A true positive is only counted if the CNN predicts the correct class label *and* the object location, as measured by an Intersection over Union (IoU) value (Fig. 3.4), when it is over a threshold, here set to 0.5 as in [105, 109], a ratio of the area of overlap and the area of union between the prediction and the ground-truth. It is important to differentiate that, while it is assumed that the fovea is centred in the object, object localization still needs to be evaluated as passing the bounding box IoU threshold. Performance is evaluated separately in the fovea and the fovea-periphery (to explicitly quantify performance in the foveal region where detection-recognition should be accurate, and in the periphery where accuracy is expected to decrease).

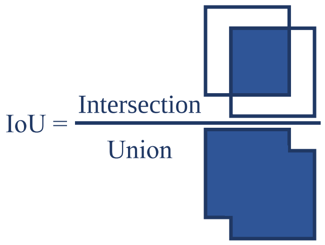


Figure 3.4: *Intersection over Union (IoU) of two bounding boxes*

3.3.1 Floating point operations

Although computing an averaged frame-rate in low-energy consumption GPUs is a very illustrative way to show speed-up in deep neural networks, as resolution is decreased, a more general approach is counting the number of Floating Point Operations (FLOPs) that the network requires. The number of FLOPs for YoloV3, for resolutions from 416 × 416 to 96 × 96, retrained for 20 categories of the COCO dataset is presented in Table 3.2, which are calculated for each layer as [110];

$$\text{giga-FLOPs} = \frac{2.0 * l_n * l_{size} * l_{size} * l_c * l_{out_h} * l_{out_w}}{1 \times 10^9} \tag{3.8}$$

Given l_n as the number of filters, l_{size} as the filter size, l_c is the number of channels and l_{out_h}, l_{out_w} are the output height and width respectively. It is divided by 1×10^9 to express it in terms of billions of FLOPs. The 2.0 multiplying factor derives from the fact that two different types of operations (multiplication and accumulation) take place in each computer clock. To illustrate the much larger number of operations in other implementations, Table 3.2 gives the number of FLOPs required for implementing Faster-RCNN using Resnet-50 ([113], a neural network with 50 convolutional layers), Resnet-101 (with 101 convolutional layers) and Inception [114] for feature extraction, making it less suitable for embedded

GPUs, while performance is examined in Section 3.3.3 for Faster R-CNN Inception.

Resolution	**YoloV3** (giga-FLOPs)	**Faster R-CNN** (giga-FLOPs)		
		Inception	**Resnet-50**	**Resnet-101**
416 × 416	65.426	152.68	192.98	511.61
384 × 384	55.75	150.55	186.11	500.95
352 × 352	46.852	148.59	179.78	491.13
320 × 320	38.728	146.8	174.01	482.17
288 × 288	31.344	145.19	168.78	474.07
256 × 256	24.776	143.74	164.11	466.82
224 × 224	18.944	142.46	159.98	460.42
192 × 192	13.955	141.36	156.41	454.87
160 × 160	9.675	140.42	153.38	450.18
128 × 128	6.2	139.65	150.9	446.34
96 × 96	3.473	139.06	148.98	443.35

Table 3.2: *Floating point operations (in billions) for YoloV3 and Faster R-CNN, with several feature extraction approaches, illustrating how the former decreases the number of operations more drastically and is overall better suited for operation in portable GPUs.*

3.3.2 Results on YoloV3 with re-training

The foveal analysis performed in this section assumes that a saliency step has already been performed that crudely aligns the fovea with a point of interest. The CNN still has to detect the object precisely, in terms of the bounding box and its label.

In Fig. 3.5, the top graph highlights the assertion that, with the object in the fovea, the performance rate can be kept relatively constant (35.20% at 416 × 416 to 32.38% at 128 × 128). This behaviour is contrasted, in the right-Y axis, with the average frame-rates that were achieved in a Jetson TX2 board. At 416 × 416 pixels, the framerate on the Jetson TX2 was just 3.59 FPS. In the top right, the increase in precision for using foveal images is notable. This boost can be expected since the object of interest occupies a big portion of the scene, but most remarkably the algorithm learns to make less false positive predictions. The bottom right quadrant shows the performance for all objects, with an accuracy decrease notably similar to that observed for normal downsampling.

The recall at size 128 × 128 using foveated images decreased only slightly to 32.38% (92.0% of the baseline result) but for uniformly sampled images decreased to 17.33% (50.1% of the baseline result) - Table 3.4. The key additional point is that frame rate increased to 15.24 FPS at an image size of 128 × 128 - this is over a 4× speed-up.

Size	Fov-rec	Norm-rec	Fov-prec	Nor-prec	FPS, 20-obj	FPS, 80obj
416 × 416	35.20	34.57	25.09	24.45	3.59	3.31
384 × 384	35.27	33.75	25.66	24.09	4.29	4.02
352 × 352	35.03	33.15	25.51	23.85	4.75	4.53
320 × 320	34.30	31.89	25.36	23.14	5.21	4.87
288 × 288	34.83	30.82	26.47	22.42	5.76	5.34
256 × 256	33.98	29.29	25.95	21.72	8.52	8.04
224 × 224	34.30	27.60	27.58	20.19	9.40	8.91
192 × 192	35.33	25.07	29.43	18.71	10.76	10.36
160 × 160	33.62	20.30	30.70	17.41	12.14	11.70
128 × 128	32.38	17.33	34.70	15.15	15.24	14.63
96 × 96	23.32	6.42	30.80	8.31	16.65	15.75

Table 3.3: *Comparison of the precision and recall (%) for only the object centred in the fovea, with image and network at varying resolutions, from* 416 × 416 *to* 96 × 96*, along with the frame-rate average on a Jetson TX2 board.*

Size	416 × 416	192 × 192	160 × 160	128 × 128	96 × 96
Foveal	1	1.003	0.955	0.919	0.663
Normal	0.982	0.712	0.577	0.492	0.182
Speed-up	1	2.997	3.381	4.245	4.637

Table 3.4: *Recall for objects in the fovea, proportional to the performance at* 416 × 416*, with foveal and normal downsamples, along with the speed-up for detection of 20 objects on a Jetson TX2.*

Interestingly, the precision performance increased for the foveated images as the image size was reduced, but decreased for the uniformly sampled images (Fig. 3.5 and Table 3.3). The increase in precision performance for foveated images is evidently a benefit of the fact that the object of interest takes up more of the visual scene, reducing the false positives.

The foveal-peripheral recall performance was similar to the foveal-only performance at the largest image size, 416 × 416 pixels, for foveated images (35.20%) and uniformly sampled images (34.57%) (Fig. 3.5). As image size was reduced to 128 × 128 pixels, recall performance for both foveal and uniformly sampled images decreased significantly, to 34.3% of baseline for foveated images and 45.3% of baseline for uniform images (Fig. 3.5 and Table 3.5). The decrease in precision is less pronounced across the same range (to about ∼30% for both image types), indicating that precision is less sensitive to the reduction in image size in the periphery. The precision-recall curve for the smaller networks is shown in Fig. 3.7.

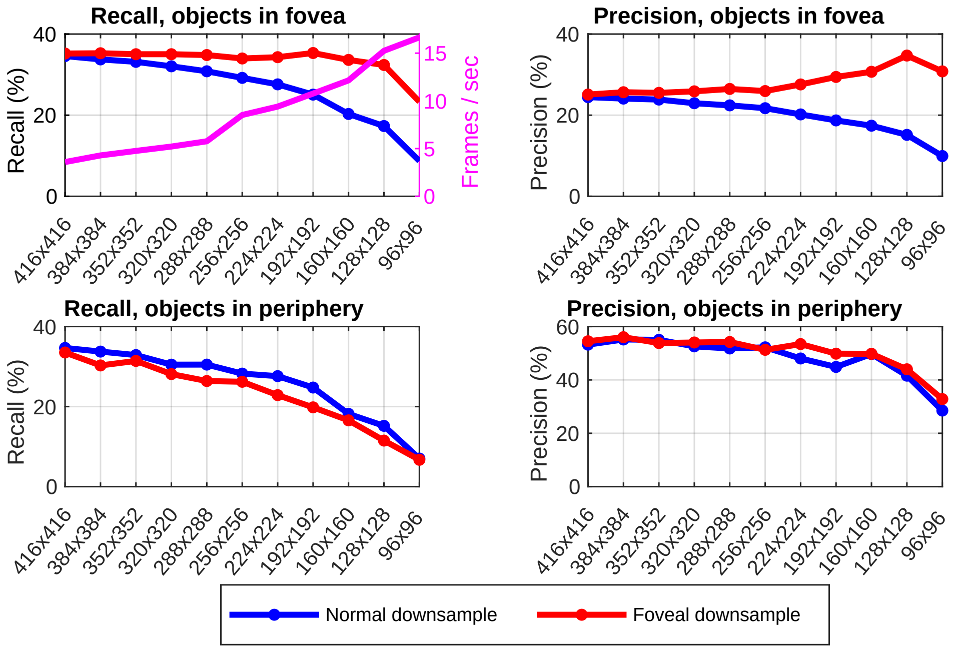


Figure 3.5: Recall and precision performance for the YoloV3 network trained at different resolutions. Top row: objects in the fovea. Bottom row: objects in the periphery. Top left also shows the average frame rate for processing each image size.

Size	416 × 416	192 × 192	160 × 160	128 × 128	96 × 96
Foveal	1	0.591	0.494	0.343	0.200
Normal	1.034	0.739	0.543	0.453	0.157

Table 3.5: Recall comparison for objects in the periphery, proportional to the performance at 416 × 416*.*

3.3.3 Comparison between Faster R-CNN and YoloV3

To corroborate the previous observations, Faster R-CNN was also tested with the foveated images and uniformly sampled images at varying sizes. Due to the lengthy training process, retraining was avoided for Faster R-CNN, and to provide a consistent comparison, in this section YoloV3 was also tested without resampled retraining. Both networks were used with all 80 object classes from their original versions. The behaviour was remarkably similar to that obtained in the previous sections, for both YoloV3 and Faster R-CNN. (Fig. 3.8). A 4× speed-up in frame rate was still observed for YoloV3, from 3.31 FPS to 14.63 FPS at 416 × 416 and 128 × 128 respectively (Fig. 3.8). This serves as some confirmation that the approach of using foveated images with reduced size is beneficial to wider CNN designs used in object detection and recognition. These results also provide evidence that the advantages of foveation in object detection are not simply due to

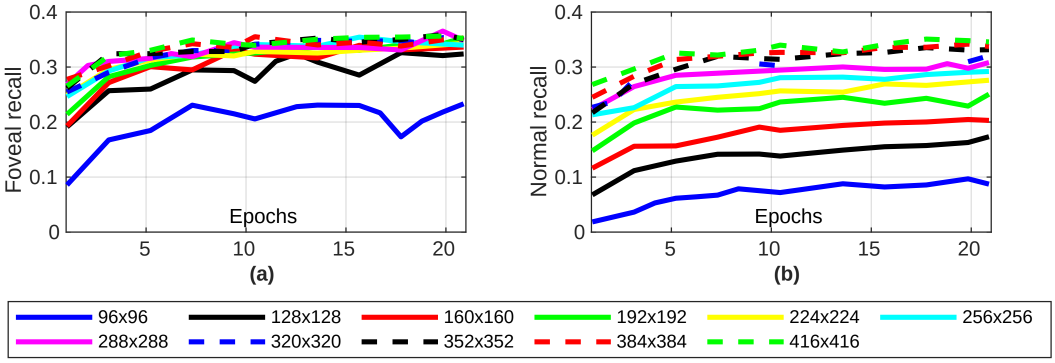


*Figure 3.6: **(a)** Recall performance evolution for foveated images, through different epochs of training. **(b)** Recall performance evolution for normally downsampled images after 20 epochs.*

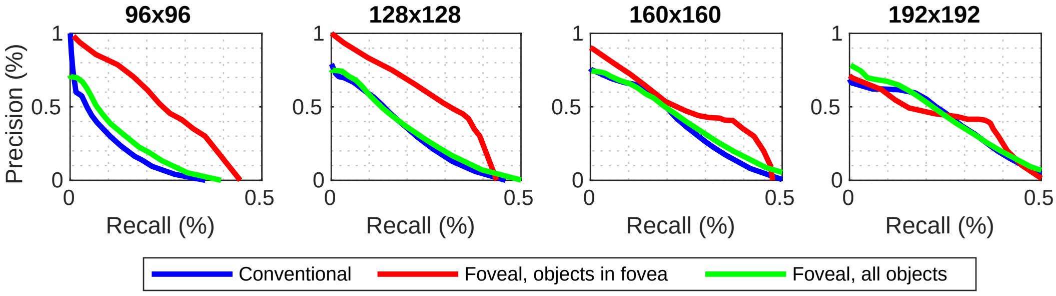


Figure 3.7: Precision and recall performance curves for the network at small resolutions (96 × 96, 128 × 128, 160 × 160 and 192 × 192). The foveal advantage is much more evident for the smaller networks, where the speed-up is also larger. In all cases, the performance is very similar between the normal downsample and the average of all objects present in the foveated image. For the larger networks, the prospect of detecting objects in the periphery increases, which affects the precision measurements when only the foveal object is considered as a true positive.

an effect of detecting image distortion due to the foveated transform itself.

3.4 Discussion

The motivation for this part of the study was to make object detection and recognition with CNNs more efficient for embedded GPU systems, with the aim to investigate quantitatively how detection, recognition and processing speed in a CNN were affected by reducing image size using a foveated transformation. These results show that images can be reduced in size from 416 × 416 to 128 × 128 pixels, with only a small decrease in recall, 8.0%, using foveated sampling. A limitation of the approach was the decrease in object detection and recognition in the periphery, which should be expected given the downsampling of pixels.

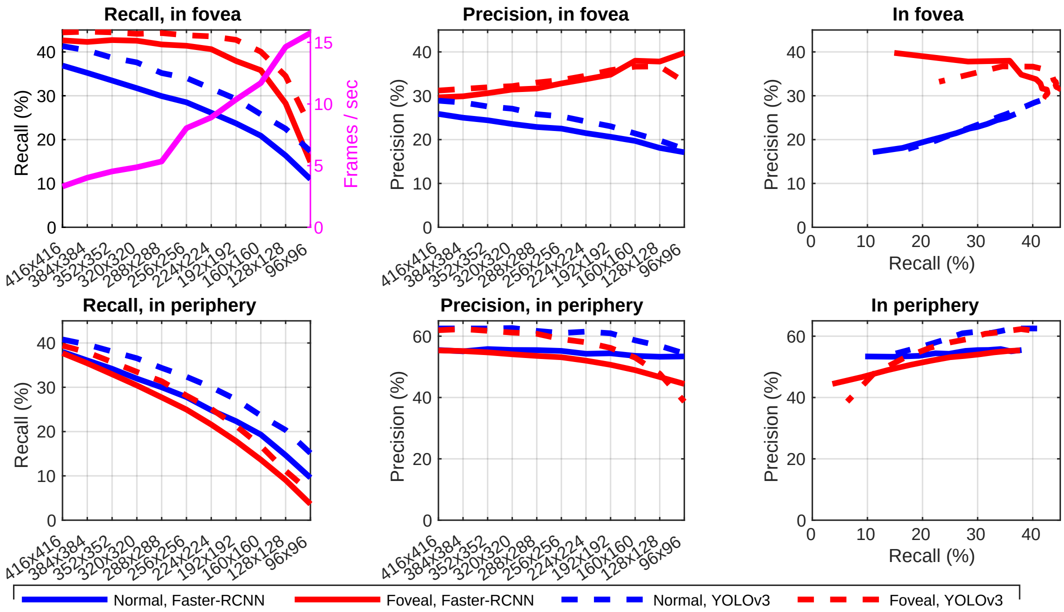


Figure 3.8: Comparison of performance changes using the un-retrained YoloV3 and un-retrained Faster R-CNN neural networks as they were trained by their initial contributors on 80 object classes. Note that frames per second in the top panel is for YoloV3 only.

The key benefit observed is in terms of processing speed-up for reduced size images, specifically a 4× speed-up with 128 × 128 pixel images. The increase in processing speed observed is advantageous for future embedded systems: in the short term embedded systems with limited GPU processing power can more readily exploit the latest advanced algorithms, whilst in the long term as GPUs advance, less resource will be needed for object detection and recognition, maximising resources and energy efficiency.

Having the capacity to process four times as many frames, exemplified with the speed up from 3.59 to 15.24 frames/second (with YoloV3 networks of size 416 × 416 and 128 × 128 respectively), increases the chances not to miss important objects for an autonomous robot navigating an unstructured environment. With the help of other sensors for navigation, video feedback is well suited to be in charge of high level scene understanding. Having the certitude of an object identified at high resolution is potentially more valuable than several with low confidence, so a comparison can be made between examining subsequent frames at low resolution or for reconstructed images with the fovea centred in a different location each.

Mechanisms to represent a foveal transformation have been studied for several decades [29], while it proves difficult to arrive at an unified model, implementations commonly arrive at similar drawbacks, such as being indeterminate in the of the fovea for log-polar representations, a technique commonly used [29]. While here the aim is to use a mechanism as efficient as possible, intended for small portable GPUs, it would be unsound to leave pixels of the final image unused, either on the of the fovea, or at the borders, due to the intrinsic discrepancy between square images and polar representations. A compromise is then made in terms of biological adherence, by using the full row/column selected and thus arriving at square images.

While using a foveal transformation still leaves the important problem of where to focus the fovea, and how to coherently switch its location in the time domain, this has the potential to be tackled with saliency methods [57, 58]. The drawback is either making the assumption that *(a)* the objects which the network has being trained to detect are inherently salient on an image or *(b)* having to train or handmake a saliency detector for every new application (or new environment). Non task specific saliency methods can still be capable of steering attention towards certain image elements, e.g. [115], but this depends on their inherent properties.

Considering how fast the human visual system allows us to interact with our environment (e.g. scan it to locate a specific object, find food, or detect an immediate threat), a robust system is needed to regulate this behaviour, by answering the question *"Where to look next?"*. The stimuli that drives us to look at something are often classified as either *bottom-up* or *top-down*, the former makes reference to purely visual stimuli (such as a bright colour, an odd shape or a sudden movement), the latter is directed by the current task; e.g. while driving, it is paramount to be *looking at* the road most of the time, even if it has an uninteresting shape.

Chapter 4

Top-down and bottom-up visual saliency with deep CNNs

The human eye has a very small area with a high density of photo-receptors, the fovea, which captures with high acuity a visual field area of around twice the size of a thumbnail at arm's length [3], and yet producing about half of the visual input to the brain [4].

4.1 Introduction

The previous chapter explored how a foveated image as input to a state-of-the-art Convolutional Neural Network (CNN) can be used to reduce its number of computations, opening the question on how to select the region of the image on which to foveate. To address said question, in this chapter a visual saliency approach is presented, based on biological principles aimed at selecting main regions of interest, so that they can be foveated into a down-sampled image. This foveated image will be processed by a CNN, as previously detailed.

Itti et al. [1] proposed a popular bottom-up engineering saliency map model, derived from the visual attention theory presented by Koch and Ullman [46], a more recent implementation of which is used here - Vocus2 [116]. This model is based on extracting colour, intensity and orientation cues from an image and iteratively comparing them against each other. This bottom-up saliency fuses with a top-down saliency path based on feedback from the CNN.

Recently, deep neural networks have been used to learn visual saliency with foveated vision in an end-to-end scheme [57]. Here, a more modular approach is taken, with the goal to more closely mimic biological structures [52, Fig. 9].

4.2 Methods

The visual information flow here proposed is illustrated in Fig. 4.1, where foveal images are used as input to a CNN, allowing to target small sections of the scene, with high object detection confidence and at high frame-rate.

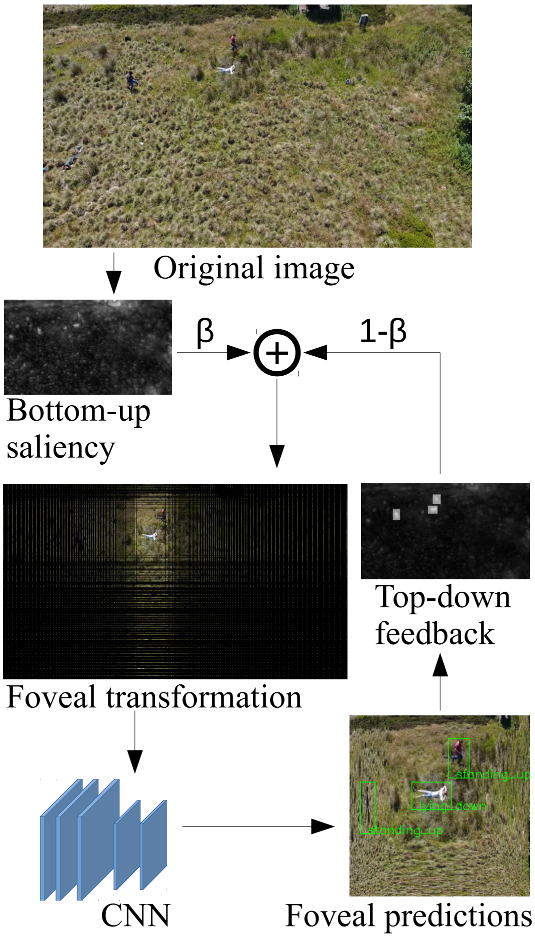


Figure 4.1: Diagram of how the bottom-up saliency orientates the top location for the foveal transformation, in its turn feeding the deep neural network. The predictions of the CNN are used as feedback to supplement the saliency computation, with a variable magnitude β, for the frame at $t+1$.

By calculating the bottom-up saliency on a normally down-sampled current frame I_o, by a magnitude m into I_D, $m = 4$ is experimentally found to be a good compromise between speed and resolution. Two main advantages are obtained; (a) given the pyramidal saliency model structure (Fig. 4.3), a smaller image will be processed considerably faster, and (b) the chance is decreased that an one-off pixel with high saliency will be marked as the most salient one. With I_D as input, the bottom-up saliency map, S_{BU}, is obtained, providing the top salient location, which is used as the centre of the fovea, f_c. Then the frame I_o is transformed into a squared foveal image I_f, of equal length and width, to match the input size of the CNN. The calculations required to foveate are done a priori, avoiding a bottleneck, since once the foveation step is required, it can be done by simply consulting a Look-up-table (LUT), with the coordinates of the desired rows and columns.

The LUT containing the pixel coordinates required to transform the input image is created following the process described in the previous chapter. In this case, starting with the original frames at 848×480 pixels, a popular resolution with a close to 16:9 ratio. Foveal images are tested at increasingly smaller images, from 416×416 to 96×96 pixels, at 32 pixel intervals. The neural network input layer size is made to match the foveated image size.

The bounding box predictions of the CNN can then be transformed back into the coordinates at the same size to I_D, and used as a top-down saliency influence, S_{TD}, in the next frame, at $t+1$, allowing to obtain an overall saliency,

$$S_O = \beta \cdot S_{BU} + (1 - \beta) \cdot \gamma \cdot S_{TD} \tag{4.1}$$

Where β is an influencing factor to control the magnitude in which the top-down information is considered. γ controls the priority level of any given class of the CNN detection. For example, with $c = 5$ categories present in the data-set, giving a priority to the third one, $\gamma = [0.5, 0.5, 1, 0.5, 0.5]$,

$$\gamma \cdot S_{TD} = \sum_{n=1}^{c} \gamma_{[n]} \cdot s_n \tag{4.2}$$

where s_n are the normalized predicted bounding boxes for the n^{th} category.

4.2.1 Bottom-up saliency

A well established saliency algorithm was chosen to compute bottom-up saliency, Vocus2 [116], for several reasons: this model is closely structured to the original Itti et al. [1] proposal, but with slight improvements to make it competitive with contemporary more computationally heavy approaches. It uses difference of Gaussian at several scales, as a representation of ganglion cells in the human retina [116]. It also provides a pixel level saliency map, which is necessary to establish the location of the fovea. A general scheme of its structure is presented in Fig. 4.2, similar to the original Itti et al. [1] model in the use of Gaussian pyramids of the main features; colours, intensity and orientations (Fig. 2.7). A partial example of the intermediary layers is also shown in Fig. 4.3.

