Mariana da Silva Barros

Intelligent Hybrid Object Detection System

A B.Sc. Dissertation presented to the Centro de Informática of Universidade Federal de Pernambuco, in partial fulfillment of the requirements for the degree of Bachelor Degree in Computer Engineering.

Advisor: Paulo Salgado Gomes de Mattos Neto

Recife
2019
ACKNOWLEDGEMENTS

During my whole life and especially throughout my undergraduate degree, the experiences I have lived and the people I have met made me who I am today. Therefore, I want to thank everyone in my life for all the support and help I have received. I thank my parents, Edna and Antônio, for being the basis that made me arrive here. I thank my brother, Tiago, and my boyfriend, Marcos, for all the encouragement given when I needed it.

I would like to thank my advisor, Prof. Paulo Salgado, for guidance during the development of this work. I also thank the Informatics Center (CIn) for providing the education, opportunities, and infrastructure that allows the formation of so many people.

More important than where we arrive at the end of a path are the friendships that we make during the journey. I would like to thank my friends Lais Bandeira, Gabriela Alves, Ladson Gomes, Thiago Moura, Júlia Feitosa, Carlos Pena, Diogo Dantas, Lucas Cavalcanti, Roberto Fernandes, Ítalo Paulino, Igor Moura, Geovanny Lucas and so many others for being part of my life.

I would also like to thank especially every member from the RobôCIn and the E.S.T.U.F.A., for sharing their routines, hopes, and dreams with me, and for helping me every day to become a better person.
Start by doing what’s necessary;
then do what’s possible;
and suddenly you are doing the impossible.

São Francisco de Assis
RESUMO

Problema cada vez mais estudado e pesquisado, a detecção de objetos, bastante presente na sociedade atual, pode ser aplicada em diversas situações diferentes. Atualmente, a aprendizagem de máquinas tem sido bastante usada para resolver este tipo de problemas, ao utilizar técnicas envolvendo redes neurais convolucionais, por exemplo. Porém, apesar da grande quantidade de aplicações já desenvolvidas na área, ainda há espaço para aperfeiçoamentos em situações específicas. Um exemplo é o uso deste tipo de algoritmo em imagens panorâmicas, que normalmente apresentam distorções geométricas que dificultam a detecção de objetos. Além disso, um dos maiores problemas ao se usar este tipo de técnica envolve determinar os melhores valores para os parâmetros usados na aprendizagem. Diante da relevância deste problema, o presente estudo desenvolveu um sistema híbrido para detecção de objetos em imagens panorâmicas utilizando técnicas de aprendizagem de máquinas e estratégia evolutiva. Com o objetivo de avaliar a evolução do método, foi escolhida a medida de precisão média do algoritmo de detecção de objetos escolhido, ”Faster R-CNN”. Os resultados mostram que a solução proposta pode ser melhorada para a obtenção de melhores resultados.

Increasingly studied and researched, object detection is a problem present in society that may be applied in different situations. Nowadays, machine learning has been considerably adopted to solve this kind of problems when using techniques involving convolutional neural networks, for example. However, despite the vast number of applications already developed in the area, there is still room for improvement in specific situations. One example is the use of panoramic images, which usually present geometric distortions that interfere in the object detection task. Besides that, one of the most significant issues when using this type of technique involves determining the best values for the parameters used in the algorithm training. Given the relevance of this problem, the present study has developed a hybrid system for object detection over panoramic images using machine learning and evolution strategy techniques. In order to evaluate the method’s evolution, it has been chosen the measurement "average precision", from the object detection algorithm, Faster R-CNN. The obtained results show that the proposed solution may be improved in order to obtain better results.

Keywords: Machine Learning, Evolution Strategy, Object Detection, Panoramic Images.
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Object recognition related tasks [16]</td>
<td>14</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Relationship between Deep Learning and Artificial Intelligence [1]</td>
<td>15</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Neural Network system [1]</td>
<td>16</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Types of panoramic image projections [9]</td>
<td>19</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Workflow from the work proposed by Yang <em>et al.</em> [10]</td>
<td>21</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Workflow from the work proposed by Yu <em>et al.</em> [3]</td>
<td>22</td>
</tr>
<tr>
<td>Figure 7</td>
<td>System workflow</td>
<td>24</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Evolution Strategy flow</td>
<td>25</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Faster R-CNN Architecture [8]</td>
<td>28</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Positive detection example on Experiment 1</td>
<td>33</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Negative detection example on Experiment 1</td>
<td>33</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Results of the Training Loss for Experiment 2</td>
<td>35</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Loss per Iteration at Generation 6</td>
<td>36</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Loss per Iteration at Generation 12</td>
<td>37</td>
</tr>
</tbody>
</table>
# CONTENTS

1  INTRODUCTION ........................................................................................................... 9  
  1.1  Motivation ........................................................................................................... 9  
  1.2  Objectives .......................................................................................................... 11  
  1.3  Document Structure .......................................................................................... 11  

2  BASIC CONCEPTS ................................................................................................. 12  
  2.1  Evolution Strategies ........................................................................................... 12  
  2.2  Computer Vision and Object Detection .............................................................. 13  
  2.3  Convolutional Neural Networks (CNNs) ............................................................. 15  
    2.3.1  Background on DNNs .................................................................................. 15  
    2.3.2  Overview of CNNs ...................................................................................... 17  
  2.4  Panoramic Image Projections .............................................................................. 18  

3  RELATED WORKS .................................................................................................... 20  
  3.1  Object Detection in Equirectangular Panorama .................................................... 20  
  3.2  Object Detection on Panoramic Images Based on Deep Learning ....................... 21  
  3.3  Grid Based Spherical CNN for Object Detection from Panoramic Images ............ 22  
  3.4  A Domain-Independent Window Approach to Multiclass Object Detection Using Genetic Programing ................................................................. 23  

4  METHODOLOGY ...................................................................................................... 24  
  4.1  Overview ............................................................................................................ 24  
  4.2  Step 1 - Evolution Strategy ................................................................................. 25  
  4.3  Step 2 - Faster R-CNN ....................................................................................... 27  
    4.3.1  First Stage .................................................................................................... 27  
    4.3.2  Second Stage ............................................................................................... 29  

5  EXPERIMENTS AND ANALYSIS ........................................................................... 30  
  5.1  Experimental Setup ............................................................................................ 30  
  5.2  Performance Measurements ............................................................................... 31  
    5.2.1  Training Loss ............................................................................................... 31
1 INTRODUCTION

1.1 Motivation

Sze et al. [1] describes Machine Learning as the field of study that gives computers the ability to learn without being explicitly programmed. In other words, it relies on the idea that computer systems can learn from data with minimal human intervention. Due to its extensive applicability, it is present in several different aspects and situations from people’s lives. Within the Machine Learning domain, object detection is a fundamental visual recognition problem in computer vision and has been widely studied in the past decades, according to Wu et al. [2]. Yu et al. [3] states that a vision-based object detection task is to recognize and locate objects of interest in a given image efficiently and accurately. It forms the basis of many other computer vision tasks, such as segmentation and object tracking, as Zou et al. [4] states.

According to Zou et al. [4], the popularization of object detection and recognition algorithms trace back to 18 years ago, with the Viola-Jones detector, achieving real-time detection of human faces for the first time without any constraints (skin color segmentation, for example). This detector used sliding windows and incorporated some crucial techniques, such as “feature selection” and “detection cascades”. In 2005, it was proposed the Histogram of Oriented Gradients (HOG) feature descriptor, according to Dalal et al. [5]. Motivated by the pedestrian detection problem, this detector can be considered as a vital improvement to the domain, for using techniques that increase the accuracy and being an essential foundation of many object detectors. In 2014, with the convolutional neural network developments, Girshick et al. [6] proposed the Region-based Convolutional Neural Networks (R-CNN) for object detection. This algorithm has been improved and optimized with the creation of Fast R-CNN [7] and Faster R-CNN [8], which are part of the state-of-art in object detection.