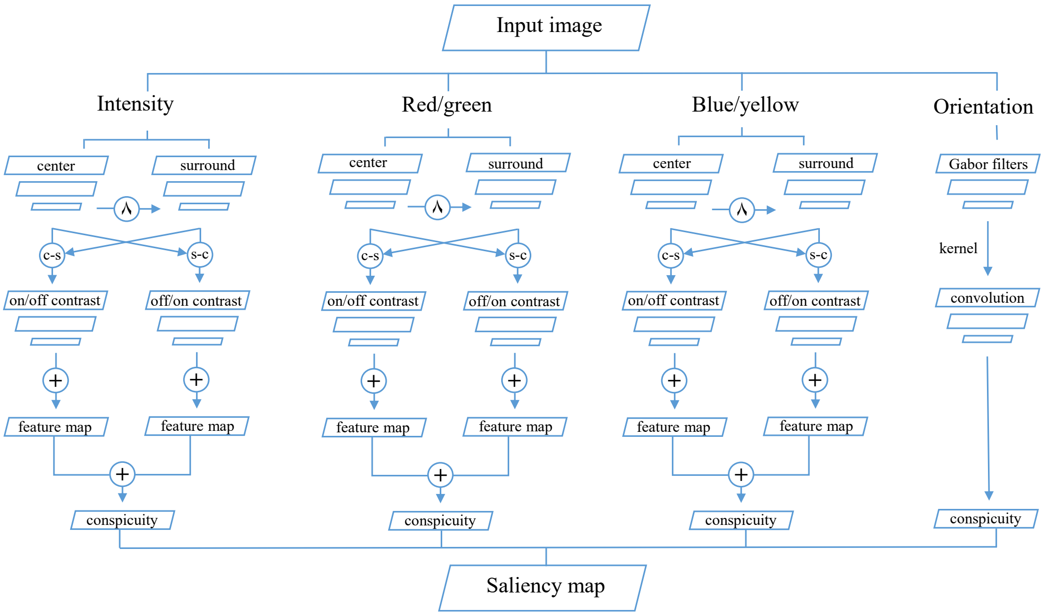


Figure 4.2: *Vocus2 bottom-up saliency diagram, showing its twin Gaussian pyramid approach, for intensity, red/green and blue/yellow contrast, as well as the use of Gabor filters for orientation. Adapted and redrawn from its original source [116].*

The structure is based in pyramids of feature channels, f, of Intensity I, red-green RG and blue-yellow BY colour contrasts, and orientations O.

$$I = \frac{1}{3}(R + G + B) \tag{4.3}$$

$$RG = R - G \tag{4.4}$$

$$BY = B - \frac{R + G}{2} \tag{4.5}$$

$$O = g(x, y, \lambda, \theta, \psi, \sigma, \gamma) \tag{4.6}$$

Centre C^f and surround S^f pyramids are computed with a Gaussian blur for each of the above channels f, which then provide on-off, X_i^f, and off-on, Y_i^f, contrast pyramids respectively (for each layer i);

$$X_i^f = C_i^f - S_i^f \quad \text{and} \quad Y_i^f = S_i^f - C_i^f \tag{4.7}$$

Feature maps are obtained as the arithmetic mean across layers i;

$$F_1^f = \frac{1}{n}\sum_{i=1}^{n} X_i \quad \text{and} \quad F_2^f = \frac{1}{n}\sum_{i=1}^{n} Y_i \tag{4.8}$$

That in turn produces conspicuity maps for each feature $C^f = f(F_1^f, F_2^f)$, where f is also applied as an arithmetic mean in this case. The orientations channel performs a Gabor filter function at the different pyramid levels as defined by;

$$\begin{aligned} g(x,y,\lambda,\theta,\psi,\sigma,\gamma) &= \exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda}+\psi\right) \\ x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta \end{aligned} \tag{4.9}$$

With orientation of the normal to the parallel stripes $\theta = \{0, \frac{\pi}{4}, \frac{2\pi}{4}, \frac{3\pi}{4}\}$. Standard deviation of the Gaussian envelope $\sigma = 2$, wavelength of the sinusoidal factor $\lambda = 0.5$ and phase offset $\psi = 2\pi$. Given the pyramidal structure of this saliency estimation method, a smaller image and limiting the number of layers provides a considerable speed-up in the saliency map estimation; one scale pyramid and and four layer levels were used, with centre pyramid smoothing factor $\sigma_C = 1$ and surround pyramid smoothing factor $\sigma_S = 2$. The final saliency map is obtained as the mean, f, of the conspicuity maps:

$$S_{BU} = f(C^I, C^{RG}, C^{BY}, C^O) \tag{4.10}$$

4.2.2 Other saliency approaches

Evidently, different types of approaches have been developed over the years for saliency estimation. As effectively summarised by Borji et al. [117], they can mainly be split into; *(a)* the early computational models derived from the Itti et al. [1] framework, *(b)* those focused on binary segmentation of the salient objects in an image, and *(c)* those formed of deep convolutional neural networks. The first group, in which Vocus2 [116] is based, fits best the overall goal of this project, since it is more focused on eye fixation and brain mechanisms. Nonetheless, it is worth briefly describing contrasting approaches; Cheng et al. [118] segment an image in N regions $\{r_i\}_{i=1}^N$, where for region r_i, saliency is defined as;

$$s(r_i) = \sum_{j=1}^{N} w_{ij} D_r(r_i, r_j) \tag{4.11}$$

given a $D_r(r_i, r_j)$ contrast appearance between two regions and a w_{ij} region size and distance weight between the two, thus being more suited for regional object masking.

Saliency can also be approached in terms of the contrast between an image I_k

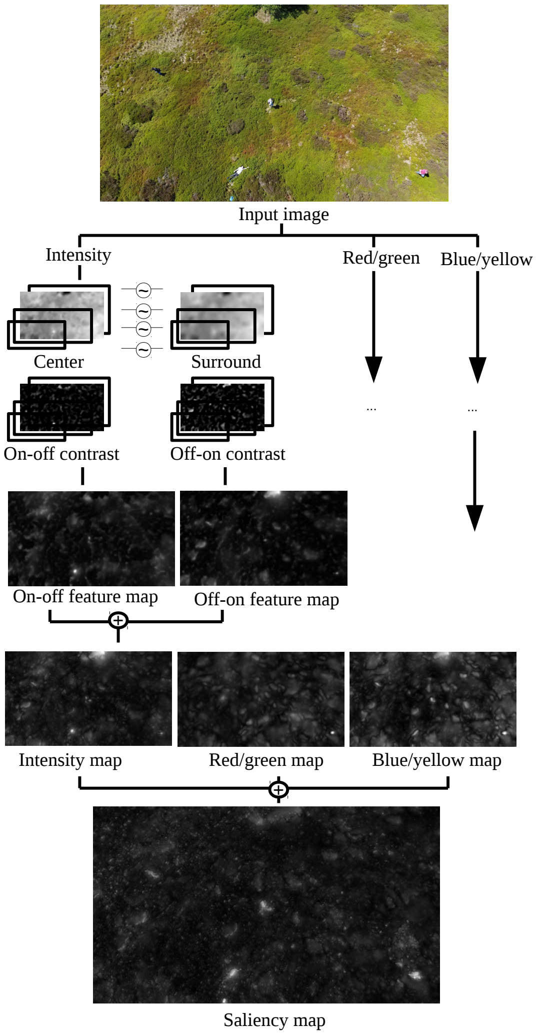


Figure 4.3: *Partial example of the Bottom-up Vocus2 [116] saliency model used, with one scale pyramid, feature and conspicuity fussing by arithmetic mean, two centre surround pyramids and four layer levels.*

and other visually similar K images, in cases where I_k has salient and non-salient descriptor annotations as a Fisher vector, respectively $(f^+_{I_k}, f^-_{I_k})$, so that patches p_{I_k} can be compared to the descriptors of the other K images; $S(p_I) = \{(f^+_{I_k}, f^-_{I_k})\}_{k=1}^{K}$. Evidently, this requires the availability of such descriptors, which can not be taken for granted in an open environment application and requires large collections of images [117].

Given, the diversity of saliency estimation models, it is appealing to use aggregation techniques to gain a better overall model. Borji et al. [119] propose that for M saliency map estimations $\{S_i\}_{i=1}^{M}$, with $S_i(x)$ saliency for the x pixel, in the saliency map i. A summation of saliency maps can be obtained as:

$$S(x) = P(s_x = 1|f_x) \propto \frac{1}{Z}\sum_{i=1}^{M}\zeta(S_i(x)) \tag{4.12}$$
$$f_x = (S_1(x), ..., S_M(x))$$

With ζ following an either identity function $\zeta(z) = z$, exponential $\zeta(z) = \exp(z)$, or logarithmic $\zeta(z) = -\frac{1}{\log(z)}$.

Since the overall goal is not to maximise a saliency score for a subjective dataset, but to provide clues for regions of foveation, the use of a single agile model is the best approach.

4.2.3 Foveal pre-processing in Look-up-tables

Given the need in this application to obtain a foveal transformation promptly, it is more suitable to calculate beforehand the indexes of the required rows and columns to transform the image in terms of its location. Such approach is possible with the compromise of the biomimetic principle where foveal images are not square, nonetheless the squareness is a desired property for embedded robotics and CNN objects detection, both to match the size of the CNN and avoid the wastefulness of leaving unassigned pixels. A logarithmic method to calculate the distancing between pixel indexes is used here, as described in the previous chapter, due to its closeness to the foveal behaviour [120]. Having a desired output size $n \times n$, the selection of rows and columns can be done independently.

It is not easy to predict the number of useful indexes (integer non-repetitive numbers) from a logarithmic generated array. The final size varies by deleting the repetitions after rounding: if 10 logarithmically spaced indexes between 1 and 30 are needed (to transform a 30 × 30 pixel image to a 10 × 10 foveal one), they would be; [1, 1.459, 2.129, 3.107, 4.534, 6.616, 9.655, 14.089, 20.559, 30]. However, of these only 9 indexes are usable; [1, 2, 3, 5, 7, 10, 14, 21, 30]. So in this case, 11 initial numbers provide the 10 desired indexes; [1, 2, 3, 4, 5, 8, 11, 15, 21, 30].

This process is done for each possible location for the most salient pixel, based on the size of the saliency map in which it is located, and create a text based Look-up-table (LUT), which is then quickly accessed with the desired index when a foveation is required.

4.3 Experimental data: Unmanned Aerial Vehicles point of view

Visual sensory data is notoriously expensive to process, and for a robot moving in an unstructured environment, the relevance of a region in the field of view can drastically change from one moment to another, even more so in the case of Unmanned Aerial Vehicles (UAVs), since the regions directly above them are key in specific scenarios (e.g. while landing or flying indoors).

UAVs represent a rapidly growing area of research, which although is not as frequently associated with biologically-inspired principles, presents the most crucial need for power-efficient processing, since more computing resources also means more mass weight to be carried, increased battery consumption and decreased flying time. UAV image type data was used in the subsequent chapters, both for this reason and to illustrate a more application oriented approach.

Saliency also proves to be naturally beneficial for aerial images given their inherent wide-view nature, an idea that has been explored in cases like [121–123]. Doherty and Rudol [123] focuses on search and rescue operations, also a natural fit operation for UAVs, given their capacity to cover wide areas. It is opted to use a combination of the collected data for this project and a previously available dataset. It is hypothesized that the model should be able to distinguish considerably visually different actions performed by a person; "standing up" and "lying down". This would be very beneficial for a drone surveying a natural area with several dispersed people present, with only one requiring assistance or special attention for search and rescue operations.

Video footage was taken using a *DJI Phantom 4 Pro* drone in a meadow region of the Peak District, United Kingdom. Using five participants either walking around in random directions or lying down, with them switching between the two categories previously described. For training, 4,804 frames were manually labelled, from 4 different scenes, and extended to 24,020 using the *Imgaug* image augmentation library [124], with transformations including Gaussian blur and noise, contrast normalization, rotation and flipping.

For better generalization, the collected data was merged with a subset of the Stanford drone dataset [125], following a few conditions to balance the number of occurrence from each category, given a predominant count of appearances of "pedestrians", which was fused with the "person standing up" label. Similarly, the "golf cart" and "car" categories were considered as one. From 18 separate videos, in 6 different scenes, the frames that contained bus were considered, with a total count of "pedestrian/person standing up"; 84,201, "biker"; 57,280, "golf cart/car";

16,040, and "bus"; 12,006. Fig. 4.4 shows an example frame at 848 × 480 pixels resolution, and foveated to 416 × 416, 288 × 288 and 160 × 160 pixels respectively, from left to right.

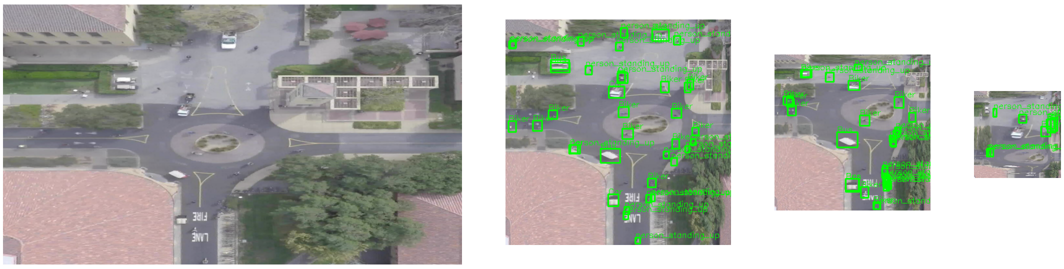


Figure 4.4: *Frame from the Stanford drone dataset, at* 848 × 480 *pixels, and downsampled by the described foveation approach to* 416 × 416, 288 × 288 *and* 160 × 160*, from left to right. As the image size decreases, less objects are present in the fovea, but they are able to maintain a resolution similar to the original.*

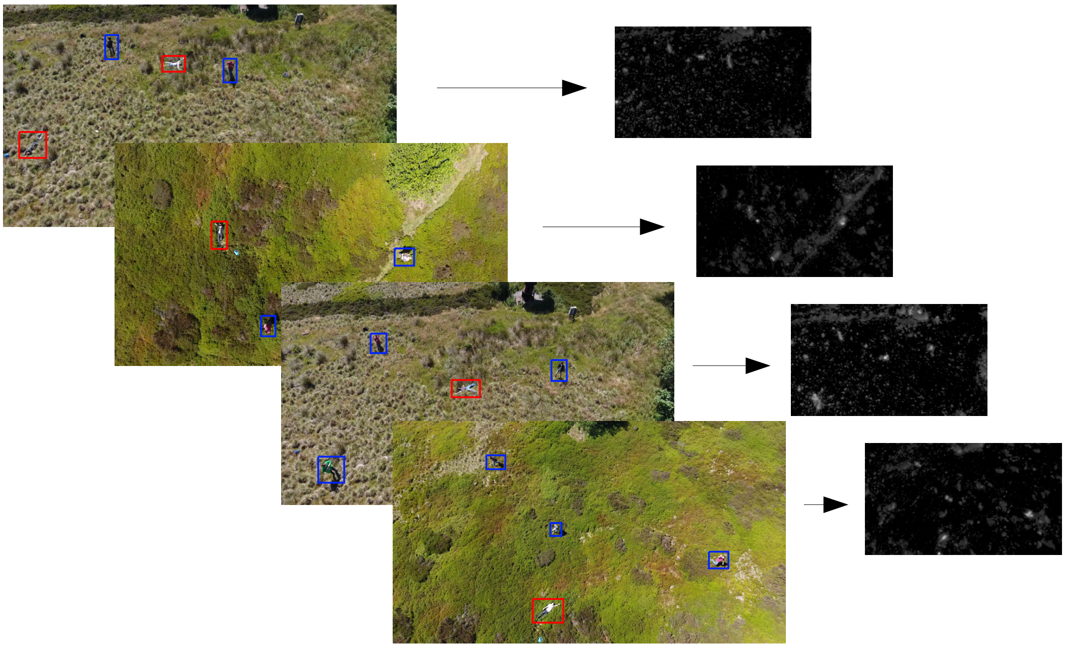


Figure 4.5: *Sample frames of our dataset, with the categories "person standing up" (in blue bounding boxes) and "person lying down" (in red bounding boxes), together with their saliency estimation.*

Frame-rate was tested in a portable GPU, from the Nvidia Jetson series, particularly the Nano Developer Kit. It has a 128-core Maxwell GPU, a quad-core ARM CPU and can be powered by a 5V-2A Micro-USB connector, with a weight of 138 grams (including the heat-sink). At this dimension, it is a good fit for lightweight drone visual processing, as exemplified in [126–128]. The number of Floating Point

Operations (FLOPs) is shown in Table 4.1 for YoloV3 and tiny-YoloV3 [105] (a simplified version of YoloV3, with 13 convolutional layers instead of the 75 in the full model), following Eq. 3.8 for every CNN layer, illustrating a drastic decline in the number of operations required as input resolution is made smaller.

Resolution	**YoloV3** (giga-FLOPs)	**tiny-YoloV3** (giga-FLOPs)
416 × 416	65.317	5.45
384 × 384	55.658	4.642
352 × 352	46.774	3.897
320 × 320	38.664	3.224
288 × 288	31.292	2.609
256 × 256	24.735	2.064
224 × 224	18.912	1.582
192 × 192	13.931	1.161
160 × 160	9.659	0.807
128 × 128	6.19	0.516
96 × 96	3.467	0.29

Table 4.1: *Floating point operations (in billions) for YoloV3 and tiny-YoloV3, trained on the 5 categories used for this project drone dataset. FLOPs for YoloV3 are very similar to those in Table 3.2, with 20 class labels, since only a few of the filter layers in the CNN need to be changed for a different number of classes. On the other hand, tiny-YoloV3 has significantly less convolutional layers (13 instead of 75), reflected in an overall smaller number of FLOPs.*

4.4 Results

Two main approaches are taken to evaluate the performance of this implementation. First, conventional saliency metrics provide insight into the influence of the top-down feedback loop. Although these metrics are designed and normally compared to human eye fixations, given as ground truth, in this context the main interest is on the objects that the CNN is trained to classify.

Second, the mean Average Precision (mAP, which is unit-less), a measure commonly used in machine learning, helps to validate that for objects that are within the vicinity of the fovea, the performance rate can be kept at a similar level than with bigger network/image sizes. While doing so, a considerable increase in frame-rate is obtained, for which it is tested using a portable GPU, with relatively low energy consumption requirements.

4.4.1 Visual saliency

The Area under ROC Curve (AUC), Pearson's Correlation Coefficient (CC), Kullback - Leibler divergence (KLdiv) and the Normalized Scanpath Saliency (NSS) are four commonly used saliency metrics. For the latter,

$$\text{NSS} = \frac{1}{N}\sum_i S_i \times G_i \tag{4.13}$$

where S_i is the overall saliency, N is the total number of pixels with a fixation and G_i is the binary ground truth map.

To study the influence of top-down versus bottom-up visual saliency, Fig. 4.6 shows the behaviour of overall saliency when varying β in Eq. 4.1. Given that both the ground truth and the top down predictions are rectangular boxes, AUC and NSS are most relevant here, as location-based metrics [129].

$1-\beta$	**AUC**	**CC**	**KLdiv**	**NSS**
0 *	0.551	0.161	3.680	1.420
0	0.546	0.084	3.49	0.820
0.1	0.591	0.158	3.620	1.374
0.3	0.730	0.289	3.306	2.301
0.5	0.796	0.392	3.035	2.982
0.8	0.802	0.462	3.00	3.432
1.0	0.813	0.481	8.507	3.573

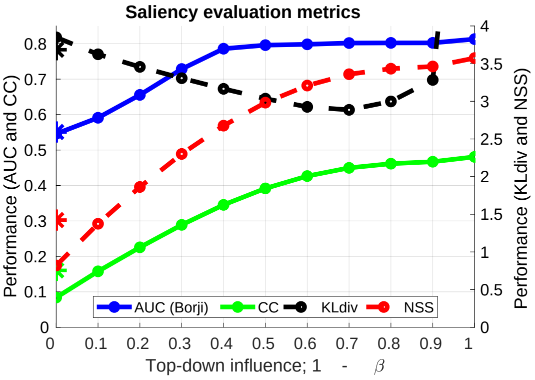


Figure 4.6: ***(Left)*** *Table with key values for performance changes using some of the conventional saliency metrics; AUC, CC, KLdiv and NSS, as presented in [129], while varying the weight β of the top-down influence.* ***(Right)*** *The graph shows the behaviour for values of $\beta \in \{0, 1\}$. Given the nature of the ground truth (binary bounding boxes with the top-down object locations), it its expected that a larger effect of the top-down information, $1 - \beta$, will give a better result. However, for most metrics, the performance flattens around $\beta = 0.3$ to $\beta = 0.6$, supporting that hypothesis that a good balance is obtained giving equal weight to the bottom-up and top-down information.*

A performance increase for these saliency metrics can be expected when the top-down information has a larger weight than the bottom-up. However the AUC reaches an almost steady level around $\beta = 0.5$. Of these four metrics, the KL-divergence is the only one for dissimilarity, instead of similarity, meaning that a lower value signifies a better prediction of saliency [129]. In this case, a key point to remark is that the best performance is obtained with approximately a similar influence for top-down and bottom-up information.

The value of how gamma (in Eq. 4.1) affects the overall saliency is illustrated in Fig. 4.7, using the NSS metric (Eq. 4.13), by making $\gamma = [a, b, c, d, e]$, where any of $\gamma_{\{a,\ldots,d\}} = 1$ when the corresponding label and $\gamma_n = 0.5$ for all the rest, for example $\gamma = [1, 0.5, 0.5, 0.5, 0.5]$ to give priority to the first label (person lying down). Even when this effect is not drastic, it can help to prioritize information.

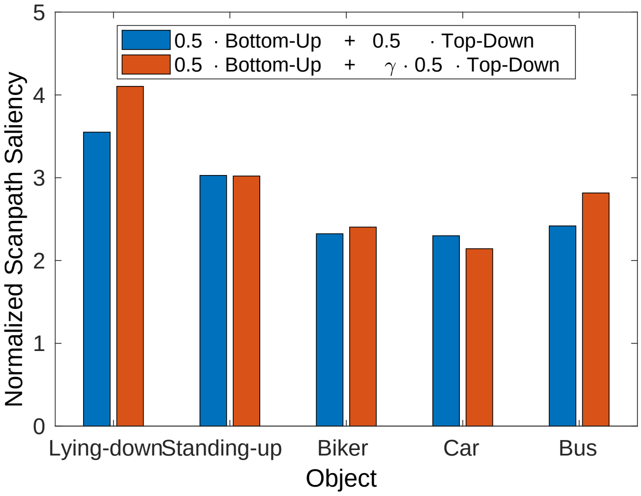


Figure 4.7: *The γ influencing factor allows to give priority to any of the top-down detection categories, to make it more likely for the fovea to stay centred on it, when it is required by the task. In this case, $\gamma_i = 1$ for every of the plotted objects, and $\gamma_i = 0.5$ for the rest of them. The effect of γ is determined by the difficulty and frequency in which each object appears, but in most cases at least a slight increase is obtained, compared to treating all categories equally (blue bars), here using the NSS metric.*

4.4.2 Object detection and recognition

The most common method for detection-recognition evaluation is to obtain the Intersection over Union (IoU) between the ground truth bounding box and the prediction box:

$$\text{IoU} = \frac{\text{Area of overlap}}{\text{Area of union}} \tag{4.14}$$

Used with an arbitrary threshold to determine if a predicted box can be considerate positive. The mean Average Precision (mAP) is then calculated using the metrics of the PASCAL VOC 2012 competition [130], with an IoU of at least 50%. The performance of the instances where the objects are at least 30% into the foveal region is measured.

As the network sizes are made smaller, fewer objects are considered. But those that are can be taken with a higher confidence as true positives, while with a conventional linear downsample performance is affected near linearly by network size. Additionally, with smaller networks, the slowdown of adding the bottom-up saliency and foveal transformation becomes more noticeable.

Fig. 4.8 gives some of the key mAP (unit-less) values while using the tiny-YoloV3 network. And Fig. 4.9 does the equivalent while using the complete YoloV3 model, together with their respective frame-rates. While the values vary considerably depending on the network, or when only using the more difficult subsection of the dataset, the behaviour is consistent, where the performance is considerably more steady for the detections that appear in the fovea, as easily seen in the right side graph of Fig. 4.9. A key result from Fig. 4.8 is that for the foveal images (last row), the performance can even be seen to increase for the objects in the fovea.

Network size	416 × 416	256 × 256	192 × 192	160 × 160	128 × 128
Frame-rate	8.89	20.31	26.92	31.48	37.54
Normal	45.12	26.12	8.07	1.3	0.08
Foveal	38.88	49.15	44.18	46.76	46.48
Normal (S)	21.71	12.37	5.17	1.57	0.43
Foveal (S)	17.55	17.52	17.72	28.43	30.79

Figure 4.8: *Key values for mean Average Precision performance (unit-less) using the tiny-YoloV3 neural network. The second row exemplifies the frame-rate averaged by all the test images on the Jetson Nano, going from 8.89, at a resolution of* 416 × 416*, to 37.54 frames/second at* 128 × 128 *pixels. In the foveal images, performance can be seen to maintain a steady level, although considering less objects as the scale decreases, only those that are at least 30% present in the rows and columns selected for the foveal transformation. The last two rows, marked by a (S), give the performance when only considering the Stanford dataset images, which proved to be considerably more difficult, but where the effect of the foveation remained.*

When using the Faster-RCNN network (Fig. 4.10), only trained at full 416 × 416 resolution, it is clear that it does not generalize as well for smaller resolutions (marked by the steep performance decline in the normal downsample). The foveated images show a behaviour similar to the one described in the previous cases. This implementation also did not show a considerable speed-up, staying around 1.1 frames/second on the Jetson Nano, due to the use of a Region Proposal Network and an overall much larger number of FLOPs.