Given the great importance of Machine Learning and object detection, they are present in the society in many different ways, at various applications, as, for example, character recognition, object tracking, face detection and recognition, medical applications, industry applications, robotics and object counting. These applications are just some of the many different areas where it may be applied.

One specific problem related to object detection is the application of these algo-
rithms over panoramic images. This type of image, also called equirectangular projections, represents a view of 180 degrees vertically and 360 degrees horizontally, according to [9]. They are also present in some different contexts, like virtual reality, autonomous vehicles, and surveillance, for example. Therefore, it may be relevant to apply object detection on images with these properties. However, due to the characteristics of this kind of image, it presents geometric distortions, especially in the corners, which creates a problem for the object detectors. This distortion, added to the high pixel resolution of panoramic images, contribute to the low accuracy of object detection algorithms over them.

There have been developed some works in the state-of-the-art that try to solve the problem with the panoramic images. Yang et al. [10] and Deng et al. [11] work with the known object detection algorithms, as Faster R-CNN and YOLO, over panoramic images, achieving a maximum detection precision of 68.7%. Yu et al. [3] proposes a grid spherical convolutional neural network, that includes the image processing as a spherical image beside the use of the object detection algorithm, reaching a mean Average Precision of 60%. Zhang et al. [12] proposes an algorithm that uses a genetic programming technique to detect small objects from multiple classes in large images.

One possible solution for improving the object detection task over panoramic images is to perform the image pre-processing before the execution of the algorithm. However, it usually requires deepness information from the image, as the point cloud, for example (which contains the 3-dimensional information about the panoramic image). Besides the fact that this data is sometimes not available, this process is costly and takes a long time.

In a general way, object detection algorithms use several configuration parameters during training and inference that are possible to change in order to better adequate to a specific set of images. Therefore, it is possible to find the parameters that give the best performance results when applied to this image set.

One possible way of discovering the desired values to these parameters is to use evolution strategies. This strategy uses evolution-based concepts to find the values that maximize the results to a specific function, as Moreira et al. [13] states. In this case, evolution strategies may be used to find parameter values that are best adapted to perform object detection over panoramic images.

This way, the present work proposes a hybrid system that uses Machine Learning
techniques to perform object detection over panoramic images, using evolution strategies to find the best values to the training parameters from the neural network algorithm.

1.2 Objectives

- To develop an intelligent hybrid system to detect objects at panoramic images using machine learning techniques.
- To optimize and improve object detection at panoramic images.
- To use an evolution strategy to determine the optimal training parameters from the object detection algorithm at panoramic images.

1.3 Document Structure

This monograph is divided as follows. Chapter 2 describes some essential concepts related to the system implemented in this work, as evolution strategies, machine learning, and object detection, convolutional neural networks, and panoramic image projections. Chapter 3 mentions and describes some works related to the proposed system, what problem they intend to solve, and how they accomplish it. Chapter 4 characterizes the system architecture and the methodology that we followed in this work. Chapter 5 specifies the experimental setup and the performance measurements, details the performed experiments and its results, and analyses them. Chapter 6 concludes the work and presents future activities.
2 BASIC CONCEPTS

The following section describes some basic concepts relevant to the development of this work.

2.1 Evolution Strategies

Evolutionary Computing is composed of a set of algorithms based on the evolution of the species, a theory presented by Charles Darwin. Inspired by the theory of evolution and natural selection, evolution strategy is an instance of the Evolutionary Algorithms. It is possible to apply it in several fields of optimization, including continuous, discrete, and combinatorial search spaces, as stated by Beyer [14].