Network size	416 × 416	192 × 192	160 × 160	128 × 128
Frames / second	1.44	4.57	5.11	6.69
Normal	72.56	21.3	20.65	10.31
Foveal	69.16	58.49	40.61	30.89
Normal (S)	35.51	15.78	12.25	5.53
Foveal (S)	30.79	28.79	25.84	21.41

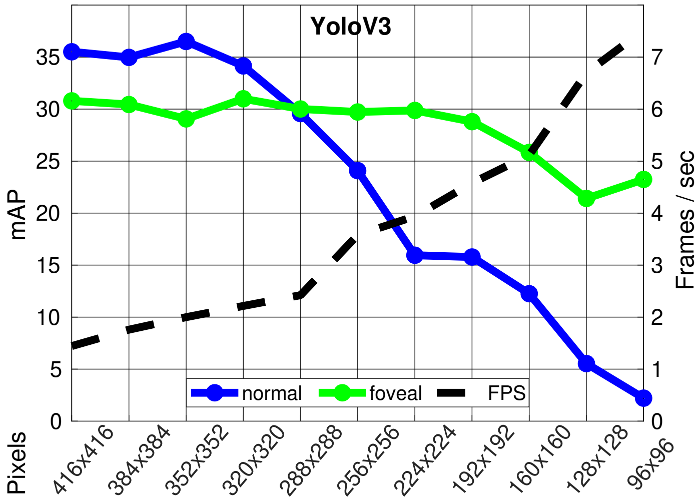


Figure 4.9: Mean Average Precision (unit-less) using the YoloV3 neural network. The second row shows averaged frame-rate for all the test images on the Jetson Nano development board (running at high priority). The last two rows give the metrics when only evaluating the Stanford drone dataset (same as the graph on the right), which proved to be considerably more difficult. But both of them show a similar performance trend.

4.5 Discussion

This chapter has presented a novel visual saliency algorithm with foveated vision that enables fast object detection and recognition using low power GPUs. It was shown how downsampling an image, while keeping a small high-resolution region, allows to maintain confidence in the CNN predictions comparable to those at higher resolutions, with the trade-off of performance on the low-resolution areas (the periphery), while still using the saliency information from said periphery. The visual saliency system was demonstrated on two datasets: the Stanford drone dataset and the collected UAV test set. The results showed the benefit of a visual saliency algorithm in the applications domain of UAVs, where objects of interest (persons, vehicles, animals, etc.) are often small and naturally different from the rest of the scene (and hence more salient).

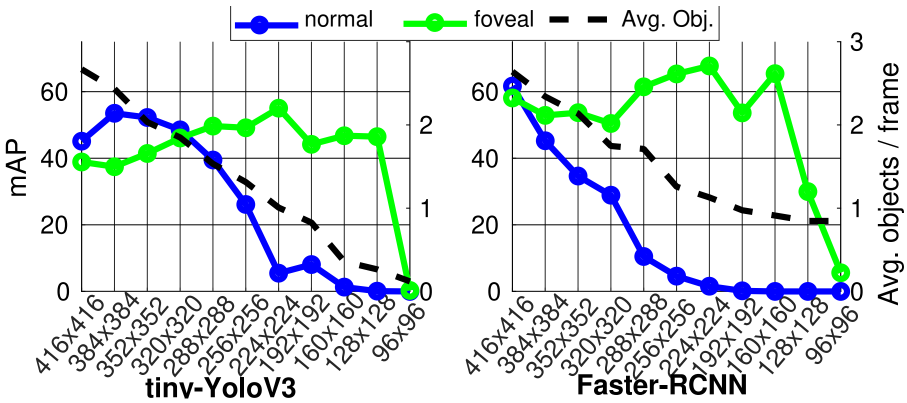


Figure 4.10: *Mean Average Precision (unit-less) evaluation for only the images taken from our drone, the performance is considerably better (as shown in these two graphs), supporting the view that this section of the test images is easier to learn for the CNN, with around 3 object appearances per frame. The right axis shows the average number of objects taken into account for evaluation in each case, selected by being at least 30% in the fovea.*

Since any robot would be presented with a video feed and not a single static image, the next challenge is how to select the best subsequent region of interest to direct the fovea, for which it is shown how conventional bottom-up saliency methods, with feedback from previous predictions, can point in the right direction. It is then required to account for a more versatile decision making for the fovea location selection, based on current understanding of how this mechanism takes place in the human brain. This scheme serves as a proof of concept on how to bring together several state-of-the-art computer vision elements that have been developed based on current understanding of the human visual system. Some of these principles, such as saliency, have been established for a few decades. Meanwhile others, e.g. Convolutional Neural Networks, have seen a very rapid grow in the last few years.

Advantages of saliency and foveation are evident in animals that permanently select what part of their field of view is most pertinent to focus on. For a robot facing an akin dilemma, also limited by its capacity to compute the full scene in high acuity, the implementation requires constraints and trade-offs that are less straight forward to implement. This becomes even more relevant in UAVs, that need to autonomously lift the weight and supply them with energy for the entire operation time.

Chapter 5

Visual saliency with multiple foveas

5.1 Introduction

One important debate about foveal implementation models and bio-inspired engineering to a larger extent, is how to determine which compromises to make in terms of biological fidelity. Different representations of foveation have been proposed over the years. Log-polar being one of the most popular ones [29], by transforming from coordinate system variables (x, y) to (ρ, θ);

$$\begin{aligned} \rho &= \log \sqrt{(x - x_o)^2 + (y - y_o)^2} \\ \theta &= \text{atan2}(y - y_o, x - x_o) \end{aligned} \tag{5.1}$$

ρ is the logarithm of the distance from (x, y) to the center of the image (x_o, y_o), and θ is the angle between the origin and the line that crosses them. Generally, those representations that preserve Cartesian geometry, are more suitable for the adaptation of methods designed for normally sampled images. In the present application, transfer learning for object detection and localization methods play an important role, given the amount of data available. As described by Traver and Bernardino [29], other approaches generally aim at preserving linearity, flexibility of foveal position and size. Also, although less typically, multiple foveation approaches have been studied for early applications like bandwidth compression [131–133]. In this chapter, it is explored how this approach can be incorporated into this implementation and its effect in performance, as well as exploring if explicitly providing movement information channel to the object detection stage gives a clear advantage in keeping the foveation focused on the objects in which it was trained.

Thus far, the foveation strategy has being set on following a logarithmic operation, with a behaviour similar to previous implementations and easily accessible in any computer system. Most foveation methods are also similar in that they attempt to show the foveation phenomenon in one eye. But for the information flow presented, where foveation is followed by a conventional CNN object detection test, it is compelling to hypothesize that there is a benefit to having multiple foveas in a single image, allowing to simultaneously pay attention to more than one section of the image.

Dhavale and Itti [132] proposed an early model of foveation based on bottom-up salient points and on multi-foveation, with the goal of video compression. In their implementation, each foveation amounts to a separate foveated image. For n of such foveations, the value of the pixel (x, y) of the final foveated image is;

$$S(x,y) = \max_{i \in [1,\ldots,n]} (w_i V_i(x,y)) \tag{5.2}$$

where $V_i(x, y)$ is the pixel value in the i^{th} foveated image with a Gaussian pyramid, and w_i is the saliency value. Oliveira et al. [134] propose the use of multi-foveation for the reduction of information in 3D point clouds, with special emphasis on the elimination of redundant points caused by the multiple foveas. Lim et al. [135] studies multiple object foveation and tracking in the log-polar space. Hunsberger et al. [136] applies multi-foveation using disparities from stereo visual input as points of interest, chosen as a cost function $C_{x,y}$:

$$C_{x,y} = \sum_{x,y} w_{x,y} \min \left(|d_{x,y} - d^*_{x,y}|, C_{sat} \right) \tag{5.3}$$

in pixel (x, y), with an estimated and ground truth disparities $d_{x,y}$ and $d^*_{x,y}$ respectively, as well as a pixel error saturation weight C_{sat}.

5.2 Methods

In this Chapter and the subsequent one, two additional deep neural networks for objected detection are used; Single Shot Multibox Detector (SSD) [137] and Mobilenet [138], given their relevance in the literature and their focus in computational lightness. These approaches are first briefly introduced in Subsections 5.2.1 and 5.2.2.

5.2.1 Single Shot Multibox Detector

SSD [137] is a CNN object detection model that takes principles both from YoloV3 and Faster R-CNN. It has two stages; first with feature extraction maps, followed by convolution filters for object detection. The former creates a set of bounding boxes of fixed size and scores for object classes in those boxes (equivalent to the anchor boxes in Faster R-CNN), and the last stage uses non-maximum suppression (nms) to eliminate duplicate predictions of the same object, ranking by confidence and Intersection over Union (IoU) score.

SSD uses as an overall loss function, the weighted sum of localization, L_{loc}, and confidence, L_{conf}, loss;

$$L(x,c,l,g) = \frac{1}{N}\left(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)\right) \tag{5.4}$$

N are the matched bounding boxes. L_{conf} is the softmax loss over class confidences, c. Given the i^{th} box of the j^{th} ground truth box in the P category, and $x_{ij}^{p} = \{1,0\}$ being and indicator of them matching:

$$\begin{aligned} L_{conf}(x,c) &= -\sum_{i\in Pos}^{N} x_{ij}^{p}\log(\hat{c}_i^{p}) - \sum_{i\in Neg}\log(\hat{c}_i^{0}) \\ \hat{c}_i^{p} &= \frac{\exp(c_i^{p})}{\sum_p \exp\left(c_i^{p}\right)} \end{aligned} \tag{5.5}$$

The L_{loc} between the ground truth g and predicted box l, is calculated using a similar formulation to the one presented by Girshick [139]:

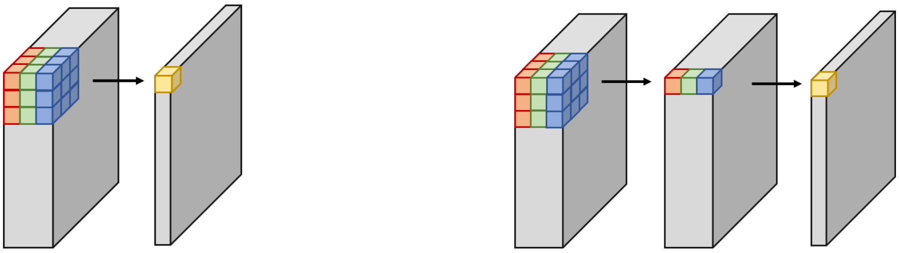
$$\begin{aligned} L_{loc}(x,l,g) &= \sum_{i\in Pos}^{N} \sum_{m\in\{cx,cy,w,h\}} x_{ij}^{k}\ \mathrm{smooth}_{L_1}(l_i^{m} - \hat{g}_j^{m}) \\ \mathrm{smooth}_{L_1}(x) &= \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \end{aligned} \tag{5.6}$$

Since it does not use a region proposal step, making it a single pass method, similar to YoloV3, it is interesting to see how performance is affected in the detection of distorted objects both in the fovea and in the periphery.

5.2.2 Mobilenet

Another appropriate CNN implementation approach on which to test the effect of foveation is Mobilenet [138], since it is explicitly designed for fast performance. It uses several implementation speed-up techniques, namely; the introduction of depth-wise separable convolutions, as first presented by Sifre and Mallat [140] and Hinton et al. [141], commonly referring to the method as *Distillation*, consisting of using the training of a large CNN to aid the training of a smaller one.

In contrast to a conventional convolution layer, where a kernel slides across an image (using a kernel with the same number of depth channels than the input) and performing a weighted sum of them. A depth-wise separable convolution, illustrated in Fig. 5.1 *(b)*, first performs the convolution of each channel independently, creating an output with the same number of channels to the input, followed by a sum of the layers (or a regular convolution with a 1x1 kernel), thus requiring a smaller amount of weights for the network to learn.



(a) *Conventional convolution layer* ***(b)*** *Depth-wise separable convolution*

Figure 5.1: *Illustration of the difference between conventional convolution implementation and depth-wise separable convolutions. By splitting the kernel into two smaller ones, the number of multiplications decreases. E.g. a 3×3 kernel takes 9 multiplications, while it can be represented by two smaller kernels with 3 multiplications each;* $\begin{bmatrix} 4 & 5 & 6 \\ 8 & 10 & 12 \\ 12 & 15 & 18 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \times \begin{bmatrix} 4 & 5 & 6 \end{bmatrix}$

The contrast between conventional convolutional layers and depth-wise separable convolution is also of special interest in this case to measure the effect of adding a 4th channel for movement on how the information is filtered.

5.2.3 Multiple foveas

In Chapter 4 it was described how the foveation of a conventional image into a smaller one can be done and saved into a Look-up-table (LUT) for quick consultation, thus avoiding the need to do online computations to run the system. To obtain an image foveated into more than one region, they are calculated in a loop until reaching the required number of rows and columns of pixels. But once this estimation is done, it mainly translates into a larger LUT. At a first glance, there is no computational limitation to the number of foveas that could be implemented in an image. From the saliency estimation point of view it is easy to catalog and sort any number of conspicuity areas, once an overall map is produced. Here it is explored how the CNN performance is affected by placing several artificial foveas in a single image.

A main hypothesis explored in this chapter is that it might be best suited to lay out two foveas, constrained to the same vertical position (dictated by the first most salient point), but with independent horizontal locations (based on the first and second salient points). This serves a couple of goals; since a lot of modern image formats have a wider ratio, close to 2:1, but most convolutional networks are square. So with two horizontal regions to foveate on in the wide axis, the amount of distortion in the periphery decreases, and more rapidly reaches the desired square matrix shape. It is also closer configuration to the one of the human field of view, due to the location of our eyes (which are vertically aligned). It is also tested with two "foveas" without the horizontal axis constraint, as well as three and four, to analyze their effect.

Adapting the previously presented foveation algorithm (Fig. 3.1), to loop over the number of desired foveas, increases the number of selected pixels to create the re-sampled image. A larger input image also signifies a larger LUT; e.g. with the original frames at 848 × 480 pixels, calculating the bottom-up saliency at a downsampled size of 120 × 212 pixels (by a factor of 4), requires a LUT of 212^{Fov} rows of indexes (in the wider size), where Fov is the number of foveas. The total of columns in the LUT is the desired final size of the image (separate LUTs are created from 96 × 96 to 416 × 416, at intervals of 32 pixels, to match the CNN).

To get the indexes required to foveate in x_{Fov}, for one fovea it simply is needed to check the row with that number in the width LUT; $LUT[row] = x_{Fov}$, and the same applies for y_{Fov} in the height LUT. On the other hand, in the case of two foveas it becomes $LUT[row] = x_{Fov1} \cdot \text{width} + x_{Fov2}$, for three; $LUT[row] = x_{Fov1} \cdot \text{width}^2 + x_{Fov2} \cdot \text{width} + x_{Fov3}$, and so on for any number of foveations.

The fastest way to run the foveation implementation using this approach is first loading the full LUT into Random-Access Memory (RAM), but given its exponential growth in size, it becomes a limitation beyond 3 foveas. For example, the LUT for all the possible locations to foveate from 480 (height) pixels into 128 is of 58.1 KB, and from 848 (width) pixels into 128 is of 105.2 KB. For two foveas, the corresponding LUTs take up 7.5 MB and 23.2 MB, respectively, and for 3 foveas; 116.2 MB and 630.2 MB. While such sizes are still manageable and can be fully loaded into the memory of a Nvidia Jetson Nano (with 4 GB of RAM), escalating to four foveas does become too big of a computational burden. Nevertheless, for reference this last one is run on a personal computer with 20 GB of memory and a Nvidia GeForce GTX 950M graphics card, exclusively using the tiny-YoloV3 object detection model. One alternative is not to load the complete LUT into memory at the start of the algorithm, but only the required elements on the run, or using swap memory, at the cost of a higher computation time.

A foveation from a 848 × 480 frame into 288 × 288 pixels is exemplified in Fig. 5.2, using from one to four "foveas" (subfigures from left-right and up-down respectively). A similar example, now foveating to 160 × 160 pixels is shown in Fig. 5.3.

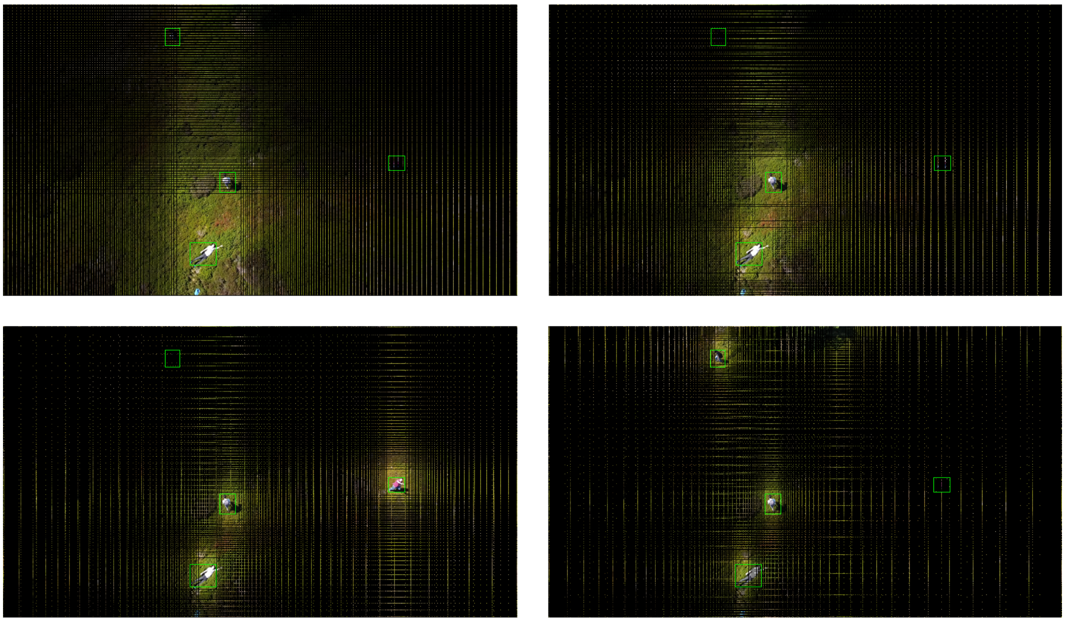


Figure 5.2: *Example of using multiple foveas in a single image; an aerial frame with resolution of* 848 × 480 *pixels is foveated into* 288 × 288*, based on their accumulated saliency. There are 4 objects of interest present in the scene. From top to bottom, and left to right;* ***(a)*** *With one fovea, one object of interest appears clearly in it, while a second one is in the edge of the fovea.* ***(b)*** *Using two foveations, an instance of objects of interest is shown clearly in each fovea, albeit their size is considerably smaller.* ***(c)*** *Three foveas exemplified with an object of interest captured in each one.* ***(d)*** *Four foveas, each considerably smaller than in the previous cases. In this example, the fourth foveas fails to locate the last of the 4 objects. The likely success of each of this configurations heavily depends on the size and saliency of the objects of interest. If it is expected that in a configuration with 4 foveas, the object of interest will fall mostly inside the fovea, having more foveas increases the possibility of finding all the objects. In contrast, if it is not believed that the objects of interest will have high saliency, a small foveal will most likely not be able to find them.*

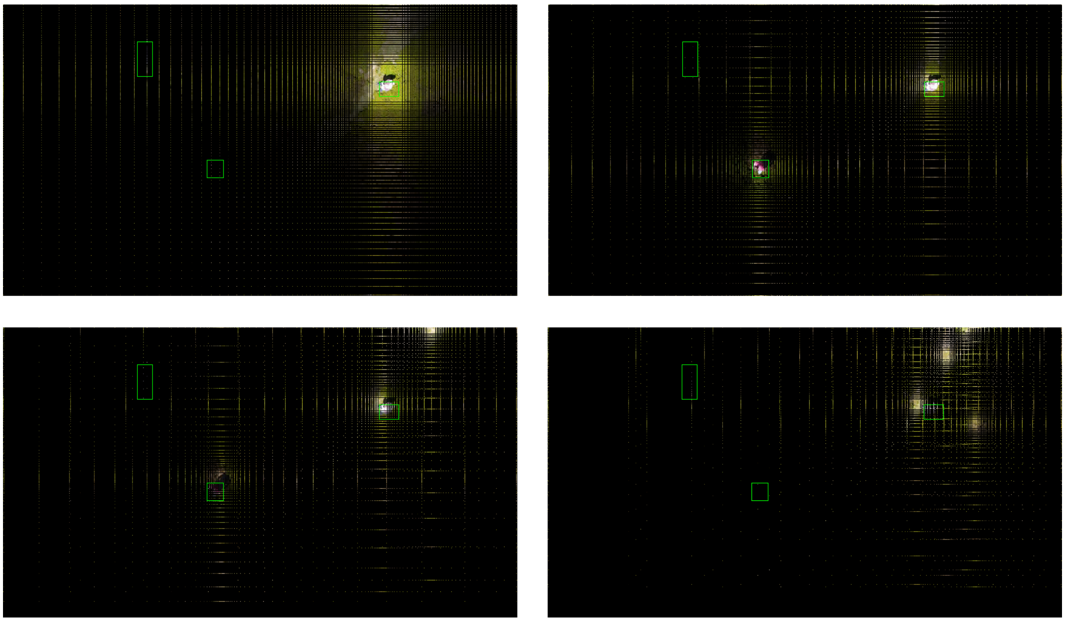


Figure 5.3: Sample of multi-foveation, from one to four foveas (left to right and top to bottom), down-scaling from 848 × 480 *to* 160 × 160 *pixels. This example is similar to Fig. 5.2, but with a much steeper downsizing and thus less successful object detection. In this frame there are 3 objects of interest, in most cases here not placed in the fovea. Given that the saliency module determines the fovea location, and that the top-down feedback from previous frames feeds into it, an unsuccessful object detection might cascade into deteriorating performance in subsequent ones, illustrating how using a single fovea might be more productive than multiple.*

5.2.4 Movement channel

A second strategy explored in this chapter is to give more prominence to the movement information, which is not included in a lot of saliency models, since most conventional approaches do not have a time element built into them [142]. From a simple visual inspection of the UAV data used in the previous chapter, in the case where the UAV is not making very drastic movements or travelling, movement can be expected to be a very good cue of the objects of interest for this type of data, since the background is mostly static and the objects (or persons) are travelling. If the UAV is travelling, it becomes more difficult to distinguish movement from objects in the flow from change in the point of view from tue UAV.

Some image formats support to have a fourth channel in addition to the conventional Red-Green-Blue (RGB), most commonly used for transparency but carrying the same weight to the rest. In this section, the effect of embedding the movement information in this channel is explored for the Convolutional Neural

Network to learn. Thus it can be weighted in addition to being part of the saliency estimation. The Tensorflow libraries where used in this Chapter [143], as well as the Tensorflow Object Detection API [144] implementations of SSD [137], and Mobilenet [138].

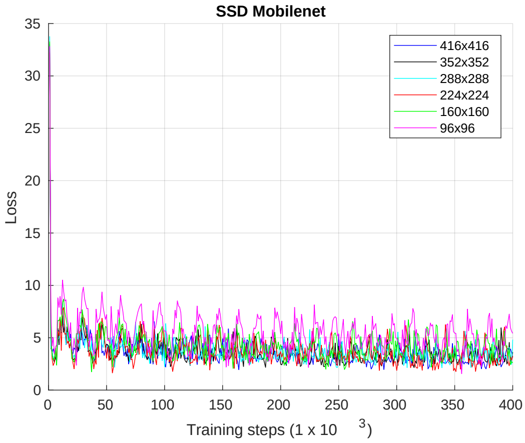
SSD [137] is interesting in the current context since it does not use feature resampling stages for region proposal and pooling. It is a single shot detector, similar to YoloV3. The implementation by Huang et al. [144] also provides the advantage of being made to compare systems against each other, thus having SSD [137] and SSD Mobilenet [138] in similar terms.

5.2.5 Training

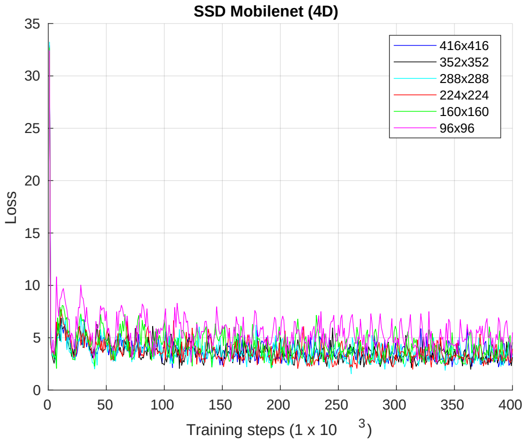
In this instance, the network pre-trained in the COCO dataset [104] is also taken as starting point, as provided by their respective authors [137, 138] and then re-train in UAV data for this project. This is done using the Tensorflow Object Detection API [144] for 400,000 steps, with a batch size of 24 and learning rate of 0.004, using the rest of the parameters as defined in [138]; momentum of 0.9 and no learning rate decay. For the 24,020 training images this means $\sim$ 400 epochs. As illustrated in Fig. 5.4 for the training of Mobilenet in both RGB and RGB-Movement (RGB-M) format, it reaches a relatively fixed state considerably sooner. For consistency, all models are trained by the exact same number of steps. Due to time constraints and access to equipment, Mobilenet in RGB format was trained at every other of the 11 size variations that have been used in previous sections, ending up with; 416×416, 352×352, 288×288, 224×224, 160×160 and 96×96, and interpolating the missing data points. Similar to previous chapters, the input images are set to the same size as the network and re-training for every resolution.

5.3 Results

Figure 5.7 shows the mean Average Precision (mAP) performance for the four different neural network models tested. Using a similar approach from Chapter 4, the considered bounding boxes are those that are at least 30% in the pixel rows and columns taken into the fovea to measure mAP. The remaining predictions (what can be considered as peripheral detections) serve as feedback into the saliency estimation for the frame in the next time step. The main observation from Fig. 5.7 is how the object detection performance remains being better when using a single fovea, consistently across CNN models, and decreasing as the number of foveas increases. Perhaps more counter intuitively, the average number of objects that meet the 30% criteria also decreases, as seen in Fig. 5.8. The exception to this

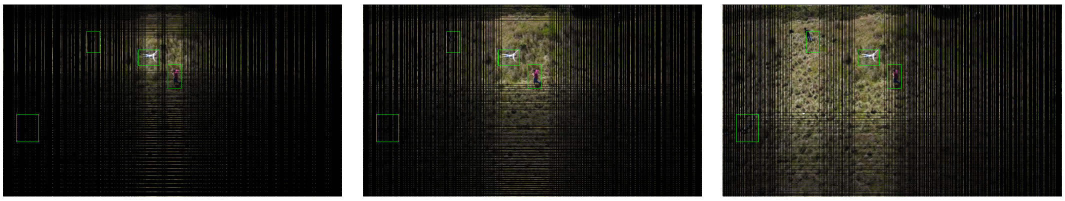


(a) *RGB training*



(b) *RGB-Movement training*

Figure 5.4: *Re-training Mobilenet and Mobilenet (RGB-Movement) at different resolutions, during* 4×10^5 *training steps, equivalent to* ~ 400 *epochs. All the implementations where trained by this same number to maintain consistency. Smaller resolutions have a loss value that fluctuates considerably more, which is consistent with out expectations.*



(a) 192 × 192

(b) 288 × 288

(c) 384 × 384

Figure 5.5: *Sample frames with a* 2×1 *foveation configuration, downsizing an image from* 848×480 *pixels to* 192×192, 288×288 *and* 384×384 *pixels respectively (from left to right). In (b), the horizontal location of both foveas is near the center, causing an effect similar to using a single fovea, this causes a smaller distortion than having two or more foveas without the vertical matching constraint, as shown in Figs 5.2 and 5.3.*

trend is the case of two foveas restricted to the same vertical position (labeled as 2×1). It shows a very similar performance to the case of a single fovea, with a slightly larger number of considered objects, this is also illustrated in summary Table 5.1 for YoloV3 and SSD Inception. Fig. 5.5 exemplifies three foveated frame into this configuration, resulting in images of 192×192, 288×288 and 384×384 pixels, respectively. It can be observed that since the vertical position is fixated in the prominent salient points, an object might or not appear in the second fovea, but its shape is much less distorted than in the other multiple-fovea cases. There is also the case where the two prominent salient point in the horizontal axis are close together, giving a similar behavior of a single fovea (Fig. 5.5b).

In the case of training with 4-channel images (RGB-M), the general observa-

Resolution	**mAP, YoloV3**		**mAP, SSD Inception**	
	One fovea	2 × 1	One fovea	2 × 1
416 × 416	22.55	22.53	21.17	20.97
384 × 384	23.2	22.93	25.15	22.89
352 × 352	21.32	21	23.66	22.16
320 × 320	20.13	20.32	23.78	25.77
288 × 288	20.95	18.95	21.22	20.34
256 × 256	20.8	15.44	24.37	22.88
224 × 224	16.56	13.42	24.62	22.14
192 × 192	15	15.31	23.73	20.07
160 × 160	17.26	12.93	22.32	19.46
128 × 128	12.21	8.7	8.65	2.36
96 × 96	4.47	5.61	12.05	8.79

Table 5.1: *mAP performance (unit-less) comparison for one fovea and two foveas fixed with a same horizontal position, for YoloV3 and SSD Inception, showing a relatively similar behaviour in most cases.*

tions from Section 4.5 still apply, with a slight increase in performance and a smoother curve, plotted in Figure 5.6. Although it is debatable if substantially enough to justify the increased number of computations, the correlation in mAP between RGB and RGB-M is presented in Table 5.2 for the case of a single fovea. For example, at 416 × 416 resolution, the performance for SSD Inception is 21.17 mAP with RGB images, and 27.37 mAP with RGB-M images, while for smaller resolutions; at 160 × 160 the performances are 22.32 mAP and 25.82 mAP respectively. Followed by a large drop in the performance curve for smaller resolutions in both cases; to 8.65 and 13.11 mAP, at 128 × 128 pixels respectively. Considering that the advantage gained by the movement channel is highly dependant on how fixed the background remains. In the case of an UAV, if it is not travelling, the movement channel is much cleaner and useful. In the recorded test images there is a combination of the drone both hovering and travelling.

5.4 Discussion

In this chapter, the overall goal was to explore two additional strategies to enhance object detection performance; introducing more than one fovea into the image downsampling, and the use of a movement channel as part of the input for object detection. In broad terms, the observed performance was in-line with a conservative expectation of it.

Given the nature of the aerial images used, in the instances where there is camera movement (camera on-board a travelling UAV), the movement of the ob-

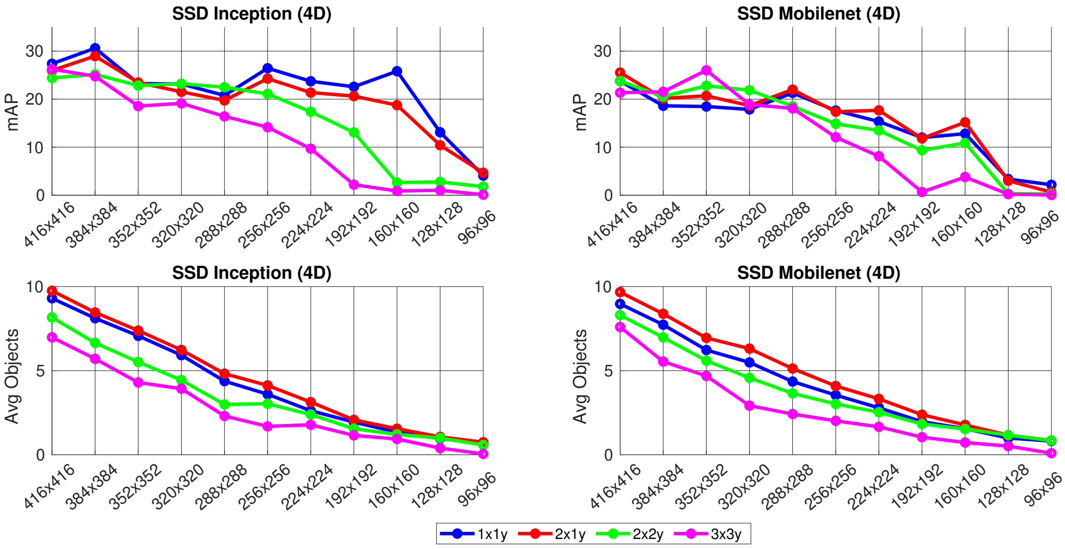


Figure 5.6: mAP performance for SSD Inception and SSD Mobilenet after retraining with RGB-Movement images. In this instances, the behaviour is very similar to the conventional RGB images (Figures 5.7 and 5.8), but with a small overall mAP boost at in most cases. The single fovea and 2×1 *configuration continue to provide the best overall performance.*

Resolution	mAP, SSD Inception		mAP, SSD Mobilenet	
	RGB	RGB-M	RGB	RGB-M
416×416	21.17	27.37	16.51	23.78
384×384	25.15	30.61	-	18.62
352×352	23.66	23.25	18.89	18.47
320×320	23.78	23.15	-	17.87
288×288	21.22	20.78	21.71	21.31
256×256	24.37	26.44	-	17.63
224×224	24.62	23.73	4.22	15.35
192×192	23.73	22.58	-	12.01
160×160	22.32	25.82	7.17	12.82
128×128	8.65	13.11	-	3.37
96×96	12.05	4.08	0.75	2.18

Table 5.2: mAP (unit-less) object detection performance using SSD Inception and SSD Mobilenet with RGB and RGB-M formats, illustrating an overall precision increase with the addition of an explicit movement information channel.

jects of interest has a different nature than the one from the camera. When the UAV is hovering, the saliency of the moving objects is much easier to detect. Since the saliency module already includes this estimation of movement, the question becomes whether stating this information much more explicitly (as a 4th channel supplementary to the conventional red-green-blue) provides a benefit large

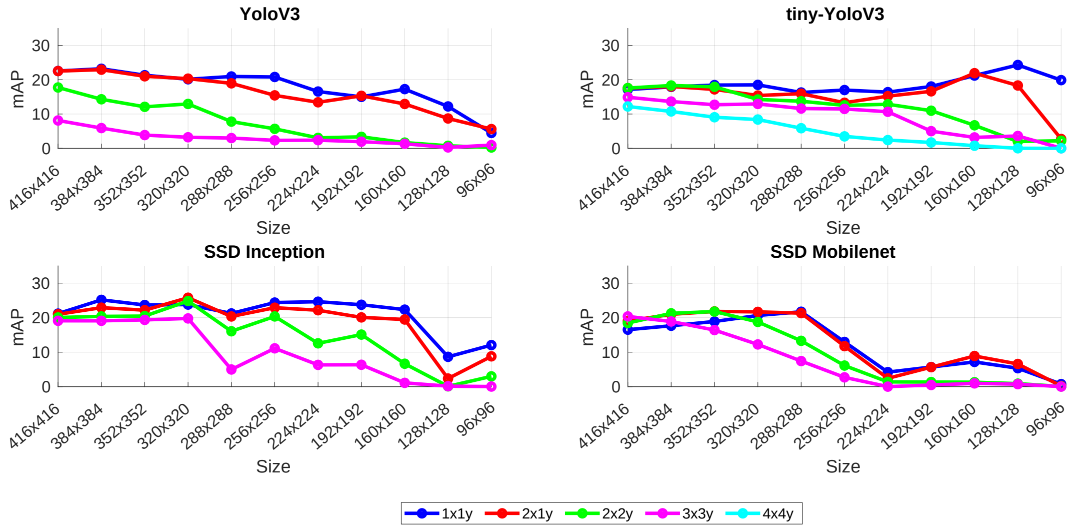


Figure 5.7: Mean Average Precision (mAP, unit-less) performance for YoloV3, tiny-YoloV3, SSD Inception and SSD Mobilenet on the aerial images test data (as described in Section 4.3). Remarkably, for all four implementations there are very similar results between using a single fovea and the 2×1 *configuration previously described (where two foveas are constraint to the same vertical position). In the other cases, precision seems to decrease as more foveas are introduced, given that they create a bigger distortion in the periphery. Overall SSD Inception gives the most stable performance, while SSD Mobilenet rapidly decreases in precision at smaller scales. Excluding this last one, a mAP at* 160×160 *pixels is still comparable to the larger* 416×416 *resolution.*

enough to justify the increase in computations in the object detection module. It was observed that while there is a mAP performance increase (more evidently in the CNNs at resolution smaller than 256×256), it might not be best suited for computation in portable GPUs, since the overall goal has been to minimize the number of convolutional operations required.

The use of multiple artificial foveas appears to be inherently a good fit for aerial perspective test images, given that there are a few objects of interest, and most of the background can be placed in the periphery. Even if the background is not a solid color and is mostly irregular (like the meadow shown in the recorded UAV images), the bottom-up saliency approach has been effective in categorizing it. As reviewed from related literature, the introduction of more than one fovea also presents some drawbacks. Here it also translates into a much larger Look-Up-Tables and a distortion in the pixels that do not fall into any of the foveas. Another promising strategy is to explore the time domain, in the form of decision making for when to trigger an object detection stage. As had been described in the thesis outline, this is undertaken in the next chapter based on a model of how decision making has been observed to take place in the basal ganglia region of the brain.

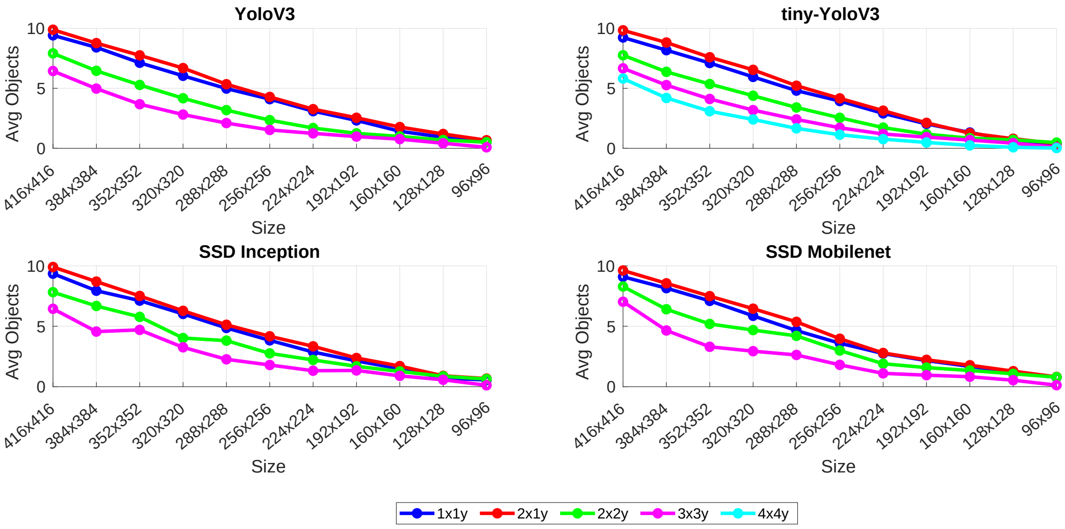


Figure 5.8: *Average number of objects that fall into the fovea in the test images, at least by 30% as described earlier, for multiple foveas and object detection implementations. The threshold is used to obtain the mAP performance in Fig. 5.7. It is evident that all the CNN models used follow a very similar behaviour, where the single fovea and the* 2×1 *configuration allow a larger number of objects, perhaps counter-intuitively, than using more foveas but with a smaller central region each.*

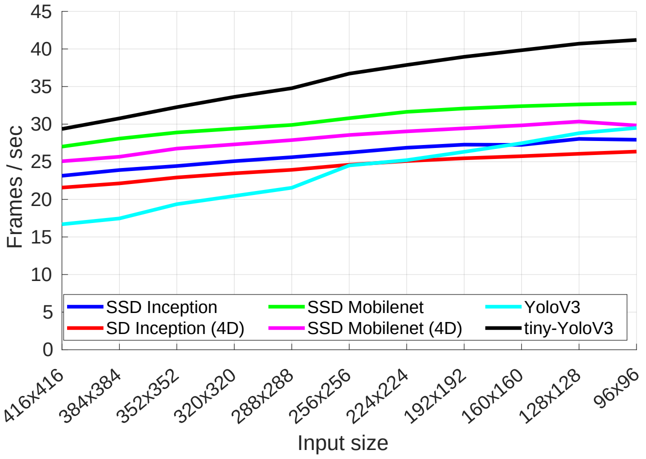


Figure 5.9: *Frame-rate performance for YoloV3, tiny-YoloV3, SSD Inception and SSD Mobilenet, the latter two also with the additional input channel (RGB-M). This test was performed on a Nvidia GeForce GTX1070, with 8GB of memory and 1920 CUDA cores. The number of Floating Point Operations (FLOPs) are also shown in Fig. 5.10. Tiny-YoloV3 and YoloV3 are respectively faster and slower than the implementations of SSD, but also seem more sensible to down-scaling, a more relevant advantage in our focus. While the overall number of FLOPs does not seem to drastically change with the introduction of the movement channel (shown in Table 5.3 and Figure 5.10), the effect on frame-rate is more noticeable.*

Resolution	**SSD Inception** (giga-FLOPs)		**SSD Mobilenet** (giga-FLOPs)	
	RGB	**RGB-M**	**RGB**	**RGB-M**
416 × 416	14.04	14.11	4.19	4.21
384 × 384	11.94	12.01	3.55	3.57
352 × 352	10.05	10.11	3.0	3.02
320 × 320	8.29	8.34	2.47	2.48
288 × 288	6.74	6.77	2.01	2.02
256 × 256	5.31	5.34	1.58	1.59
224 × 224	4.07	4.1	1.22	1.23
192 × 192	2.99	3.0	0.888	0.894
160 × 160	2.08	2.1	0.625	0.629
128 × 128	1.33	1.33	0.394	0.397
96 × 96	0.752	0.757	0.227	0.228

Table 5.3: *Floating point operations (in billions) for SSD Inception and SSD Mobilenet. Both implementations are significantly lighter than YoloV3, and Mobilenet is of a very similar scale than tiny-YoloV3 (Table 4.1). Remarkably the use of a 4th input channel, with RGB-Movement format, does not have a very large effect in terms of FLOPs, assisted by the implementation of depth-wise separable convolutions. Although a it does cause a drop in frame-rate, as seen in Fig. 5.10, which might also be affected by other factors such as image loading time.*

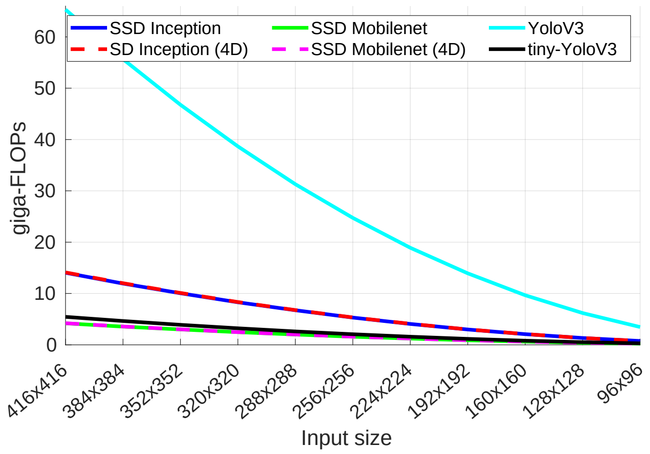


Figure 5.10: *Floating point operations (in billions) for input size resolutions from* 416 × 416 *to* 96 × 96 *for SSD Inception, SSD Mobilenet, tiny-YoloV3 and YoloV3. It illustrates how the latter takes a much larger number of operations (with considerably more convolutional layers), specially at larger resolutions. This can be observed as a steeper change in the number of FLOPs as the resolution is decreased. There is also a remarkably similitude between the number of FLOPs between tiny-YoloV3 and SSD Mobilenet, although empirically "Darknet" [110], the GPU framework into which YoloV3 is built, seems to be lighter than Tensorflow [143], hence the faster frame-rate observed in Fig. 5.9.*

Chapter 6

Visual saliency using bio-inspired decision making for region selection

6.1 Introduction

Recently, substantial deep neural network approaches have been proposed to perform visual saliency estimation [145, 146], however these methods are often end-to-end rather than modular, modality being an important feature of the human visual system that may add redundancy, robustness and flexibility. On the other hand, visual saliency in robotics has been performed using the popular bottom-up method of Itti et al. [1] for a few decades, but since these are bottom-up approaches, i.e. based on image movement, intensity or colour in independent images, not permitting the use of high-level attention mechanisms related to tasks or goals. There is still a gap in designing a visual saliency system for robots that incorporates both bottom-up and top-down information over time, making it robust for intrinsically salient objects (bottom-up), and adaptable to a given task (top-down information). To address this gap, this chapter presents a bio-inspired scheme based on a standard model of human visual saliency, extend primarily by introducing a computational model of the basal ganglia, in order to make a smoother behaviour over time and make it more robust to noise.

The simple additive fusion of top-down and bottom-up saliency, as described by Kimura et al. [52], is less robust to noise because it uses only the present time-step value of each saliency stream to perform the top-down and bottom-up information merging (equivalent to winner-takes-all). This can cause the fovea to jitter about the visual field in the presence of noise. In the human brain it is thought that the basal ganglia acts as a central device that accumulates saliency evidence over time to take robust decisions [147]. Therefore, the robustness problem is addressed here by using a simple computational model of the basal ganglia [72, 73], that has been shown to be related to the multi-hypothesis sequential probability ratio test (MSPRT) for decision making [98], drawn in Fig. 6.1. The MSPRT is a computationally lightweight model of basal ganglia decision making function, not a detailed biophysical model such as in [94, 95, 148], but well suited to robotic systems for its computational efficiency. The decision making algorithm uses evidence accumulation of the saliency over time and threshold testing to decide which region is most salient, meaning that the saliency decision making is based on an accumulation of data and is therefore inherently more robust to noise.

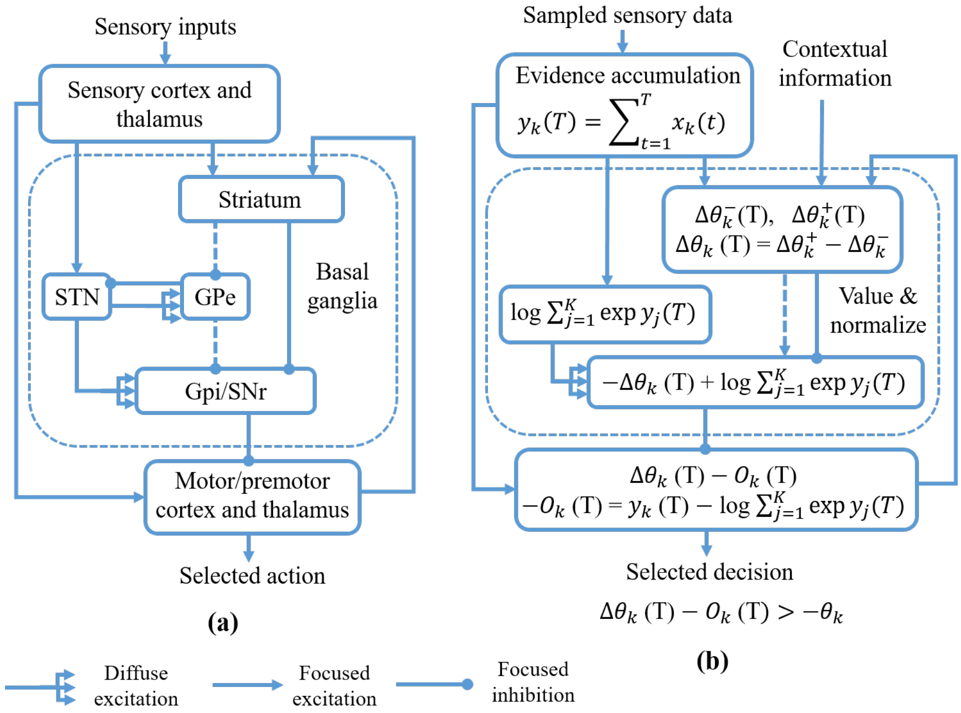

Sensory inputs
Sensory cortex and thalamus
Striatum
STN
GPe
Basal ganglia
Gpi/SNr
Motor/premotor cortex and thalamus
Selected action
(a)
Sampled sensory data
Evidence accumulation
$y_k(T) = \sum_{t=1}^{T} x_k(t)$
Contextual information
$\Delta\theta_k^-(T)$, $\Delta\theta_k^+(T)$
$\Delta\theta_k(T) = \Delta\theta_k^+ - \Delta\theta_k^-$
$\log \sum_{j=1}^{K} \exp y_j(T)$
Value & normalize
$-\Delta\theta_k(T) + \log \sum_{j=1}^{K} \exp y_j(T)$
$\Delta\theta_k(T) - O_k(T)$
$-O_k(T) = y_k(T) - \log \sum_{j=1}^{K} \exp y_j(T)$
Selected decision
$\Delta\theta_k(T) - O_k(T) > -\theta_k$
(b)
Diffuse excitation
Focused excitation
Focused inhibition

Figure 6.1: *Map between the basal ganglia and the cerebral cortex and sequential analysis:* ***(a)*** *Cortico-Basal ganglia connectivity;* ***(b)*** *Representation of the decision making process. Redrawn from [73, Fig. 2].*

Akbas and Eckstein [32] presented a model matching the detection performance on full resolution images with foveal ones in about five exploratory fixations. It is estimated that humans make around three eye movements every second [32]. The duration and type of each movement varies significantly, the present chapter aims to study the implementation of a decision making process for said eye movements.

Despite having a good understanding of the different type of eye movements, there is no current agreement in the literature of what and how they are triggered, neither of how saliency maps fit exactly in the human attention system. It is not feasible for our eyes to track every immediate stimulus that is presented in the human visual field. For our eyes and head to make coherent and smooth movements, an evidence accumulation system has to be in place. When representing eye movements from human data, studies use relatively simple stimulus [149], for the viability of implementation.

An engineering implementation is taken here, testing how the MSPRT hypothesis compares to human data in a contemporary database of eye tracking of users viewing a series of omnidirectional videos. Testing this hypothesis serves a double goal; having a large field of view that requires both head and eye movements magnifies the need of changing the point of view and, from a robotics perspective, addressing the challenge of omnidirectional camera analysis in lightweight embedded systems.