According to Moreira et al. [13], evolution strategies were developed in order to solve optimization problems based on natural mechanisms of evolution of the species that occur in nature. In a general way, this kind of strategy works intending to find the optimal solution to a given problem. With this purpose, the candidate solutions consist of the values that we want to discover. The approach uses some concepts related to the evolution of the species to explore the space and find the best candidate solution. This section will describe and explain some key concepts and the general idea of an evolution strategy.

1. Individual: It is the primary element of the evolution strategy. Also called a candidate solution, it will change during the execution of the algorithm in order to find the best possible solution.

2. Population: It is a set of individuals. Despite the techniques to apply variance at the individuals, the population should remain constant at each generation.

3. Genotype: It is the content of each individual. It contains the actual information that each individual holds.

4. Fitness function: It is the objective to be achieved during the algorithm. It can be maximized or minimized.

5. Fitness value: It is the value obtained when the algorithm applies the fitness function to the genotype of each individual. It is used to sort the individuals of a population and choose which ones will remain and which ones will be discarded.
6. Generation: It is each iteration of the algorithm. At each generation, the method performs the reproduction, which can be through crossover, mutation, or both of them. At the ended of the generation, it sorts the population and checks the stopping conditions. If they are not satisfied, the next generation will run until it reaches the maximum number of iterations or an early stop condition.

7. Mutation: It is the process of changing something of the individual genotype. In a probabilistic way, one or more of the individual’s elements will change, creating a new genotype. That allows for exploring the space solution.

8. Stop conditions: The are three possible ways to stop the execution of the algorithm. The first one is the maximum number of iterations, defined at the beginning of the process. However, an early stop may occur. If it achieves the maximum run time or the target fitness value, the algorithm reaches its end.

The iteration loop is the same as the genetic algorithm, mainly composed of four steps: initialization, selection, genetic operators, and termination, which corresponds to the steps in natural selection. However, according to Beyer et al. [15], evolution strategies are capable of self-adaptation, meaning that each individual has its strategy parameters, which are subject to variation. After mutation, these parameters are used to control the statistical properties of the operators applied to the individual’s object parameters. It means that, during the evolution strategy, the individual will learn the optimal strategy parameters during the evolution process.

2.2 Computer Vision and Object Detection

In the present day, the use of Computer Vision is getting more and more common in people’s lives. The applications from this domain are numerous, but one crucial example is object recognition, which describes a set of Computer Vision related tasks that involve identifying objects in images. According to Agarwal [16], among them, it is possible to distinguish three main tasks: image classification, object detection, and object segmentation.

1. Image Classification: Given an image with a single object, it aims to recognize semantic categories of objects in it.
2. Object Detection: A general case of the problem involving classification and localization, when the number of objects is not known. It locates the presence of objects in the image with a bounding-box (a box around the detected object) and classifies each one of them.

3. Object Segmentation: Includes identifying parts of the image and understanding what object they belong to. Each pixel from the recognized objects present in the image is identified and assigned a specific category label.

Figure 1 shows some tasks related to object recognition.

![Figure 1: Object recognition related tasks](image)

In order to solve this kind of problem, there has been developed different object detection algorithms. According to Wu et al. [2], before the use of deep learning, the pipeline of object detection was divided into three steps: proposal generation, feature vector extraction, and region classification. The objective of the first step is to search locations in the image, which may contain objects, called regions of interest. During the second step, a feature vector is obtained for each region present in the image, which is usually encoded by low-level visual descriptors. Then, in the third step, the region classifiers learn to assign categorical labels to the regions.

However, some of the limitations of these detectors include a significant number of redundant proposals generated (resulting in a large number of false positives during classification) and difficulty in capture semantic information in complex contexts, making it impossible to reach the optimal global solution. As a workaround to these limitations, deep learning algorithms were inserted in the computer vision-related tasks, improving their performance and precision. One example is the Convolutional Neural Networks (CNNs) that is better explained in the following section.
2.3 Convolutional Neural Networks (CNNs)

2.3.1 Background on DNNs

Figure 2 shows that, according to Sze et al. [1], the Deep Neural Networks (DNNs) are included in the field of Artificial Intelligence.