The visual system presented is tested and evaluated here on two types of camera: aerial point of view images (as in Section 4.3) and omnidirectional. For the latter, using an equirectangular projection dataset that includes human eye fixations (location and duration) [150]. These type of videos are typically preserved in an equirectangular projection covering the horizontal field of view of 360° and of 180° vertically. It would be computationally infeasible to process the entire projection in a CNN, and selecting a sub-region from the field of view to transform to rectilinear coordinates is far from a trivial question. The use of deep neural networks represents a large computational bottleneck for robotic systems, the visual saliency present here is adapted efficiently tackle this problem, with an approach where regions of the omnidirectional image are only processed by the CNN once enough saliency evidence is accumulated by the MSPRT algorithm for a given region.

6.2 Methods

Given the large number of models that are present in the literature aiming to establish a functional scheme of the brain and visual processing, it is difficult to present an unanimous agreement of its connectivity. One prominent theory, often referred to as the "two-streams hypothesis" [67], is that the brain has two distinct processing pathways; the dorsal and ventral streams. The former focuses on where objects are located in space and in action planning [68], often called "where" stream. The latter stream is associated with object categorization, also known as the "what" stream. The streams work for both visual and auditory information. Fig. 6.2 presents a diagram of the analogy between the current model and the connectivity of regions in the macaque brain and their functionality, derived from the schematic representation of brain functions presented in Fig. 2.10.

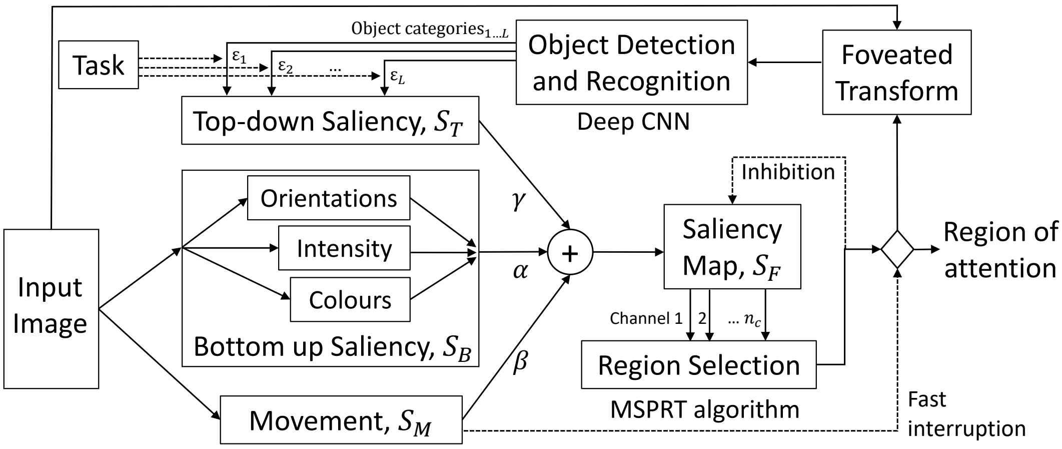


Figure 6.2: *Full scheme of evidence accumulation and information flow.*

The complete model of visual attention has the following components; each frame from the camera is processed by a bottom-up pathway to extract low-level features such as orientations, colour and intensity, as well as movement, emulating low-level processing in the thalamus and visual cortex. The image undergoes a foveated transform that is processed in a top-down pathway by a CNN object detection and recognition system, emulating the "where" and "what" high-level processing in the ventral and dorsal pathways of the brain. Thus, task relevant information is embedded in terms of the discrete label categories that the CNN is trained to detect. The resulting saliency maps of the top-down and bottom-up pathways are additively fused using a weighted average. The resulting saliency of each image region is transmitted to the MSPRT algorithm to select the most salient region, emulating basal ganglia function. The selected region modifies a final sensorimotor map, emulating the sensorimotor map in superior colliculus that directs gaze. The peak in the sensorimotor map defines the direction of the fovea. A fast pathway interrupt is also included, which links low-level image processing movement detection to the final sensorimotor map.

6.2.1 Bottom-up saliency

The same bottom-up saliency model Frintrop et al. [116] described in previous chapters was used, where it is averaged from orientation, intensity and colour conspicuity maps,

$$S_B = \frac{1}{3}\left(C_O + C_I + C_{RG} + C_{BY}\right) \tag{6.1}$$

Several approaches have been proposed to adapt saliency implementations to omnidirectional cameras; most of these are inherently computationally heavy, e.g. by (i) using deep convolutional neural networks [151, 152], (ii) making several rectilinear projections from the image, obtaining their saliency and fusing them back to an equirectangular one [153], or (iii) calculating several complementary approaches and then aggregating them into a global saliency [154]. It is opted here to use, for the omnidirectional frames, the same bottom-up saliency method based on the computational model presented in [116], i.e. Vocus2, by making the adaptation from having a prior, or bias (λ), towards the equator instead of the central point, as illustrated in Fig. 6.3, along which the vast majority of human fixations fall [155]. Using Vocus2 [116], the bias is defined at $\lambda_{eq} = 5 \times 10^{-5}$. The rest of the major parameters are set to: $\sigma_{center} = 1$, $\sigma_{surround} = 2$, four stop layers and the use of the arithmetic mean for feature and conspicuity fusion, which leads to lightweight bottom-up saliency for the omnidirectional camera.

For full equirectangular image representations, bottom-up saliency fits into this approach by first making a global bottom-up estimation using a normally downscaled frame. This greatly speeds up the process (using a downscale ratio of 5:1, to bring the frame to Wide-VGA resolution).

6.2.2 Movement saliency

Movement is obtained using a computationally lightweight method, simply as the Pythagorean distance between the current frame f_n and the previous one f_{n-1}, over the three colour channels red, green and blue,

$$S_M = \frac{\sqrt{\Delta_R^2 + \Delta_G^2 + \Delta_B^2}}{\sqrt{\max(\Delta_R)^2 + \max(\Delta_G)^2 + \max(\Delta_B)^2}} \tag{6.2}$$

where $\Delta_R = f_{R,n} - f_{R,n-1}$ is the difference in the red channel between frames at time-steps n and $n-1$ (and similarly for the blue and green channels), and $\max(\Delta_R), \max(\Delta_G), \max(\Delta_B)$ is the maximum color difference in red, green and blue channels respectively, so typically $\max(\Delta) = 255$.

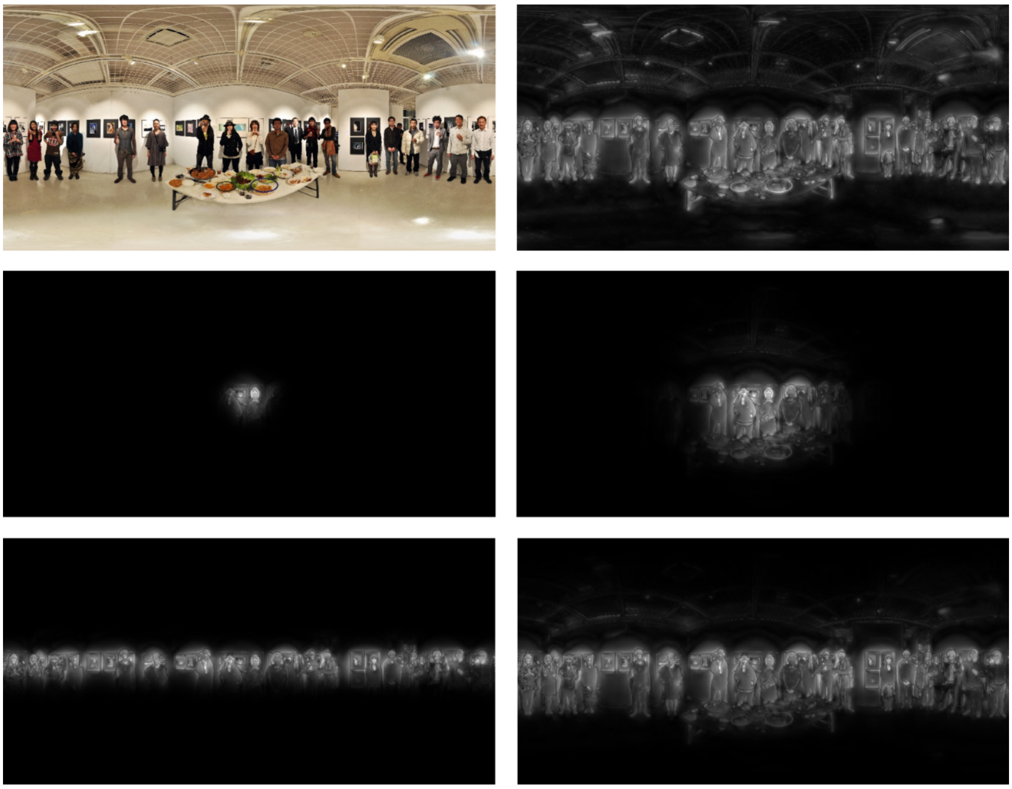


Figure 6.3: *Example of an equirectangular frame and its saliency estimation using the Vocus2 system [116]. From top to bottom and left to right;* ***(a)*** *Sample frame,* ***(b)*** *Saliency without central or equatorial bias ($\lambda = 0$),* ***(c)****, with a central bias $\lambda_{cr} = 5 \times 10^{-4}$* ***(d)****, with a central bias $\lambda_{cr} = 5 \times 10^{-5}$* ***(e)*** *with an equatorial bias $\lambda_{eq} = 5 \times 10^{-4}$, and* ***(f)*** *with an equatorial bias $\lambda_{eq} = 5 \times 10^{-5}$. Qualitatively, the last one was chosen as the best fit for the stimulus present in the used dataset.*

6.2.3 Object detection

Two popular state-of-the-art single pass detector implementations are used in the object detection-recognition stage; YoloV3 [105] and SSD Inception [137]. The former is designed with successive, repeating blocks, where each block is composed of a 1×1 convolutional layer, followed by a 3×3 convolutional layer, and a residual layer. Blocks are repeated numerous times with occasional shortcut connections, followed by average pooling then a fully connected layer with softmax output. YoloV3 uses dimension clusters as anchor boxes to predict the object bounding boxes along with the class label [105]. The system outputs 4 coordinates to define each bounding box: the centre coordinates of the box (x, y), the width, w, and height, h. The loss function for each of these bounding box regression variables is defined as the sum-of-squared error. The class label prediction is done for the objects contained in each bounding box using multilabel classification, which is trained using a binary cross-entropy loss function.

For the drone dataset, the model is retrained for resolutions from 96 × 96 to 416 × 416 pixels. The drone frames are also injected with artificial noise (Eq. 6.9) to test its resilience to it, evaluated as mean Average Precision (mAP). For omnidirectional frames, tiny-YoloV3 [105] is used as provided by the original author, trained on 80 categories from the COCO dataset [104]. Such categories are fairly generic, it is expected that for the 19 videos of omnidirectional video dataset [150], the most common occurrence is of "person", also true in the COCO data. The former does not provide ground truth object categorisation, but visual inspection and consistency are used as part of the analysis in Section 6.5.

Ideally mask bounding boxes could be used to have more rigorous feedback, based on works like [156], but the computational cost increase is deemed too high to outweigh the benefits, effectively prohibiting the system to be run on a Nvidia Jetson Nano.

Overall top-down saliency influence S_T is defined as the sum of all detections from the current frame f_n plus detections from previous ones curtailed by a factor of a half for each time-step

$$S_T = \sum_{k_a=1}^{m_n} R_{k_a} + \frac{1}{2}\sum_{k_b=1}^{m_{n-1}} R_{k_b} + \frac{1}{4}\sum_{k_c=1}^{m_{n-2}} R_{k_c} + \ldots + \frac{1}{32}\sum_{k_c=1}^{m_{n-5}} R_{k_c} \tag{6.3}$$

where m_n is the number of bounding boxes found on frame n, and $R_{k=[1\to m_n]}$ are all the bounding boxes for that corresponding frame.

6.2.4 Target selection using evidence accumulation with MSPRT

The fused saliency map S_F is calculated as the weighted average of the movement saliency map S_M, bottom-up map S_B and top-down map S_T,

$$S_F = \alpha S_B + \beta S_M + \gamma S_T \tag{6.4}$$

where α, β and γ are weights that can be tuned to adjust the influence of each saliency map, so that $\alpha + \beta + \gamma = 1$. Here their values are set as $\alpha = \beta = 0.4$ and $\gamma = 0.2$, empirically finding them to give a coherent behaviour. The top-down weight γ is tuned to be smaller than the other weights because the top-down influence consists of filled rectangular bounding boxes for every object detected with confidence ≥ 0.5, thus the bounding box tends to be larger than the actual detected object, necessitating a reduction in the weight.

As previously described, a MSPRT decision making model of the basal ganglia [72] is used to perform region selection in the fused saliency map S_F. The MSPRT algorithm is adapted to take discrete channels of salience as input and accumulate evidence for each channel until a threshold is reached and a decision is made (note that the algorithm derived in [72] is only strictly an MSPRT if certain conditions are met, such as that the inputs follow a Gaussian distribution, although, relaxation of these assumptions have been considered elsewhere [73]).

To implement the MSPRT algorithm, a small number of discrete input channels is required, but the fused saliency map, S_F, consists of many contiguous pixels of the same size as the input image. Therefore, a small number, n_c, of candidate regions in S_F is used as the discrete input channels. It would be possible to use fixed, static regions by dividing the image into e.g. a 3×3 grid ($n_c = 9$), or dynamic regions where the most salient regions at the current time-step are chosen. It uses the dynamic region approach for the standard camera, where the n_c most salient regions are chosen at each time-step. For the omnidirectional camera, fixed regions are used corresponding to locations from an equirectangular projection described in subsection 6.2.6. For both camera types, a binary mask M is used to extract the region of interest, which is illustrated in Fig. 6.4 for the omnidirectional camera.

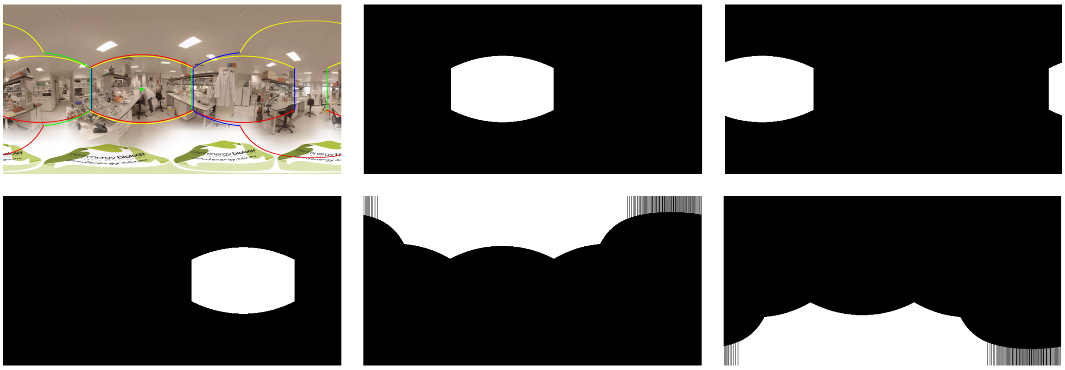


Figure 6.4: *Sample of region masks M_i; (from left to right)* ***(a)*** *Sample frame,* ***(b)*** *mask M_0 for region centred at $[\lambda, \phi]$,* ***(c)*** *mask M_1 for region $[\lambda - 0.3, \phi]$,* ***(d)*** *mask M_2 for $[\lambda + 0.3, \phi]$,* ***(e)*** *mask M_3 for region $[\lambda, \phi - 0.4]$,* ***(f)*** *mask M_4 for region $[\lambda, \phi + 0.4]$. The borders of the M_i masks are pre-saved on a lookup table (LUT) for all possible locations with a 0.01 resolution for λ and ϕ.*

To implement the MSPRT for region selection, the fused saliency map, S_F, is divided into n_c discrete regions corresponding to the number of input channels, and a channel i is selected (disinhibited in the context of the basal ganglia) if the output O_i, from accumulated evidence in channel i, crosses a fixed threshold Θ;

$$O_i(T) = -y_i(T) + \log \sum_{j=1}^{n_c} \exp\left(y_j(T)\right) \text{ for } i = 1, \ldots, n_c \tag{6.5}$$

given $y_i(T) = gY_i(T)$ (g is a scaling parameter here set to $g = 1$), and

$$Y_i(T) = \sum_{t=t_0}^{T} s_i(t) \text{ for } i = 1, \ldots, n_c \tag{6.6}$$

where $Y_i(T)$ is the accumulated evidence in channel i, between frames $[t_0, T]$ and s_i is a scalar value of saliency obtained from summing over a region $S'_{F,i}$, which is the i-th region of S_F extracted using a binary mask M_i.

$t_0 = T - 25$ is taken so that $Y_i(T)$ takes as evidence the stimuli present in the last ~ 1 second (assuming a frame-rate of 25 - 30 FPS). Note that the threshold Θ is a hyperparameter that must be tuned to give effective performance, typically in a speed-accuracy sense (i.e. faster decisions with less accuracy or slower decisions with more accuracy).

In order to incorporate an inhibition of return (IOR) influence, which partially impairs a winning region once it crosses the threshold Θ, an influencing factor Ω_i is added to each evidence channel in Eqn. (6.6),

$$Y_i(T) = \Omega_i \sum_{t=t_0}^{T} s_i(t) \tag{6.7}$$

where $\Omega_i = 1$ for a channel that has not been selected and $\Omega_i = 0.5$ for a channel that has just been selected, returning linearly over time to $\Omega_i = 1$ (over an interval of 10 time-steps here, enough to ensure that the algorithm does not become stuck in a single channel, while not completely blocking it). An illustrative scheme of the full information flow in conventional images is presented in Fig. 6.5.

6.2.5 Foveation

Once channel i has crossed the threshold and is deemed to contain enough salient information, $Y_i(n)$ is transformed into the rectilinear projection $Y_i(n) \Leftrightarrow \widehat{Y}_i(n)$, following [157]. Subsequently, the top salient location of that region is taken as the central point to foveate;

$$f_0[\lambda, \phi] = \max(S_{overall}) \in \widehat{Y}_i(n) \tag{6.8}$$

The foveation process described in the previous chapter is thus followed, making $\widehat{Y}_i(n) \Rightarrow F(n)$, as illustrated in Fig. 6.6. The foveated frame $F(n)$ is then passed through the CNN, providing the R_k bounding box detections described in Eqn. 6.3 as feedback for frame $n + 1$.

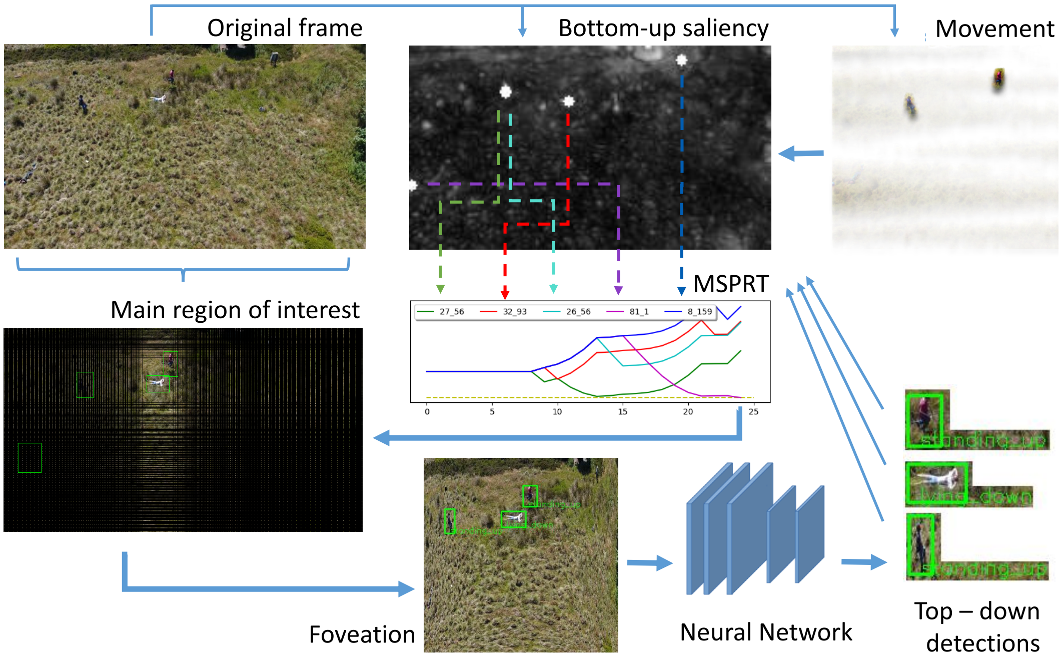


Figure 6.5: Full MSPRT example for conventional images

6.2.6 Equirectangular field of view

Transforming from an equirectangular to rectilinear projection (where straight lines are displayed undistorted), and then performing object detection, is a computationally heavy operation. But even if the former can be considered simply an implementation requirement, the latter (or the combination of both) presents the biggest bottleneck, in a comparable sense in which the human eye can only target the fovea, thus identify objects with certainty in a very small region. There is also an equivalence in how the eye and head movement can not be guided by the immediate prominent stimulus in every instant, given movement constraints, both for a human and for a mobile robot.

Functionally, projecting from an equirectangular representation to a rectilinear one is restricted to a section of about 0.3 of the equirectangular frame along the equator. Since 1.0 of the figure is the full 360°, a region of 114° represents; $114 \div 360 = 0.3166$. With help of the Mutha [157] Toolbox, a sub-region of this size of the 4K equirectangular frame is transformed to a frame in the range of Wide-VGA resolution (768×384 pixels). Although for certain applications, this image size might be enough to run through object detection, depending on the GPU capability, here is taken a step further and foveated in the most promising region of interest, enabling the system to run on low power GPUs, which is here exemplified using the tiny-YoloV3 [105] at 224×224 input resolution.

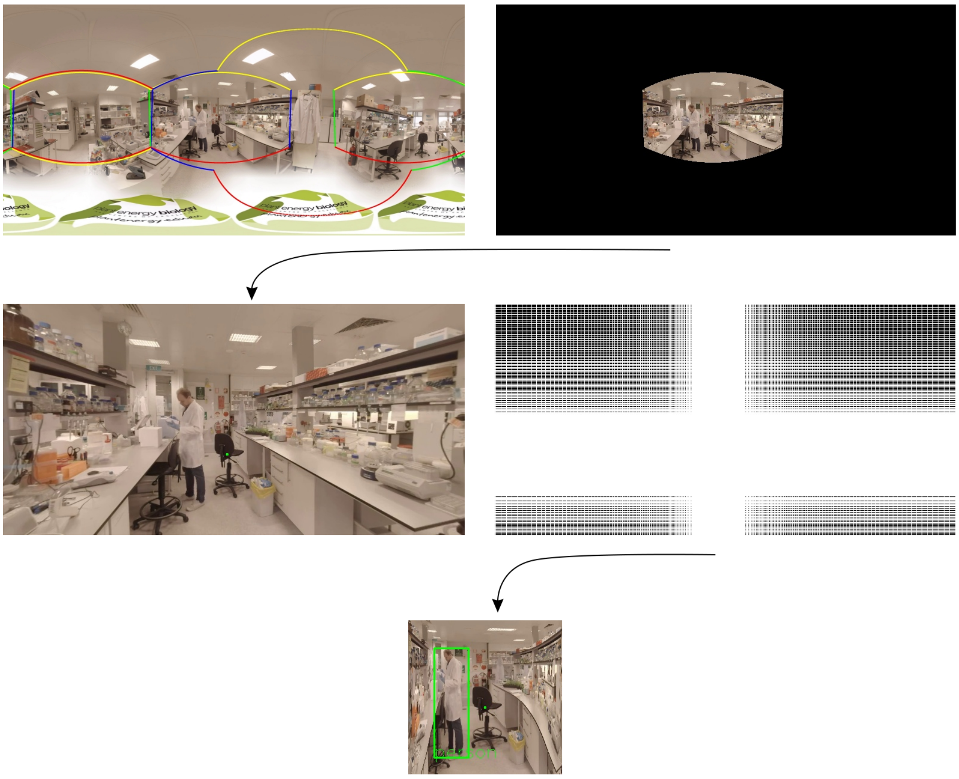


Figure 6.6: ***(a)*** *Equirectangular sample frame (at* 3840 × 1920 *resolution),* ***(b)*** *masked region of interest by the MSPRT,* ***(c)*** *rectilinear projection of the area of interest (at* 768 × 384 *resolution), a small dot in the chair near the centre indicates it to be the most salient point of that region.* ***(d)*** *Selected rows and columns, following the method detailed in [152], to foveate the image.* ***(e)*** *Foveated frame at* 256 × 256 *resolution, with the bounding box of the tiny-YoloV3 [105] prediction, run at the same resolution.*

The information flow described in the previous paragraph also helps to sustain the analogy to the human vision system, which is continually faced with the same dilemma. The average human field of view has an approx. 210° forward-facing horizontal range, and a 150° vertical one [158]. Although the high acuity region, known as fovea centralis, is concentrated in a region of about 5°, and the foveola in 1°, with the highest visual acuity. Hence, humans constantly choose from an unbounded, often unstructured environment, the region to fit into our field of view (by body/head movements) and subsequently, fixate our fovea into small targets (by eye movements), leaving the majority of the region in our peripheral vision.

Following this train of thought, three consecutive 114° regions along the latitude would cover most of the 360° field of view, depending on where the central point is located. Equivalently three 57° (conserving the 2:1 image ratio) regions cover most of the 180° longitude field of view. As exemplified in Fig. 6.7, with

$[\lambda_0, \phi_0] = [0.5, 0.5]$ (i.e. the centre of the frame) the area that can be easily transformed to a rectilinear representation is delimited by the yellow border.

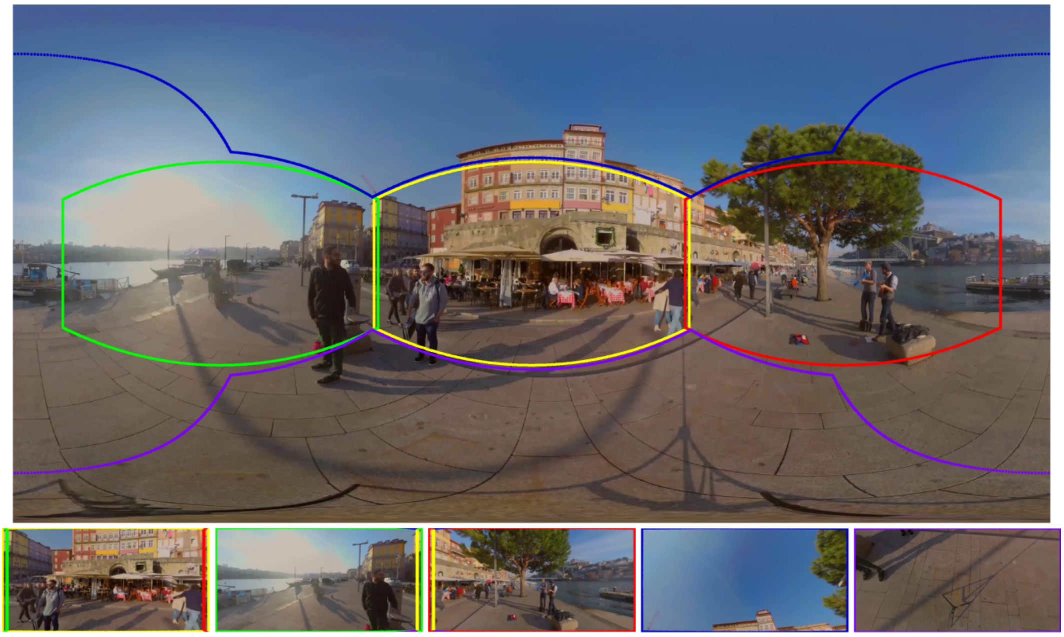


Figure 6.7: ***(Top)*** *Example of an equirectangular frame, at* 3840×1920 *pixel resolution.* ***(Bottom, from left to right)****,* ***(a)*** *Rectilinear projection centred in [λ (horizontal coordinate system), ϕ (vertical coordinate system)] = [0.5, 0.5], with a resolution of* 768×384 *pixels, and representing about* 114° *in latitude, as illustrated by the yellow borders in the equirectangular frame* ***(b)*** *Projection centred at [0.2, 0.5], green region in the equirectangular frame* ***(c)*** *Region centred in [0.8, 0.5] and illustrated by the red border* ***(d)*** *Centred in [0.5, 0.1] and bordered in blue* ***(e)*** *Centred in [0.5, 0.9] and bordered in purple.*

Moving the focus point to $[\lambda_0 - 0.3, \phi_0]$ and $[\lambda_0 + 0.3, \phi_0]$ would produce the regions delimited by green and red borders, respectively. A similar process can be done to transform the blue and purple bordered regions by centring in $[\lambda_0, \phi_0 - 0.4]$ and $[\lambda_0, \phi_0 + 0.4]$ accordingly. This is a coarse rule where, due to the equirectangular distortion, a different amount of overlap is caused depending on where the central point is, as shown in Fig. 6.8. Even with this overlap, most of the scene is covered by the five regions, particularly along the equator, where the vast majority of fixation will fall [150].

By bordering the omnidirectional scene in such a way, it can be approached with a divide-and-conquer outlook. By first exploring the saliency of the full scene (a computationally lighter process), and transforming to rectilinear only the subsection and once enough saliency has been accumulated for it. Similar to previous chapters, the bottom-up saliency estimation is here performed using the Frintrop et al. [116] saliency model.

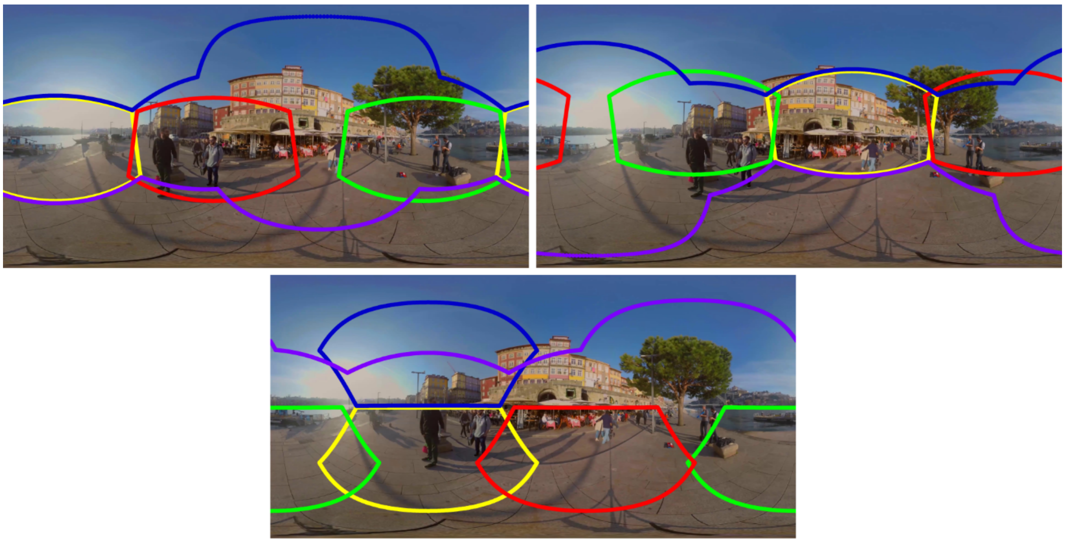


Figure 6.8: Equirectangular frame with the central location (yellow region) at ***(a)*** $[\lambda, \phi] = [0.1, 0.55]$ ***(b)*** $[\lambda, \phi] = [0.6, 0.45]$ ***(c)*** $[\lambda, \phi] = [0.3, 0.7]$*. The subsequent regions (left, right, up and down, relatively to the central one) are fixed at* $[\lambda - 0.3, \phi]$*,* $[\lambda + 0.3, \phi]$*,* $[\lambda, \phi - 0.4]$*,* $[\lambda, \phi + 0.4]$ *respectively.*

Following the same rationale from previous chapters, given the pyramidal structure of the saliency model, bottom-up saliency is computed in a normally downscaled frame, by a magnitude of five to bring the 4K omnidirectional frame to Wide-VGA resolution). By using the image borders previously described, five regions are used as channels for evidence accumulation; the central one (where the current fixation is positioned), left, right, up and down. A full sample diagram of the information flow in omnidirectional images is shown in Fig. 6.9.

6.3 Experimental Data

6.3.1 Standard camera: drone dataset

The aerial image dataset as described in Section 4.3 was also used here. For training the tiny-YoloV3 and SSD CNNs in object detection and recognition, 4804 frames where manually labeled, from 4 different scenes, and extended to 24,020 using the *Imgaug* image augmentation library [124], with transformations including Gaussian blur and noise, contrast normalization, rotation and flipping.

To test the effectiveness of the MSPRT algorithm in making the visual saliency more robust, synthetic noise is added to the images. Gaussian noise $p(n_{(x,y)})$ was

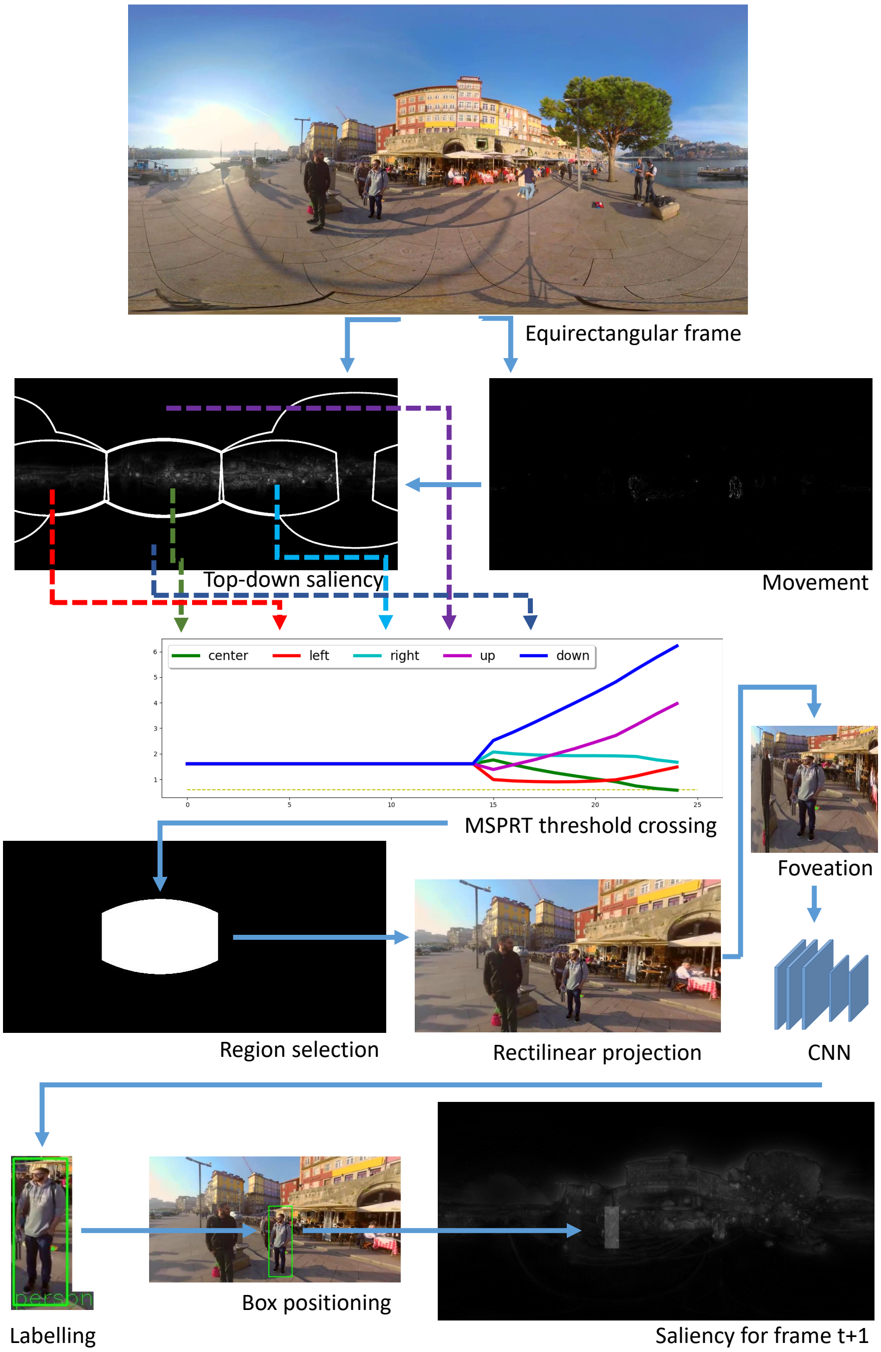


Figure 6.9: *Full MSPRT example for omnidirectional images*

added at every pixel $n_{(x,y)}$, with mean and standard deviation; $\mu = 0$ and $\sigma = 0.1$,

$$p(n_{(x,y)}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(n_{(x,y)}-\mu)^2}{2\sigma^2}} \tag{6.9}$$

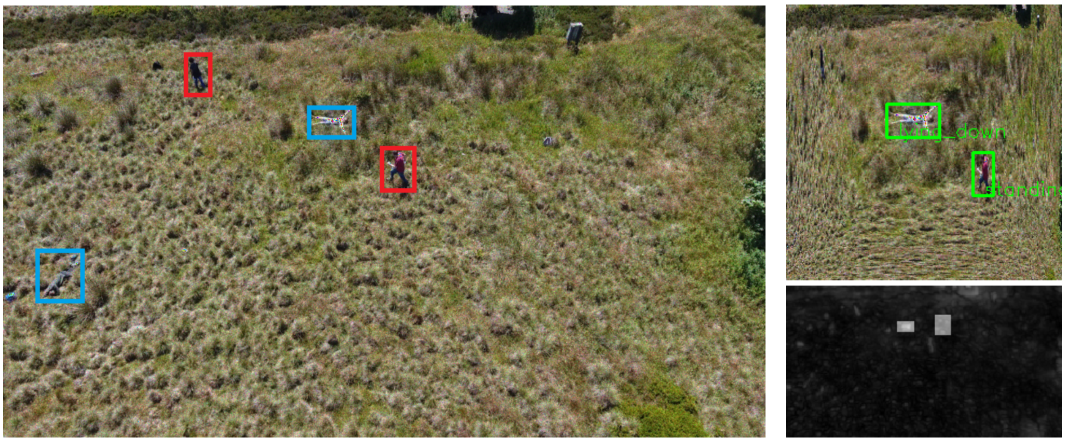
so that a new frame $F(n_{(x,y)})$ is defined by $F(n_{(x,y)}) = F_0 + p(n_{(x,y)}) * F_0$.

The task-specific goal of finding a person lying down was introduced into the top-down saliency map by defining whether each pixel falls within the detected object,

$$S_T = \begin{cases} n_{(x,y)} = \epsilon_L & \text{if} \in \text{Bounding box of detection,} \\ n_{(x,y)} = 0 & \text{if} \notin \text{Bounding box.} \end{cases} \tag{6.10}$$

where ϵ_L in an task-specific influencing factor for object category L, to allow prioritizing depending on the task.

In the UAV data, note that a person walking would likely be more salient than a person lying down due to the motion dominating the saliency computation. The use of this task-specific biasing enables the robot to focus on the more relevant region of the scene, this effect is illustrated in Fig 6.11.



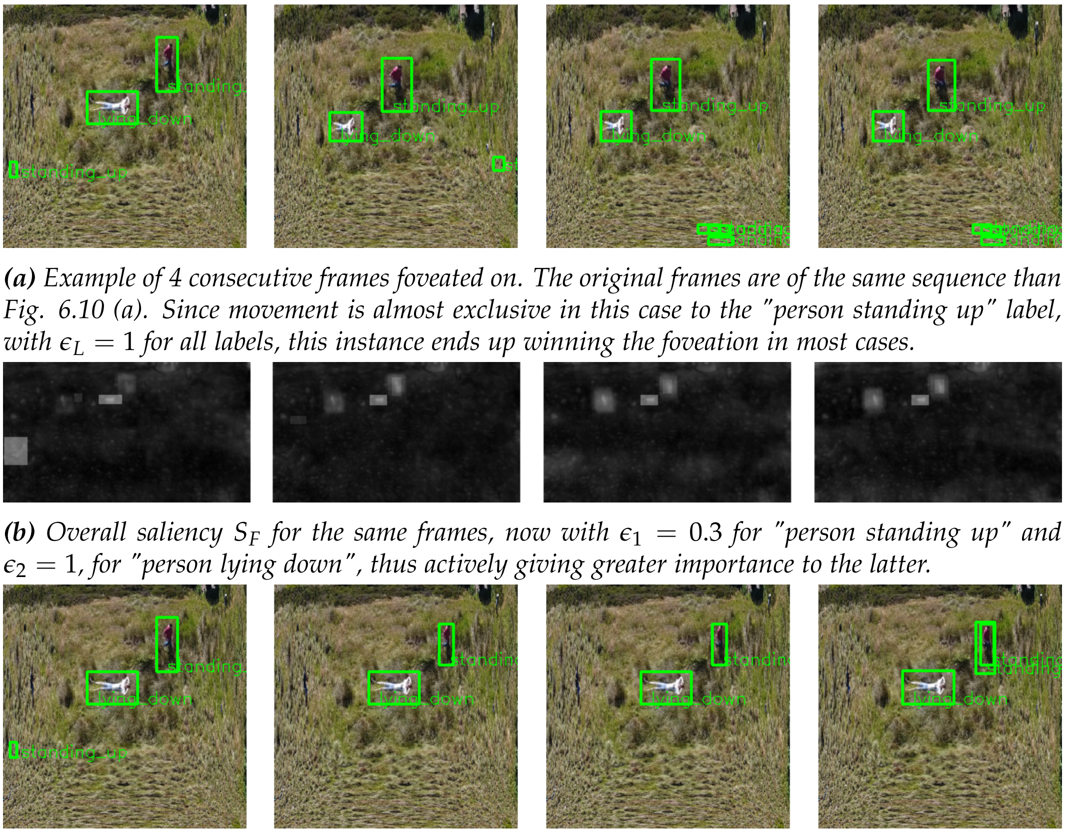
(a) Sample frame from the drone dataset, with two instances of "person lying down", in blue bounding boxes, and two instances of "person standing up", in red bounding boxes, with added artificial noise as described in Eq. 6.9

(b) Foveal image and top-down gained information added to the overall saliency

***Figure 6.10:** Sample frame and object detection performed on its foveated representation.*

6.3.2 Omnidirectional camera: Eye movements for 360° videos

The David et al. [150] Omnidirectional dataset that was used consists of 19 videos, each of 20 seconds in equirectangular format. With frame rates between 24 - 30 FPS, observed by an average of 49.63 participants (median = 50), it amounts to over half a million fixated-on frames. With this data, the aim is to get an overall behaviour comparison between human fixations and the proposed visual saliency system, whilst also demonstrating how the regions of interest selected by the system (and foveated on) proffer a good strategy to select visual information



(a) *Example of 4 consecutive frames foveated on. The original frames are of the same sequence than Fig. 6.10 (a). Since movement is almost exclusive in this case to the "person standing up" label, with $\epsilon_L = 1$ for all labels, this instance ends up winning the foveation in most cases.*

(b) *Overall saliency S_F for the same frames, now with $\epsilon_1 = 0.3$ for "person standing up" and $\epsilon_2 = 1$, for "person lying down", thus actively giving greater importance to the latter.*

(c) *Foveation caused by the effect of prioritizing the ϵ_2 channel, resulting on it winning the foveation in most cases.*

Figure 6.11: *Sequence of frames foveated on, exemplifying the effect of switching from treating all top-down information equally (top sub-figure), to giving task oriented precedence to the label "person lying down", $\epsilon_2 = 1$ and $\epsilon_2 = 0.3$ to the rest of the label categories, illustrating the task of a search and rescue operation (bottom sub-figure).*

to run through a light-weight CNN object detection system, without it requiring any special modification or retraining.

As a point of interest, the distribution of the duration of fixations (in terms of frames) is compared between the dataset of human fixations and those produced by the MSPRT system. To ensure that both systems are looking at similar data (since neither a human observer nor the present system look at the whole 360° at a time), the fixation point $[\lambda, \phi]$ from the observers in the dataset is followed, also deriving the rest of the I channels from it. The MSPRT threshold cross $O_i(n) < \Theta$, triggers a new foveal transformation and CNN detection feedback.

6.4 Evaluation methods

6.4.1 Object detection and recognition

One main point to test is the hypothesis that the MSPRT evidence accumulation system provides stability to foveation selection. A simple assessment is to compare the effectiveness of object detection (in terms of mean Average Precision, mAP) if it is performed at every frame or only when the evidence accumulation threshold is passed. Furthermore, it is expected that by inserting noise to hinder precision, the effect is even more evident. A significant level of noise is expected to debilitate saliency estimation, and thus the location of the fovea, switching its position erratically between every frame.

As described in previous sections, the first step is to calculate the Intersection over Union (IoU) between the ground truth bounding box and the prediction box,

$$\text{IoU} = \frac{\text{Area of overlap}}{\text{Area of union}}$$

and then use it as a threshold to determine if a predicted box can be considerate positive. The mean Average Precision (mAP) is then calculated using the metrics of the PASCAL VOC 2012 competition [130], with an IoU of at least 50%. Performance is measured of the instances where the objects are at least 30% into the foveal region, as used in the previous chapter, an empirically found threshold that provides a good valance between good performance and *accepting* as many objects as possible.

Fig. 6.10 (a) shows a frame with artificial noise as described in Eq. 6.9. While Fig. 6.10 (b) illustrates a basic example of evidence accumulation; the top salient points are tracked as input into the MSPRT. When one of them crosses the threshold, an object detection is triggered. The detected bounding boxes of detections are then used as feedback into the overall saliency.

6.4.2 Duration of fixations

The Kolmogorov-Smirnov Chi-Square *K-S* and Chi-Square χ^2 are two *goodness-of-fit* test values (for statistical hypothesis testing), that are commonly used to analyse visual saliency algorithms. Given the histograms of duration of fixations, for the ground truth and the presented implementation, H_1 represents the distribution of duration of human fixations, and H_2 represents the duration of MSPRT fixation, both with a bin-width equal to 1, and considering $m = 45$ bins. Then the *K-S* and

χ^2 are defined as [159],

$$\text{K-S} = \max_{1 \leq i \leq m} | \sum_{i=1}^{m} H_1(i) - \sum_{i=1}^{m} H_2(i) | \tag{6.11}$$

$$\chi^2 = \sum_{i=1}^{m} \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)} \tag{6.12}$$

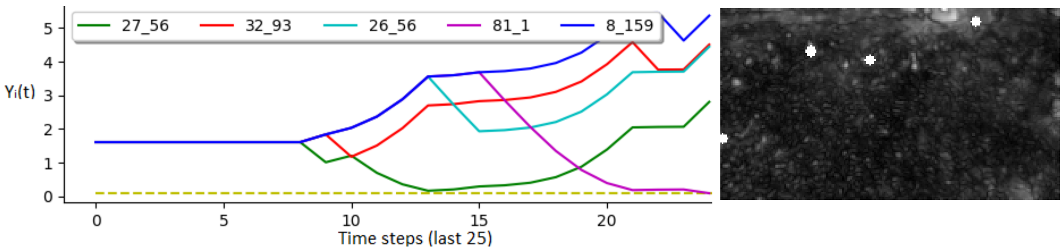
6.5 Results

In this section the goal is to evaluate the visual saliency system using a standard camera on a drone performing a search-and-rescue type task, with a main interest on omnidirectional camera focusing on comparing to human visual saliency. Note that the drone dataset is well suited to evaluate factors like biasing of top-down saliency in searching for potentially injured people, whilst the omnidirectional data is well suited to comparing with human visual saliency because it includes data on human fixations.

6.5.1 Object detection and classification performance using the aerial images

Fig. 6.12 shows the MSPRT curve for the five top salient locations. The value of their saliency is accumulated over the last second, in this example at the top five locations from the saliency frame (right side figure). Fig. 6.13 plots the mAP performance across neural network resolutions and foveated image size from 416 × 416 to 160 × 160 pixels using the YoloV3 (left side figure) and SSD models (right figure), showing a moderate performance increase. However, more interestingly, allowing a considerable jump in frame rate, by bypassing the frames that do not show a significant change. This also serves the purpose of giving better resilience to noise, as defined in Eq. 6.9, especially true at smaller resolutions. The jump in frame rate can be observed as the difference between the dotted lines in the two bottom plots of Fig. 6.13, while the change in resilience to noise gives the better performance in the bottom two plots, in contrast to the top two plots, in the same Fig. 6.13.

Object detection performance is also empirically shown on the omnidirectional dataset, showing a big advantage to foveating over different segments of the frame, in contrast to trying to perform detection on the full frame. Fig. 6.14 illustrates how for YoloV3 and tiny-YoloV3 at lower resolutions, the system fails to make detections, partially because of the omnidirectional distortion (although most objects are near the equator, where the distortion is less) and partially because the



(a) MSPRT graph example (unit-less, from Eq. 6.5).

(b) Top salient locations.