![Figure 2: Relationship between Deep Learning and Artificial Intelligence [1]](image)

According to Sze et al. [1], Artificial Intelligence (AI) is the science and engineering of creating intelligent machines that can achieve goals as humans do. Inside the AI domain, Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed. A sub-field of it is the brain-inspired computation, where the way the brain works inspire some aspects of its basic form and functionality. Inside that, there is the area of spiking computing, and, on the other side, the neural networks field.

Based on the biological nervous system operation, neural networks are composed of several neurons organized in different layers. Each neuron’s computation involves a weighted sum of the input values. Figure 3 describes a neural network system.

In a general way, the neurons in the input layer receive the input values and propagate them to the middle layer or the “hidden layer” of the network. Each value is associated with a “weight”. The network propagates these weighted values through the hidden layers until the output layer, and the output layer will present the network outputs to the user.

Several studies define deep learning as the area inside the neural networks where
there are more than three layers (more than one hidden layer). The number of layers in a DNN is usually between 5 and 1000. The learning process in a DNN, also called training, involves determining the value of the weights in the network, while the inference involves running the network with the found weights.

The inference receives an image as input and outputs a vector of scores, one for each class. The vector contents indicate the probability of the object belonging to that class. Usually, the object with the highest score represents the most likely class to it. We can describe the loss as the gap between the correct ideal scores and the scores computed by the DNN based on its current weights. Therefore, the DNN training goal is to determine the weights that maximize the correct class score, or that minimize the average loss over an extensive training set.

The optimization process used in network training is called “gradient descent”. In this process, the partial derivative of the loss related to each weight is used to update it. That is repeated at each iteration in order to reduce the overall loss. The algorithm uses backpropagation to compute the partial derivatives of the gradients, which operates by passing values backward through the network to compute how the loss is affected by each weight. In order to improve the training performance and efficiency, it is used a series of techniques, as, for example, batch, supervised learning, reinforcement learning, and fine-tuning.

Since the first developments in DNNs, specialists in the area have been developing
various models, each one with a different network architecture (different layer shapes and types or number of layers). One of the most famous is VGG-16, composed of 16 layers, from which 13 are convolutional layers and 3 are fully-connected layers (meaning that all outputs connect to all the inputs from the next layer). Some accessible datasets usually used in object detection tasks with deep learning are PASCAL VOC, Microsoft COCO, and ImageNet.

2.3.2 Overview of CNNs

With the development and improvement of deep learning algorithms, they started to be applied in image classification and object detection. One example is the Convolutional Neural Network (CNN), a deep learning algorithm that takes an image as input, assigns learnable weights to various objects in it, and can differentiate one from another. According to Pathak et al. [17], object detection methods have been extensively using deep learning techniques based on Convolutional Neural Networks (CNNs). They are composed of multiple convolutional layers. At each one, the network generates a feature map, which consists of a higher-level abstraction of the input data that preserves essential yet unique information.

This kind of technique uses supervised learning, that receives, during the training phase, a set of input examples labeled with the desired output value. As it trains, the algorithm calculates the error between the value predicted by the system and the real output value. Deep learning techniques, besides, are composed of several levels of representation, and each level uses the information generated by the previous one.

One of the most known groups of these algorithms is the R-CNN model family, whose name means “Region-based Convolutional Neural Networks”. The first algorithm in this family is R-CNN [6], which, according to Ouaknine [18], combines the selective search method to detect region proposals and deep learning to find out the object in these regions. An optimization of this method is the Fast R-CNN model [7], which takes the entire image as input and detects in there Regions of Interests (ROIs). This step of the algorithm is called Region Proposal Network (RPN). These ROIs feed neural networks that predict and classify the observed object, drawing a bounding-box around it. One example of optimization of these algorithms, Faster R-CNN [8], was chosen in this work and is better explained in the next sections.
Nevertheless, other algorithms also work well in these situations, as, for example, YOLO (You Only Look Once) [19], which is an object detection system targeted for real-time processing. It uses a single neural network that is applied to the full image and divides it into regions, locating and classifying objects in each one. Also, there is still the SSD algorithm [20], which optimizes the Faster R-CNN algorithm using lower resolution images in order to achieve real-time object detection.