Figure 6.12: *Sample of downwards threshold crossing from the MSPRT for the top salient locations (shown in the right side figure). When the evidence of a salient point passes the threshold, the evidence of all of them is reset. It can be seen that in this case, the current aggregation has been active for the last 16 frames, until the purple channel is foveated on (after downward crossing the light green dashed line, representing the fixed threshold.*

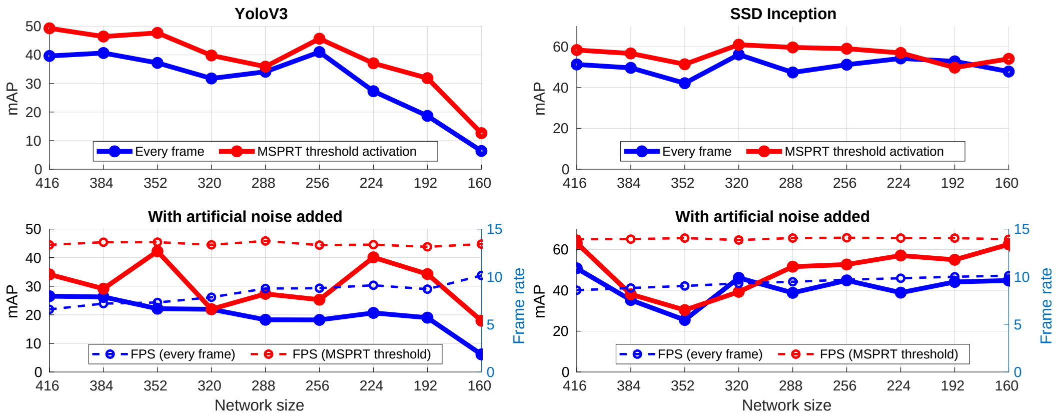
wide angle representation ends up making most objects too small to be found by a lightweight object detector. One important point to note is that by varying the location of the fovea, different but complementary detections are obtained, in contrast to performing detection at every frame at a high resolution, where an undetected object will remain so, even if the algorithm is run several times.

6.5.2 Comparison to human fixation on omnidirectional videos

In this section it is tested how the system compares to human fixation behaviour, an interest more in line with saliency map modeling. A simple first test is to study the histograms of the duration of fixations, using the human recorded data provided for the omnidirectional dataset [150]. Fig. 6.15 shows an histogram of the distribution of duration for the first five videos of the dataset. Tables 6.1 and 6.2 compile the Kolmogorov-Smirnov (*K-S*) and Chi-Square (χ^2) *goodness-of-fit* test values, two commonly used measures of statistical hypothesis testing. Here for the two histograms; H_1 for the distribution of duration on human fixations, and H_2 for the duration of MSPRT fixation, both with a bind-width equal to 1, and considering $m = 45$ bins.

With total average values of $\overline{\text{K-S}} = 0.269$ and $\overline{\chi^2} = 0.284$, illustrating the similarity for most videos between both histograms, noting that if $H_1 = H_2$, then $\text{K-S} = \chi^2 = 0$. Evidently the proximity between the histograms can be brought closer for each video, by parameter tuning, if it were the sole goal.

Another relevant aspect to evaluate is how often top salient locations are fixated-on, both in contrast to human fixations and to a simple Winner-Takes-All (WTA) rule. Fig. 6.17 illustrates how the WTA fixates on a small number of prominent locations (the larger the circle is drawn represents the number of times



(a) YoloV3 model performance.

(b) SSD performance.

***Figure 6.13:** Mean Average Precision (unit-less) performance for both YoloV3 and SSD for decreasing network sizes. The bottom half of the image shows the performance for images with artificial added noise, showing better resilience using the basal ganglia model, as well as the gain in frame rate due to the selection of the frames in which to perform object detection.*

***Table 6.1:** Kolmogorov-Smirnov (K-S) and Chi-Square (χ^2) goodness-of-fit tests for stimulus videos 1 to 10.*

Video	1	2	3	4	5	6	7	8	9	10
K-S	0.149	0.306	0.254	0.280	0.128	0.463	0.457	0.252	0.224	0.141
χ^2	0.113	0.342	0.298	0.275	0.086	0.563	0.556	0.243	0.159	0.185

that the location is fixated-on), while the human data and the MSPRT and more broadly distributed. Fig. 6.16 supplements this premise by plotting how many times each of the top 20 locations is fixated-on, for all the 19 videos. The number of appearances is shown in a logarithmic scale. One noticeable aspect of it is how, for the eye fixations, there is not a prominent location that vastly dominates the rest. The prominent fixations for the MSPRT are in a similar range than those of the eye fixations from the dataset. While in the WTA, a few pixels are always considered most salient.

A final interesting point to evaluate is the eye distance travelled for the human

***Table 6.2:** Kolmogorov-Smirnov (K-S) and Chi-Square (χ^2) goodness-of-fit tests for stimulus videos 11 to 19.*

Video	11	12	13	14	15	16	17	18	19
K-S	0.219	0.324	0.174	0.255	0.059	0.250	0.496	0.365	0.313
χ^2	0.241	0.333	0.159	0.181	0.045	0.218	0.634	0.408	0.367

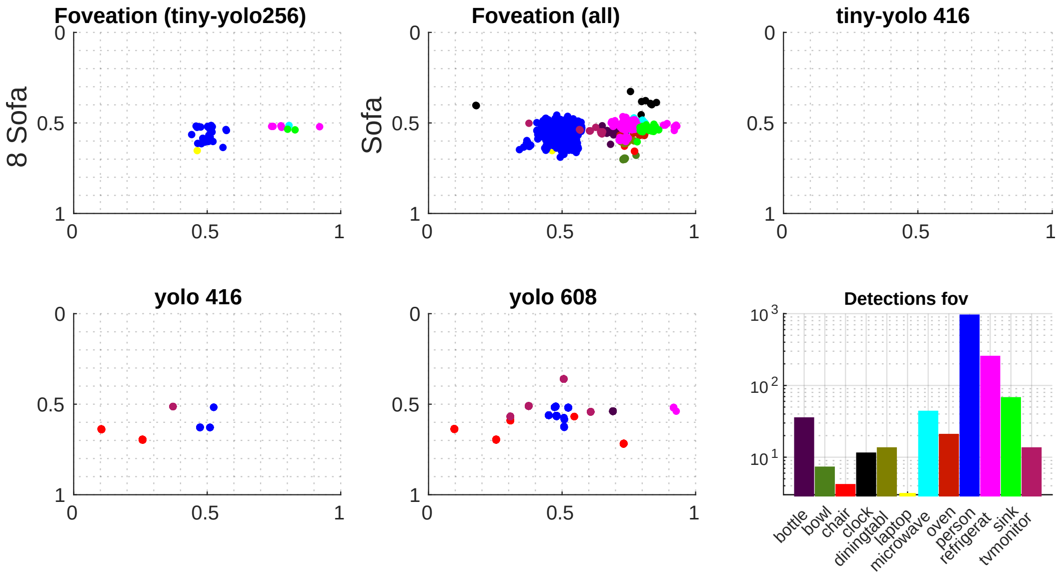


Figure 6.14: Detections by different CNN implementations in the foveated and original equirectangular frame, using 8_Sofa.mp4 stimuli video the as input. ***(From left to right)****;* ***(a)*** *Detections on foveated frames, mapped back to their equirectangular location, using the tiny-YoloV3 network at* 256 × 256 *resolution,* ***(b)*** *Repeating the detection on foveated frames at the same resolution, following the observation position of all participants in the omnidirectional dataset* ***(c)*** *No detections were possible when tested on every equirectangular frame using tiny-YoloV3 at* 416 × 416 *resolution* ***(d)*** *Detections using YoloV3 at* 416 × 416 *on omnidirectional frames* ***(e)*** *Detections using YoloV3 at* 608 × 608 *on omnidirectional frames* ***(f)*** *Graph of the main detected objects on the foveated frames.*

recordings in contrast to the current model of decision making. Higher eye distance travelled implies more sensitivity to image noise. Fig. 6.18 (a) illustrates the travelled distances between frames for 5 participants, and the MSPRT and winner-takes-all (WTA) equivalent following the same central fixation. The behaviour is similar for most observers, albeit difficult to plot for all observers in all the input videos. However Fig. 6.18 (b) gives the distance accumulation for all observers in each video. In an omnidirectional image, π is the farther distance that can be travelled in a given frame, marking a striking contrast between the required movements for a frame by frame WTA and the delay in decision making between human participants and the MSPRT, which is coherent with the times of eye/head movements of a human participant and the physical limitations for robot movements. These results confirm to a certain extent that the MSPRT algorithm reduces the sensitivity of the visual saliency system to noise compared with a WTA system.

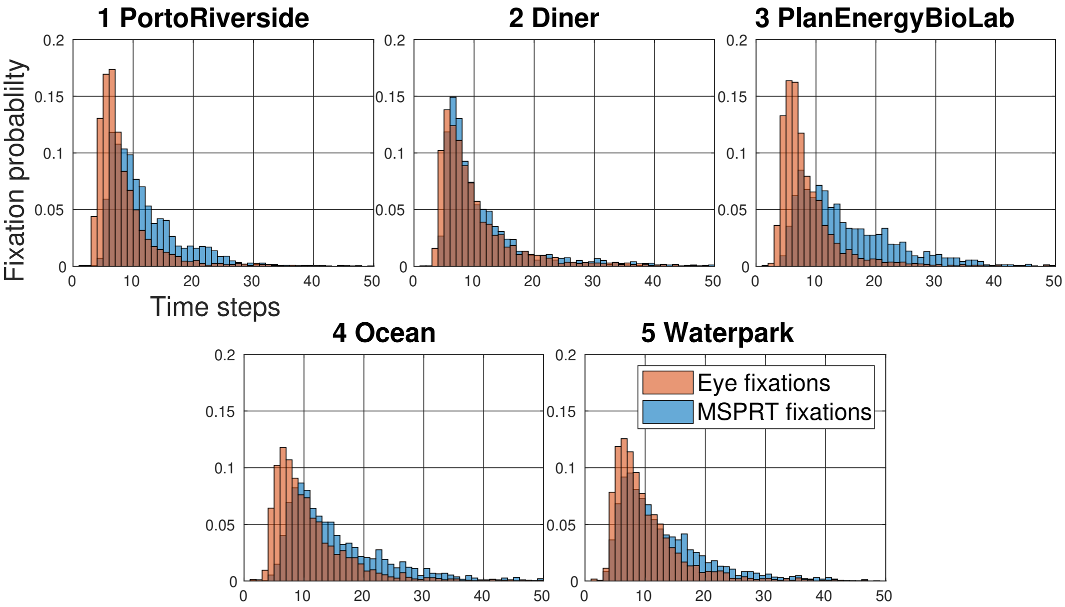


Figure 6.15: Histogram of duration of the fixations in the first five videos of the David et al. [150] dataset. For each video, the horizontal axis represents the number of frames that the fixation lasts, and the vertical axis its probability.

6.6 Discussion

This chapter has presented a novel robust visual saliency system for mobile robots, to select a region of interest in the visual scene, which fuses top-down and bottom-up saliency, and performs region selection using a bioinspired evidence accumulation algorithm related to basal ganglia decision making. The key components of the system are: top-down saliency using a foveated image transform with CNNs for fast object detection and recognition (the where and what), which is combined with biasing by task relevant information; bottom-up saliency using standard low-level image processing from intensity, colour and orientation maps; movement saliency combined with a fast path interrupt to by-pass the evidence accumulation algorithm and rapidly direct attention towards potential hazards. The results demonstrate that the visual saliency system works effectively to select regions of attention; which in turn speeds up object detection; and that the system emulates human visual saliency more closely than schemes that use a winner-take-all decision making rule; also providing more robustness to noise.

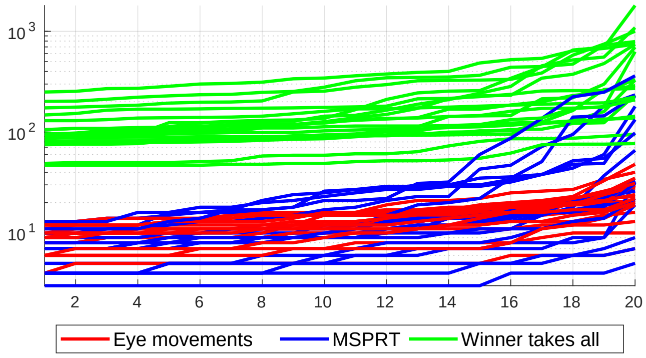


Figure 6.16: *Number of times in which each of the top 20 winning salient locations are fixated, for each method; human eye movements ground truth, MSPRT and a Winner takes all rule for every frame.*

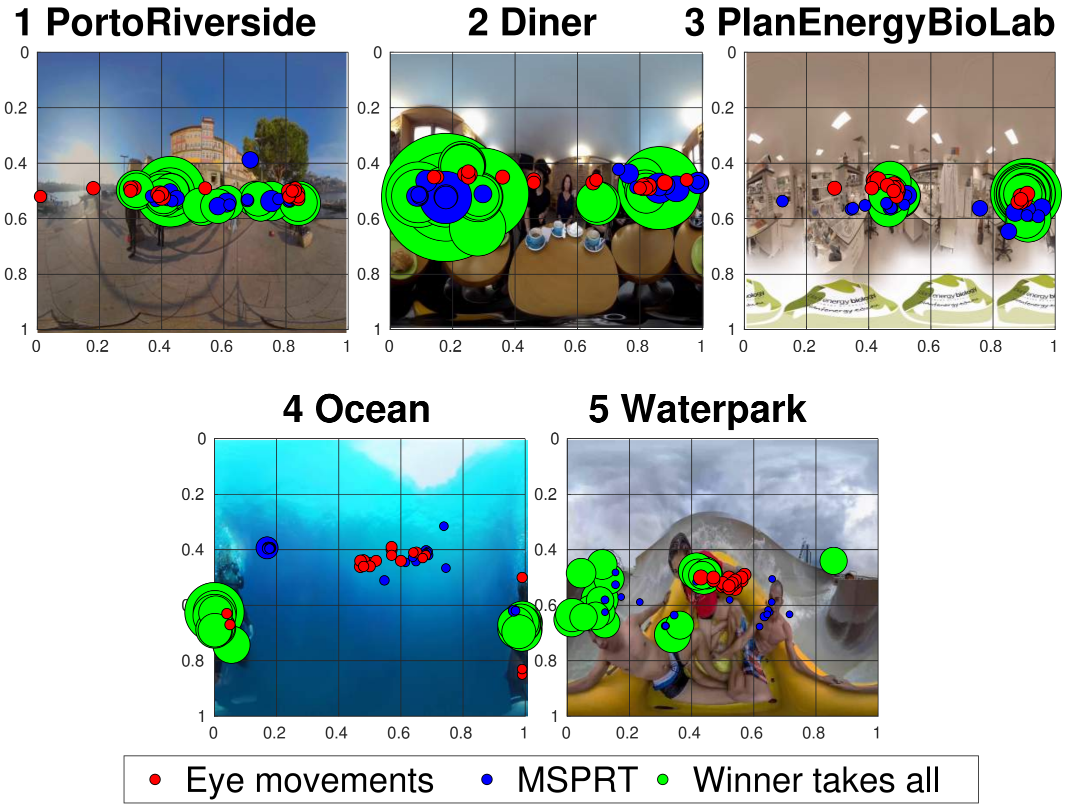


Figure 6.17: *Representation of the top 20 fixations for the first five stimulus videos. The size of the circle represents how many times each location appears, with a $[\lambda, \phi]$ precision of $[0.01, 0.01]$. The eye fixations from the dataset (illustrated in red) are not too often dominated by a single location, behaviour which is much more closely imitated by an evidence accumulation system, like the presented MSPRT, than by a WTA rule in every frame.*

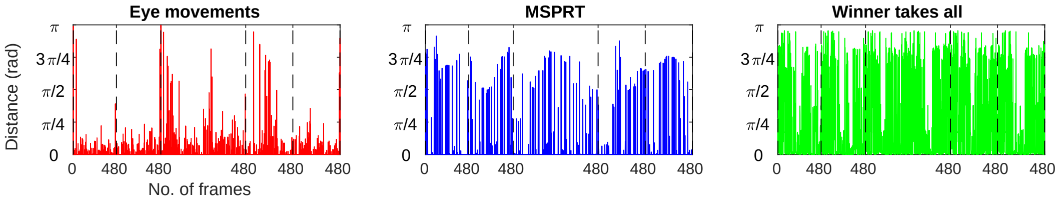


Figure 6.18: *Example of travelled distance, from frame to frame, for the first 5 participants in the 8_Sofa.mp4 stimuli video. Every dashed black line represents a new observer. This video is processed at 24 FPS (thus the total of 480 frames). The eye fixations from human dataset show a significantly smaller number of "distant" eye trips. The MSPRT lets evidence be accumulated before changing the fixation, while a WTA makes relatively distant trips in every frame.*

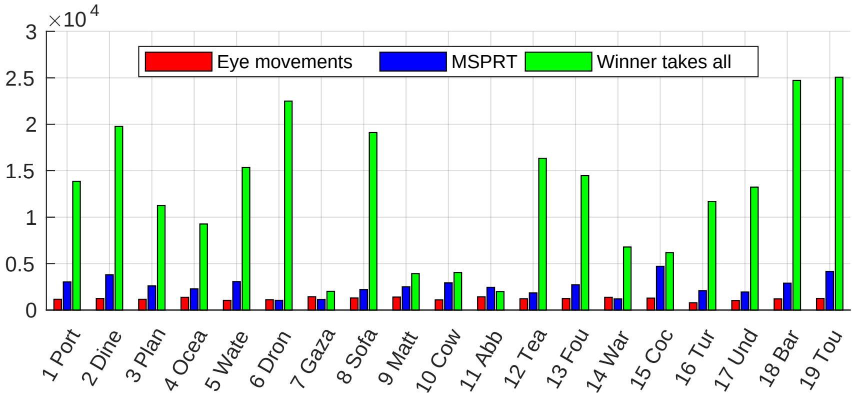


Figure 6.19: *Comparison of cumulative travelled distance between eye fixations ground truth data, MSPRT and Winner-takes-all, showing a much grater travelled distance in the case of Winner-takes-all, since the switching is done in every time-step.*

Chapter 7

Conclusions and future work

7.1 Conclusions

The work presented in this thesis aimed at testing the computational benefit and feasibility of a series of principles based on current understating of the human visual system. We took foveation as starting point, a well studied human vision system, which has been one of the most thoroughly studied in the literature of both physiological principles and computational interpretations, inspired by the rapidly advancing area of object detection and classification via deep neural networks, that have opened an increased need for information prioritization.

One of the broad goals of this project was to determine if with a simple foveation mechanism, we could maintain a baseline performance in a downscaled image, by foveating on the object of interest (therefore preserving most of its visual information) to classify and outline it, in contrast to the performance of a CNN at a larger resolution. Reaching this target, although relatively modest, allowed us to extend the hypothesis to place foveation in a wider computational context.

The next logical goal to tackle was the assumption that we can know beforehand where the object of interest will be, or a least make a statistically good estimation, so that we can foveate on it. This takes place in the form of saliency maps. We explored how, while saliency has also been extensively covered in the literature, its use in a wider context is less common. Using bottom-up saliency to detect conspicuous elements provided generally good results, subject to the complexity of the images and the number of elements, but doing considerably better in less crowded images (Section 4.10, Fig. 4.10) than those with more instances and categories present (Section 4.10, Fig. 4.9).

Saliency proved to be a good fit for images from an aerial perspective, given that they inherently have a wide view angle and that the objects of interest are relatively small (such as persons, bicycles, cars). While most of the background is not required for object detection and can be downsampled, the counterpart is that there would most likely be more than one instance of the object of interest in the scene (e.g. for a drone flying over a city). This, in turn, can be approached by using multiple foveas in each image.

Multi-foveation was explored in Chapter 5, consistently with other approaches described in the literature, it presented a set of drawbacks such as the overlap of elements between foveas, creating pixel patterns more difficult to arrange into a downsampled square image. The main strategies taken to create multiple foveas was relatively simple and should be more thoroughly studied as future work.

The last stage of this approach consisted of bringing it into the time domain, where evidence is accumulated for different salient locations until one of them is distinct enough to foveate and perform object detection, allowing to obtain a substantially better frame-rate and decrease in the use of GPU memory, making it more suitable for an autonomous robot implementation.

The decision making mechanism was based on a comprehensive model of the basal ganglia available in the literature, which proved to be adequate. The test case of omnidirectional images made some of these benefits much more evident, since the number of potential sub-regions of interest is much larger in a full field of view, and the distance between selected regions is steeper. This data also allowed us to show consistency with behaviour from human fixations, although more suitable data and more extensive tests will provide a deeper insight into the underlying principles.

7.2 Future work

7.2.1 Sensor fusion

Even though studies of multi-camera setups have existed for awhile, (e.g. sensor fusion from separate peripheral and foveal cameras [40, 41, 43]), the emerging technology of Dynamic Vision Sensors (DVS) [160] provides a new set of advantages, well suited for peripheral sensing; the lack of frame rate restricting (working with a latency in the range of μs [161]) is a huge advantage that can be exploited as a movement and fast interruption channel to complement more conventional saliency approaches. The *on / off* nature of DVS data, not containing colour information is also consistent with peripheral vision, saving computing resources than can be allocated instead for object detection in the fovea.

The effective aggregation of these two very distinct technologies needs a detailed study, which will be interesting to address as future work. A coarse representation of this approach is presented in Fig. 7.1.

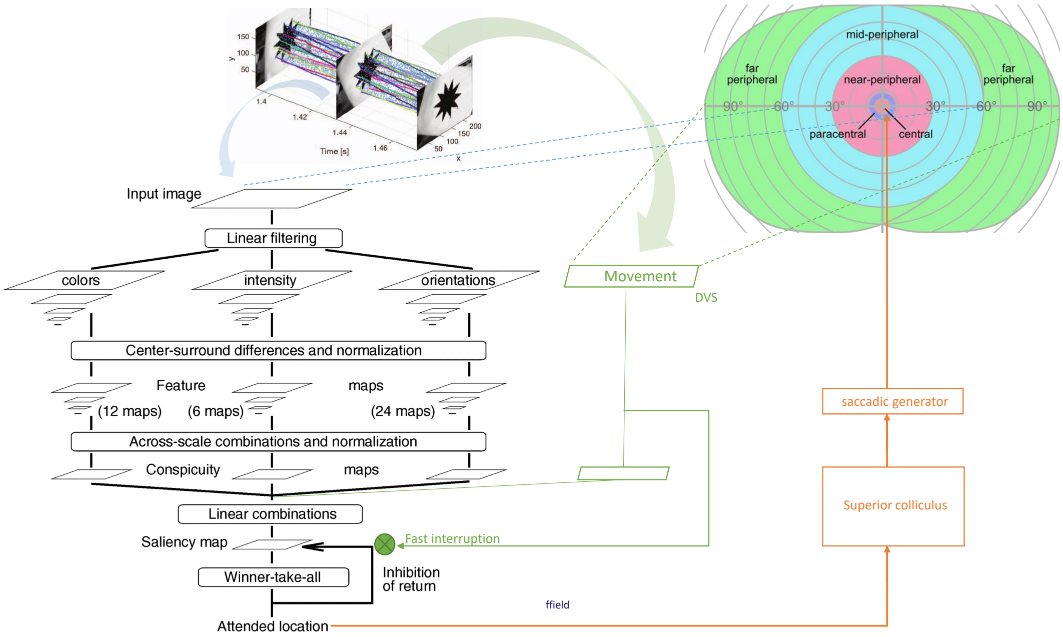


Figure 7.1: Proposed foveal and peripheral Sensor fusion with Dynamic Vision Sensor (DVS) to tackle saliency and foveation using the advantages of both technologies; speed for DVS and level of detail for conventional cameras

7.2.2 Multi-foveation

The concept of multiple foveas in a single image can provide additional advantages, although we also encountered issues consistent with the literature (described in Section 5.1). Since the approach taken was relatively simple, it would be an interesting area on which to expand. This could also encompass a hardware implementation, with several foveal cameras in a pan-tilt setup and one DVS peripheral camera as described in the previous subsection.

7.2.3 Top-down saliency

A few key aspects remain without consensus about saliency estimation; an apt sole metric for evaluation is needed and, as pointed out by [142], the temporal aspect of saliency estimation has been widely ignored by present models. It has an importance that be comes clear in mobile robotics. Bylinskii et al. [49] made a good summary of the most common shortcomings of state of the art saliency implementations, often converging in top-down features, such as text, social context cues (the inherent saliency of a leader in a group), or other elements providing additional information to the observer (e.g. cues that help identify the location of an image).

The presented approach allows to give a weighted bias towards the most context relevant top-down feedback, which an off-the-shelf CNN can be trained to detect, but it should be further studied how this bias can be addressed or measured in accordance to biomimetic evidence. There are also some limitations to current saliency estimation models, particularly background motion, with special importance in the case of mobile robots. We should also address in future what other metrics can be used to better compare to human behaviour, as well as the limitation of requiring rectangular images as input to convolutional neural network models.

In general, several important compromises where made from bio-mimetism, but allowing for more efficient information flow in a computational context. It should be studied in more detail how to approach these hardware restrictions to be able to have a more strict bio-mimetism. A likely approach would be the use of spiking neural networks, for which CNNs have started being developed in recent years, as well as their use in dedicated hardware, such as SpiNNaker and Field Programmable Gate Arrays.

Bibliography

[1] L. Itti, C. Koch, E. Niebur, et al. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[2] L. Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing*, 13(10):1304–1318, 2004.

[3] M.D. Fairchild. *Color Appearance Models*. The Wiley-IS&T Series in Imaging Science and Technology. Wiley, 2013.

[4] J. Krantz. *Experiencing Sensation and Perception*. Pearson, 2012.

[5] Z. Li. A saliency map in primary visual cortex. *Trends in cognitive sciences*, 6 (1):9–16, 2002.