2.4 Panoramic Image Projections

In order to display 3-dimensional data in a 2-dimensional image, it is necessary to create a projection from this image. The different image projections apply different ways of representing in an image the information present in the surface of a sphere, for example. According to what is described in [9], among the various types of projection, there are the following ones:

1. Rectilinear projection: A portion of a sphere is projected on a flat surface.

2. Cylindrical Projection: The sphere is seen inside a cylinder. It projects the portion of the sphere (the cylinder) on a flat surface (the image).

3. Spherical or Equirectangular Projection: It is used to project the whole sphere on a flat surface. In the resulting image, the width is exactly twice the height. It covers 360 degrees horizontally and 180 degrees vertically.

4. Cubic Projection: Used as a way to reduce distortion from other projections, as the equirectangular panorama. The sphere is inside a cube, and this cube is unfolded, generating its six faces.

Figure 4 shows the different types of panoramic image projections. Figure 4.a shows a rectilinear projection. Figure 4.b shows a cylindrical projection. Figure 4.c shows an equirectangular projection, and the Figure 4.d shows a cubic projection.

In this work, it was used images in the equirectangular projection, which correspond to most of the 360-degree panoramic images.
Figure 4: Types of panoramic image projections [9]
3 RELATED WORKS

The following section describes some articles and projects related to the application of object detection algorithms in panoramic images.

3.1 Object Detection in Equirectangular Panorama

Yang et al. [10] performs a study on object detection on panoramic images based on deep learning algorithms. In order to evaluate the chosen detectors on 360-degree images, the author created an equirectangular dataset. It was selected 22 4k resolution VR (virtual reality) videos, from which he took 903 frames and 7199 annotated objects, that are also available in the Microsoft COCO dataset. In comparison with the traditional COCO dataset, the created dataset includes more small objects, and the images present geometric distortion present in equirectangular panoramas. In the experiments, the author selected 6431 annotated objects also available in the COCO dataset. The objects belonged to the classes person, car, and boat, as the number of objects belonging to the other classes was not expressive.

Another experiment performed in the article is the comparison between Faster R-CNN and YOLO, two state-of-the-art object detectors based on deep learning. In the first step of the experiment, the authors compared the original detectors in the object detection on rectangular panoramas. In the second one, the same detectors were re-trained with examples in the COCO dataset.

Lastly, as shown in Figure 5, he proposed a variant of YOLO based on multi-projection, where the results obtained were better than the traditional YOLO. The original version of the algorithm is pre-trained with ImageNet and fine-tuned with COCO. However, the geometric distortion from the 360-degree images is a significant problem in object detection. In order to remove these distortions, it is possible to project sub-windows from the 360-degree panorama onto a 2-D plane. However, the smaller the sub-windows are, the more significant processing is needed. The article implements a multi-projection approach with a wide field-of-view and soft selection adopted to select detections produced by multiple windows.
3.2 Object Detection on Panoramic Images Based on Deep Learning

Deng et al. [11] states the importance of collecting panoramic images from indoor environments and automatically recognizing and detecting objects in them. It also describes the distortions as the main challenge to object detection. In order to train and test deep learning algorithms into equirectangular panoramic images, the author proposed a fast method for panorama construction and created a panoramic image dataset for the indoor environment. With the use of fisheye cameras, they have designed a 360-degree panoramic camera system, and the algorithm created performed image processing for distortion correction.