[6] U. Jaramillo-Avila and S.R. Anderson. Foveated image processing for faster object detection and recognition in embedded systems using deep convolutional neural networks. In *Conference on Biomimetic and Biohybrid Systems*, pages 193–204. Springer, 2019.

[7] U. Jaramillo-Avila, J.M. Aitken, and S.R. Anderson. Visual saliency with foveated images for fast object detection and recognition in mobile robots using low-power embedded gpus. In *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 773–778. IEEE, 2019.

[8] U. Jaramillo-Avila, A. Hartwell, K. Gurney, and S.R. Anderson. Top-down bottom-up visual saliency for mobile robots using deep neural networks and task-independent feature maps. In *Towards Autonomous Robotic Systems*, volume 10965, pages 489–490. Springer Verlag, 2018.

[9] I. Repin. Unexpected visitors, 1886.

[10] A.L. Yarbus. *Eye Movements and Vision*. Springer US, 1967.

[11] K. Li, L. Guo, J. Nie, G. Li, and T. Liu. Review of methods for functional brain connectivity detection using fmri. *Computerized Medical Imaging and Graphics*, 33(2):131–139, 2009.

[12] M.P. Van Den Heuvel and H.E.H. Pol. Exploring the brain network: a review on resting-state fmri functional connectivity. *European neuropsychopharmacology*, 20(8):519–534, 2010.

[13] C. Kaufmann, R. Wehrle, T. Wetter, F. Holsboer, D. Auer, T. Pollmächer, and M. Czisch. Brain activation and hypothalamic functional connectivity during human non-rapid eye movement sleep: an eeg/fmri study. *Brain*, 129 (3):655–667, 2005.

[14] R.A. Berman, C. Colby, C. Genovese, J. Voyvodic, B. Luna, K. Thulborn, and J. Sweeney. Cortical networks subserving pursuit and saccadic eye movements in humans: an fmri study. *Human brain mapping*, 8(4):209–225, 1999.

[15] C. Rosano, C.M. Krisky, J.S. Welling, W.F. Eddy, B. Luna, K.R. Thulborn, and J.A. Sweeney. Pursuit and saccadic eye movement subregions in human frontal eye field: a high-resolution fmri investigation. *Cerebral Cortex*, 12(2): 107–115, 2002.

[16] D.H. Hubel, J. Wensveen, and B. Wick. *Eye, brain, and vision*. Scientific American Library New York, 1995.

[17] D. Purves, R. Cabeza, S.A. Huettel, K.S. LaBar, M.L. Platt, M.G. Woldorff, and E.M. Brannon. *Neuroscience*. Sunderland: Sinauer Associates, Inc, 2004.

[18] G. Indiveri, B. Linares-Barranco, T.J. Hamilton, A. Van Schaik, R. Etienne-Cummings, T. Delbruck, S.C. Liu, P. Dudek, P. Häfliger, S. Renaud, et al. Neuromorphic silicon neuron circuits. *Frontiers in neuroscience*, 5:73, 2011.

[19] T. Delbrück, B. Linares-Barranco, E. Culurciello, and C. Posch. Activity-driven, event-based vision sensors. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2426–2429. IEEE, 2010.

[20] T. Delbrück. Neuromorophic vision sensing and processing. In *46th European Solid-State Device Research Conference (ESSDERC)*, pages 7–14. IEEE, 2016.

[21] F. Rea, G. Metta, and C. Bartolozzi. Event-driven visual attention for the humanoid robot icub. *Frontiers in neuroscience*, 7, 2013.

[22] D. Sonnleithner and G. Indiveri. A neuromorphic saliency-map based active vision system. In *45th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2011.

[23] D. Sonnleithner and G. Indiveri. A real-time event-based selective attention system for active vision. In *Advances in Autonomous Mini Robots*, pages 205–219. Springer, 2012.

[24] F. Galluppi, K. Brohan, S. Davidson, T. Serrano-Gotarredona, J.A.P. Carrasco, B. Linares-Barranco, and S. Furber. A real-time, event-driven neuromorphic system for goal-directed attentional selection. In *International Conference on Neural Information Processing*, pages 226–233. Springer, 2012.

[25] S.B. Furber, F. Galluppi, S. Temple, and L.A. Plana. The spinnaker project. *Proceedings of the IEEE*, 102(5):652–665, 2014.

[26] H. Strasburger, I. Rentschler, and M. Jüttner. Peripheral vision and pattern recognition: A review. *Journal of Vision*, 11(5):1–82, 2011.

[27] H. Wässle, U. Grünert, J. Röhrenbeck, and B.B. Boycott. Cortical magnification factor and the ganglion cell density of the primate retina. *Nature*, 341 (6243):643–646, 1989.

[28] R. Rosenholtz. Capabilities and limitations of peripheral vision. *Annual Review of Vision Science*, 2(1):437–457, 2016.

[29] V.J. Traver and A. Bernardino. A review of log-polar imaging for visual perception in robotics. *Robotics and Autonomous Systems*, 58(4):378–398, 2010.

[30] F. Tong and Z.N. Li. Reciprocal-wedge transform for space-variant sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):500–511, 1995.

[31] J. Martinez and L. Altamirano. A new foveal cartesian geometry approach used for object tracking. In *Proceedings of the IASTED International Conference on Signal Processing, Pattern Recognition, and Applications, SPPRA 2006*, pages 133–139, Innsbruck, Austria, 2006.

[32] E. Akbas and M.P. Eckstein. Object detection through search with a foveated visual system. *PLoS Computational Biology*, 13(10):e1005743, 2017.

[33] J. Wei and Z.N. Li. On active camera control and camera motion recovery with foveate wavelet transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):896–903, 2001.

[34] W.S. Geisler and J.S. Perry. Variable-resolution displays for visual communication and simulation. In *SID Symposium Digest of Technical Papers*, volume 30, pages 420–423. Wiley Online Library, 1999.

[35] D.G. Bailey and C.S. Bouganis. Reconfigurable foveated active vision system. In *3rd International Conference on Sensing Technology (ICST)*, pages 162–167. IEEE, 2008.

[36] M. Bjorkman and D. Kragic. Combination of foveal and peripheral vision for object recognition and pose estimation. In *2004 IEEE International Conference on Robotics and Automation (ICRA)*, volume 5, pages 5135–5140. IEEE, 2004.

[37] A. Ude, C. Gaskett, and G. Cheng. Foveated vision systems with two cameras per eye. In *2006 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3457–3462. IEEE, 2006.

[38] A. Ude, C.G. Atkeson, and G. Cheng. Combining peripheral and foveal humanoid vision to detect, pursue, recognize and act. In *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2173–2178. IEEE, 2003.

[39] T. Shibata, S. Vijayakumar, J. Conradt, and S. Schaal. Biomimetic oculomotor control. *Adaptive Behavior*, 9(3-4):189–207, 2001.

[40] A. Ude. Active humanoid vision and object classification. In *2009 24th International Symposium on Computer and Information Sciences*, pages 387–391. IEEE, 2009.

[41] D. Kragic and M. Bjorkman. Strategies for object manipulation using foveal and peripheral vision. In *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, pages 50–50. IEEE, 2006.

[42] P.M. Sharkey, D.W. Murray, S. Vandevelde, I.D. Reid, and P.F. McLauchlan. A modular head/eye platform for real-time reactive vision. *Mechatronics*, 3 (4):517–535, 1993.

[43] C. Craye, D. Filliat, and J.F. Goudou. Biovision: a biomimetics platform for intrinsically motivated visual saliency learning. *IEEE Transactions on Cognitive and Developmental Systems*, 11(3):347–362, 2018.

[44] P. Dean, J. Porrill, S.R. Anderson, M. Dutia, J. Menzies, C. Melhuish, A.G. Pipe, A. Lenz, and T. Balakrishnan. Functions of distributed plasticity in a biologically-inspired adaptive control algorithm: From electrophysiology to robotics.

[45] A. Lenz, S.R. Anderson, A.G. Pipe, C. Melhuish, P. Dean, and J. Porrill. Cerebellar-inspired adaptive control of a robot eye actuated by pneumatic artificial muscles. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(6):1420–1433, 2009.

[46] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer, 1987.

[47] D. Walther and C. Koch. Modeling attention to salient proto-objects. *Neural networks*, 19(9):1395–1407, 2006.

[48] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):185–207, 2013.

[49] Z. Bylinskii, A. Recasens, A. Borji, A. Oliva, A. Torralba, and F. Durand. Where should saliency models look next ? *European Conference on Computer Vision*, 2016.

[50] V. Navalpakkam and L. Itti. A goal oriented attention guidance model. In *International Workshop on Biologically Motivated Computer Vision*, pages 453–461. Springer, 2002.

[51] V. Navalpakkam and L. Itti. Modeling the influence of task on attention. *Vision research*, 45(2):205–231, 2005.

[52] A. Kimura, R. Yonetani, and T. Hirayama. Computational models of human visual attention and their implementations: A survey. *IEICE TRANSACTIONS on Information and Systems*, 96(3):562–578, 2013.

[53] P.E. Forssén, D. Meger, K. Lai, S. Helmer, J.J. Little, and D.G. Lowe. Informed visual search: Combining attention and object recognition. In *2008 IEEE international conference on Robotics and automation (ICRA)*, pages 935–942. IEEE, 2008.

[54] D. Meger, P.E. Forssén, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J.J. Little, and D.G. Lowe. Curious george: An attentive semantic robot. *Robotics and Autonomous Systems*, 56(6):503–511, 2008.

[55] S. Ekvall, P. Jensfelt, and D. Kragic. Integrating active mobile robot object recognition and slam in natural environments. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5792–5797. IEEE, 2006.

[56] B. Rasolzadeh, M. Björkman, K. Hübner, and D. Kragic. An active vision system for detecting, fixating and manipulating objects in the real world. *The International Journal of Robotics Research*, 29(2-3):133–154, 2010.

[57] A.F. Almeida, R. Figueiredo, A. Bernardino, and J. Santos-Victor. Deep networks for human visual attention: A hybrid model using foveal vision. In *Robot 2017: Third Iberian Robotics Conference*, pages 117–128, 2017.

[58] A. Recasens, P. Kellnhofer, S. Stent, W. Matusik, and A. Torralba. Learning to zoom: a saliency-based sampling layer for neural networks. *arXiv preprint arXiv:1809.03355*, 2018.

[59] Z. Ghahramani. Unsupervised learning. In *Summer School on Machine Learning*, pages 72–112. Springer, 2003.

[60] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[61] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[62] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[63] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[64] M.J. Shafiee, B. Chywl, F. Li, and A. Wong. Fast YOLO: A fast "you only look once" system for real-time embedded object detection in video. *arXiv preprint arXiv:1709.05943*, 2017.

[65] N. Tijtgat, W. Van Ranst, B. Volckaert, T. Goedemé, and F. De Turck. Embedded real-time object detection for a uav warning system. In *ICCV2017, the International Conference on Computer Vision*, pages 2110–2118, 2017.

[66] B. Wu, F.N. Iandola, P.H. Jin, and K. Keutzer. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *CVPR Workshops*, pages 446–454, 2017.

[67] M. Goodale and A. Milner. Separate visual pathways for perception and action. *Trends in neurosciences*, 15(1):20, 1992.

[68] N. Kruger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A.J. Rodriguez-Sanchez, and L. Wiskott. Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1847–1871, 2012.

[69] L. Melloni, S. van Leeuwen, A. Alink, and N.G. Müller. Interaction between bottom-up saliency and top-down control: how saliency maps are created in the human brain. *Cerebral cortex*, 22(12):2943–2952, 2012.

[70] R. Veale, Z.M. Hafed, and M. Yoshida. How is visual salience computed in the brain? insights from behaviour, neurobiology and modelling. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1714):20160113, 2017.

[71] J.P. Gottlieb, M. Kusunoki, and M.E. Goldberg. The representation of visual salience in monkey parietal cortex. *Nature*, 391(6666):481–484, 1998.

[72] R. Bogacz and K. Gurney. The basal ganglia and cortex implement optimal decision making between alternative actions. *Neural computation*, 19(2):442–477, 2007.

[73] N.F. Lepora and K.N. Gurney. The basal ganglia optimize decision making over general perceptual hypotheses. *Neural Computation*, 24(11):2924–2945, 2012.

[74] N.K. Medathati, H. Neumann, G.S. Masson, and P. Kornprobst. Bio-inspired computer vision: Towards a synergistic approach of artificial and biological vision. *Computer Vision and Image Understanding*, 150:1–30, 2016.

[75] O. Hikosaka, Y. Takikawa, and R. Kawagoe. Role of the basal ganglia in the control of purposive saccadic eye movements. *Physiological reviews*, 80(3): 953–978, 2000.

[76] M.T. Wallace, L.K. Wilkinson, and B.E. Stein. Representation and integration of multiple sensory inputs in primate superior colliculus. *Journal of neurophysiology*, 76(2):1246–1266, 1996.

[77] G. Mather. *Foundations of perception*. Taylor & Francis, 2006.

[78] T.W. Robbins and B.J. Everitt. Functions of dopamine in the dorsal and ventral striatum. In *Seminars in Neuroscience*, volume 4, pages 119–127. Elsevier, 1992.

[79] A. Parent and L.N. Hazrati. Functional anatomy of the basal ganglia. i. the cortico-basal ganglia-thalamo-cortical loop. *Brain Research Reviews*, 20(1): 91–127, 1995.

[80] M. Morita and T. Hikida. Distinct roles of the direct and indirect pathways in the basal ganglia circuit mechanism. *Japanese journal of psychopharmacology*, 35(5-6):107–111, 2015.

[81] O. Hikosaka, R.H. Wurtz, et al. Visual and oculomotor functions of monkey substantia nigra pars reticulata. ii. visual responses related to fixation of gaze. *J Neurophysiol*, 49(5):1254–1267, 1983.

[82] C. Hamani, J.A. Saint-Cyr, J. Fraser, M. Kaplitt, and A.M. Lozano. The subthalamic nucleus in the context of movement disorders. *Brain*, 127(1):4–20, 2004.

[83] D. Purves, D. Fitzpatrick, L.C. Katz, A.S. Lamantia, J.O. McNamara, S.M. Williams, and G.J. Augustine. *Neuroscience*. Sinauer Associates, 2001.

[84] S.M. Sherman and R.W. Guillery. *Exploring the thalamus*. Elsevier, 2001.

[85] U.J. Ilg. The role of areas mt and mst in coding of visual motion underlying the execution of smooth pursuit. *Vision research*, 48(20):2062–2069, 2008.

[86] S. Gilaie-Dotan. Visual motion serves but is not under the purview of the dorsal pathway. *Neuropsychologia*, 89:378–392, 2016.

[87] V.S. Chakravarthy, D. Joseph, and R.S. Bapi. What do the basal ganglia do? a modeling perspective. *Biological cybernetics*, 103(3):237–253, 2010.

[88] S. Grillner, J. Hellgren, A. Menard, K. Saitoh, and M.A. Wikström. Mechanisms for selection of basic motor programs-roles for the striatum and pallidum. *Trends in neurosciences*, 28(7):364–370, 2005.

[89] T.H. Chung and J.W. Burdick. A decision-making framework for control strategies in probabilistic search. In *2007 IEEE International Conference on Robotics and Automation*, pages 4386–4393. IEEE, 2007.

[90] F.M. Montes-González, T.J. Prescott, K. Gurney, M. Humphries, and P. Redgrave. An embodied model of action selection mechanisms in the vertebrate brain. *From animals to animats*, 6:157–166, 2000.

[91] T.J. Prescott, F.M. Montes-González, K. Gurney, M.D. Humphries, and P. Redgrave. A robot model of the basal ganglia: behavior and intrinsic processing. *Neural Networks*, 19(1):31–61, 2006.

[92] B. Girard, V. Cuzin, A. Guillot, K.N. Gurney, and T.J. Prescott. Comparing a brain-inspired robot action selection mechanism with 'winner-takes-all'. In *From Animals to Animats 7: Proceedings of the seventh international conference on simulation of adaptive behavior*, volume 7, page 75. MIT Press, 2002.

[93] B. Girard, V. Cuzin, A. Guillot, K.N. Gurney, and T.J. Prescott. A basal ganglia inspired model of action selection evaluated in a robotic survival task. *Journal of integrative neuroscience*, 2(02):179–200, 2003.

[94] K. Gurney, T.J. Prescott, and P. Redgrave. A computational model of action selection in the basal ganglia. I. A new functional anatomy. *Biological Cybernetics*, 84(6):401–410, 2001.

[95] K. Gurney, T.J. Prescott, and P. Redgrave. A computational model of action selection in the basal ganglia. II. Analysis and simulation of behaviour. *Biological cybernetics*, 84(6):411–423, 2001.

[96] A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.

[97] S. Lauritzen. The sequential probability ratio test. In *BS2 Statistical Inference, Lecture 14 notes*. University of Oxford, 2004.

[98] C.W. Baum and V.V. Veeravalli. A sequential procedure for multihypothesis testing. *IEEE Transactions on Information Theory*, 40(6), 1994.

[99] N.F. Lepora. Threshold learning for optimal decision making. In *Advances in Neural Information Processing Systems*, pages 3756–3764, 2016.

[100] D. Katz, J. Kenney, and O. Brock. How can robots succeed in unstructured environments. In Science and Systems, editors, *Workshop on Robot Manipulation: Intelligence in Human Environments at Robotics*, pages 01–06, Zurich, Switzerland, June 2008.

[101] L. Hardesty. Explained: neural networks. *Retrieved from MIT News: http://news. mit. edu/2017/explained-neural-networks-deep-learning-0414*, 2017.

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

[103] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[104] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[105] J. Redmon and A. Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[106] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[107] E.L. Schwartz. Spatial mapping in the primate sensory projection: analytic structure and relevance to perception. *Biological cybernetics*, 25(4):181–194, 1977.

[108] S.W. Wilson. On the retino-cortical mapping. *International Journal of Man-Machine Studies*, 18(4):361–389, 1983.

[109] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[110] J. Redmon. Darknet: Open source neural networks in C. [online] `http://pjreddie.com/darknet/`, 2016. Accessed: 2018-08-25.

[111] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. Accessed: 2018-10-20.

[112] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4): 427–437, 2009.

[113] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[114] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[115] N.J. Butko, L. Zhang, G.W. Cottrell, and J.R. Movellan. Visual saliency model for robot cameras. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2398–2403. IEEE, 2008.

[116] S. Frintrop, T. Werner, and G. Martin Garcia. Traditional saliency reloaded: A good old model in new shape. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 82–90, 2015.

[117] A. Borji, M.M. Cheng, Q. Hou, H. Jiang, and J. Li. Salient object detection: A survey. *Computational visual media*, pages 1–34, 2019.

[118] M.M. Cheng, N.J. Mitra, X. Huang, P.H. Torr, and S.M. Hu. Global contrast based salient region detection. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):569–582, 2014.

[119] A. Borji, M.M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE transactions on image processing*, 24(12):5706–5722, 2015.

[120] A.S. Rojer and E.L. Schwartz. Design considerations for a space-variant visual sensor with complex-logarithmic geometry. In *[1990] Proceedings. 10th International Conference on Pattern Recognition*, volume 2, pages 278–285. IEEE, 1990.

[121] R. Roberts, D.N. Ta, J. Straub, K. Ok, and F. Dellaert. Saliency detection and model-based tracking: a two part vision system for small robot navigation in forested environment. In *Unmanned Systems Technology XIV*, volume 8387, 2012.

[122] J. Sokalski, T.P. Breckon, and I. Cowling. Automatic salient object detection in uav imagery. *Proc. of the 25th Int. Unmanned Air Vehicle Systems*, pages 1–12, 2010.

[123] P. Doherty and P. Rudol. A uav search and rescue scenario with human body detection and geolocalization. In *Australasian Joint Conference on Artificial Intelligence*, pages 1–13. Springer, 2007.

[124] A. Jung. imgaug. *URL: https://github. com/aleju/imgaug (visited on 26/02/2019)*, 2017.

[125] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*. Springer, 2016.

[126] S. Jung, S. Hwang, H. Shin, and D.H. Shim. Perception, guidance, and navigation for indoor autonomous drone racing using deep learning. *IEEE Robotics and Automation Letters*, 3(3):2539–2544, 2018.

[127] M.K. Al-Sharman, Y. Zweiri, M.A.K. Jaradat, R. Al-Husari, D. Gan, and L.D. Seneviratne. Deep-learning-based neural network training for state estimation enhancement: application to attitude estimation. *IEEE Transactions on Instrumentation and Measurement*, 69(1):24–34, 2019.

[128] S. Hossain and D.j. Lee. Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with gpu-based embedded devices. *Sensors*, 19(15):3371, 2019.

[129] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand. What do different evaluation metrics tell us about saliency models? *arXiv preprint arXiv:1604.03605*, 2016.

[130] M. Everingham and J. Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 2011.

[131] P. Camacho, F. Coslado, M. González, and F. Sandoval. Multifoveal imager for stereo applications. *International journal of imaging systems and technology*, 12(4):149–165, 2002.

[132] N. Dhavale and L. Itti. Saliency-based multifoveated mpeg compression. In *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.*, volume 1, pages 229–232. IEEE, 2003.

[133] W. Geisler and P. Kortum. Implementation of a foveated image coding system for bandwidth reduction of video images. In *Human Vision and Electronic Imaging, SPIE proceedings*, volume 2657, pages 350–360, 1996.

[134] F.F. Oliveira, A.A. Souza, M.A. Fernandes, R.B. Gomes, and L.M. Goncalves. Efficient 3d objects recognition using multifoveated point clouds. *Sensors*, 18 (7):2302, 2018.

[135] E. Lim, G.A. West, and S. Venkatesh. Tracking in a space variant active vision system. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 1, pages 745–749. IEEE, 1996.

[136] E. Hunsberger, V.R. Osorio, J. Orchard, and B.P. Tripp. Feature-based resource allocation for real-time stereo disparity estimation. *IEEE Access*, 5: 11645–11657, 2017.

[137] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, and A.C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

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

[139] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[140] L. Sifre and S. Mallat. Rigid-motion scattering for image classification. *Ph. D. thesis*, 2014.

[141] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[142] C. Fosco, A. Newman, P. Sukhum, Y.B. Zhang, A. Oliva, and Z. Bylinskii. How many glances? modeling multi-duration saliency. In *SVRHM Workshop at NeurIPS*, 2019.

[143] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[144] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017.

[145] X. Huang, C. Shen, X. Boix, and Q. Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 262–270, 2015.

[146] X. Zhang, T. Gao, and D. Gao. A new deep spatial transformer convolutional neural network for image saliency detection. *Design Automation for Embedded Systems*, pages 1–14, 2018.

[147] P. Redgrave, T.J. Prescott, and K. Gurney. The basal ganglia: a vertebrate solution to the selection problem? *Neuroscience*, 89(4):1009–1023, 1999.

[148] M.D. Humphries, R.D. Stewart, and K.N. Gurney. A physiologically plausible model of action selection and oscillatory activity in the basal ganglia. *Journal of Neuroscience*, 26(50):12921–12942, 2006.

[149] B.R. Beutter, M.P. Eckstein, and L.S. Stone. Saccadic and perceptual performance in visual search tasks. i. contrast detection and discrimination. *JOSA A*, 20(7):1341–1355, 2003.

[150] E.J. David, J. Gutiérrez, A. Coutrot, M.P. Da Silva, and P.L. Callet. A dataset of head and eye movements for 360 videos. In *Proceedings of the 9th ACM Multimedia Systems Conference*, pages 432–437, 2018.

[151] R. Monroy, S. Lutz, T. Chalasani, and A. Smolic. Salnet360: Saliency maps for omni-directional images with cnn. *Signal Processing: Image Communication*, 69:26–34, 2018.

[152] T. Suzuki and T. Yamanaka. Saliency map estimation for omni-directional image considering prior distributions. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2079–2084. IEEE, 2018.

[153] P. Lebreton and A. Raake. Gbvs360, bms360, prosal: Extending existing saliency prediction models from 2d to omnidirectional images. *Signal Processing: Image Communication*, 69:69–78, 2018.

[154] T. Maugey, O. Le Meur, and Z. Liu. Saliency-based navigation in omnidirectional image. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2017.

[155] Y. Rai, J. Gutiérrez, and P. Le Callet. A dataset of head and eye movements for 360 degree images. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pages 205–210, 2017.

[156] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[157] N. Mutha. Equirectangular-toolbox. `https://github.com/NitishMutha/equirectangular-toolbox`, 2017.

[158] H.M. Traquair. *An introduction to clinical perimetry*. Kimpton, London, 5th ed. edition, 1946.

[159] K. Meshgi. Histogram of color advancements. `https://github.com/meshgi/Histogram_of_Color_Advancements/tree/master/distance`, 2014.

[160] D. Tedaldi, G. Gallego, E. Mueggler, and D. Scaramuzza. Feature detection and tracking with the dynamic and active-pixel vision sensor (davis). In *2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, pages 1–7. IEEE, 2016.

[161] J.A. Leñero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco. A 3.6 μ s latency asynchronous frame-free event-driven dynamic-vision-sensor. *IEEE Journal of Solid-State Circuits*, 46(6):1443–1455, 2011.