In a second step, a multi-class object detection was performed on the created images manually labeled according to the PASCAL VOC format, focusing on indoor contexts. The algorithm chosen to execute the object classification and detection was Faster R-CNN, for its capability of detecting objects due to a wide range of scales and aspect ratios. The implementation model of the algorithm selected was the alternating optimization, which we can divide into four steps of training. First, the region proposal network is trained using a pre-trained ImageNet model. Then, the generated proposals are used to train the detection network using the R-CNN network, also initialized with an ImageNet pre-trained model. In the third step, the parameters learned in the previous one are used to initialize the Region Proposal Network, keeping the shared convolutional network fixed. Finally, in the last step, the classification and detection layers are added to the network, keeping the shared convolutional network still fixed. The detection precision reached is over 68.7%. 

![Figure 5: Workflow from the work proposed by Yang et al. [10]](image)
3.3 Grid Based Spherical CNN for Object Detection from Panoramic Images

Yu et al. [3] presents a grid-based Spherical Convolutional Neural Network (SCNN) for detecting objects in panoramic images. It starts from the Spherical CNN technique [21], which shows some advantages related to the object detection in spherical images when compared to the planar CNNs. However, the SCNN presents some restrictions. In the first place, the accurate location information is lost, meaning it is difficult to retrieve the bounding-box from object detection. Furthermore, in the second place, the pixel resolution in a planar image needs to be wide enough to preserve the small object’s signal. In order to overcome these limitations, the article proposes some changes that extend the applications of SCNNs to object detection.

This technique’s idea, shown in Figure 6, is to unwrap a sphere to a conformal grid map, which is the input of the network. Then, the feature maps of an object are scaled to cover a particular area of the grid map, instead of using their original size. This strategy solves the resolution problem from the traditional SCNN. The method partially solves the problem of location information loss by using a rotation augmented planar region proposal network (RPN).

![Figure 6: Workflow from the work proposed by Yu et al. [3]](image)

The main contribution from the article is the extension of the S-CNN’s capacity to object detection. The authors compare the results from this method with five other methods: Faster R-CNN, Faster R-CNN with Feature Pyramid Network (FPN), SCNN, SSD, and another variant from Faster R-CNN. Besides, they also create an omnidirectional image dataset of real street scenes with multi-class annotations, which they use in the performed experiments in the article.
3.4 A Domain-Independent Window Approach to Multiclass Object Detection Using Genetic Programming

Zhang et al. [12] has the objective of detecting small objects from multiple classes in large images. In order to do so, the algorithm locates and classifies the objects of interest by scanning the images. Its goal is to investigate an adaptive, single-stage, and domain-independent approach to be applied in multiple-class object detection problems, without pre-processing, segmentation, or feature extraction. Instead, it uses a genetic programming technique that the author extends to this kind of problem.

The single-stage approach means that it uses a single program in the whole object detection procedure. It is segmentation-free, and it does not use feature extraction, for example. Alternatively, the input to the evolved programs is pixel statistics. The fact that it is domain-independent means that the same method will work unchanged on a range of problems. It is also related to the fact that no features are used, just the raw pixels directly as inputs to the detector or classifier. However, this approach takes long learning times, as a large number of pixels is necessary. It also requires a large number of training examples.

The method starts from an image database that the author manually annotates and divides into training and test set. After that, the algorithm determines the appropriate square size necessary to cover all single objects of interest from the input set. Then, it applies the evolutionary process with images on the training set in order to generate a program to determine the class of an object. In sequence, the method applies the generated program as a moving window template on the images at the test set. At the same time, it obtains the locations of all objects of interest in each class. Finally, it calculates the performance measures.

The first used performance measure is the detection rate, which is the number of small objects correctly reported by a detection system. The second one, the false alarm rate, describes the number of non-objects incorrectly reported as objects by a detection system. Both measurements are computed as a percentage of the total number of actual objects in the image. The detection rate is expected to be the highest possible, and the false alarm rate, the smallest possible. The genetic program algorithm searches this trade-off.
4 METHODOLOGY

4.1 Overview

The presented project proposes a hybrid system to perform object detection on panoramic images. There are currently various methods and algorithms to recognize and detect objects. However, they are trained and implemented on planar images, and the geometric distortions present on spherical images often make it difficult to detect objects on this kind of images.

Given this scenario, the proposed system consists of a combination of an evolution strategy and a convolutional neural network, developed in order to find the best training parameters from CNN through the evolution strategy. Figure 7 describes the overview of the system workflow.

![Figure 7: System workflow](image)

The blue steps are related to the Evolution Strategy, and the red ones are related to the object detection algorithm (in this case, Faster R-CNN). The technique uses the training and test steps from the algorithm in order to calculate the fitness value that the
Evolution Strategy evaluates (in this case, Average Precision). The following sections explain the two parts of the system.

4.2 Step 1 - Evolution Strategy

As explained in section 2, the Evolution Strategy is an algorithm that uses some principles related to the evolution theory and searches the values that give the best result when applied in the fitness function. It follows a basic flow that the Figure 8 describes.

![Figure 8: Evolution Strategy flow](image)

As shown in Figure 8, the basic flow of an evolution strategy follows the following steps:

1. Initialization: The population is initialized, usually in a random way.

2. Evaluation: The population is sorted using the fitness value.

3. Recombination: The algorithm combines the individuals in the population (here called parents) among themselves (which can occur in many different ways), generating an offspring.

4. Mutation: The offspring individuals suffer small changes probabilistically.

5. Evaluation: The population is sorted using the fitness value.
6. Selection: In order to keep the population number constant after the reproduction, the whole population (parents and offspring) are sorted using the fitness value and the best individuals are kept in the population, while the other ones do not remain.

7. Termination: If it reaches an early stop condition, or if the maximum number of generations has arrived, the algorithm ends.

In the present situation, the main objective is to find the best training parameters for the CNN algorithm when applied to panoramic images. The implementation of the evolution strategy is based on the following configuration:

- Individual’s genotype: The contents of the genotype correspond to four training hyperparameters from the Faster R-CNN algorithm. These parameters that are better explained in the following section are:
  - RPN positive overlap
  - RPN batch size
  - Batch size
  - Bounding-box threshold

- Initialization: The initial Individual from the population is determined in a way that its genotype corresponds to the default parameters from the Faster R-CNN algorithm implementation. This genotype, called "warm start", is combined to a normal distribution to start the initial population.

- Fitness function: The measurement that is used as fitness value is the mean Average Precision (or ”mAP”), the result used to evaluate the object detection algorithm. It measures the precision from the inference performed after the model training from the Faster R-CNN algorithm.

- Strategy: The strategy used in the generation of the offspring was Single Variance. This strategy uses the same factor applied to the whole parent genotype in order to generate the child. Each element from this genotype is multiplied by the same value in order to perform the mutation.
• Fitness value evaluation: In this case, we are trying to maximize the fitness value (we want the highest possible average precision). Therefore, the population is sorted in descending order.

• Selection: After the population sort using the fitness value, the best individuals substitute the worst in the population.

In summary, the fitness function is the average precision obtained in the Faster R-CNN inference. This inference results from the application of the model trained using the chosen hyperparameters. This way, every time the fitness function is evaluated, the Faster R-CNN algorithm trains a model, the inference uses in order to detect objects at panoramic images. The target fitness value that acts as one of the stopping conditions is the maximum possible value that the average inference precision can reach (100%).

4.3 Step 2 - Faster R-CNN

Figure 9 describes the general architecture from Faster R-CNN [8], a deep learning algorithm widely used for object detection. The algorithm consists of two main parts: the first stage, which tries to know if a region of interest represents either an object or the background, and the second stage, where it is performed the object detection and recognition, as described by Mohan [22].

Before being generated the region proposals, the input image passes through a Convolutional Neural Network, called backbone, that acts as a filter and extracts the feature map. The objective of this step is to reduce the image dimensions, maintaining its features. The Faster R-CNN training goal is to share the convolutional layers from both parts of the algorithm, so it uses the same feature map as the input of the first and the second stage.

4.3.1 First Stage

The first stage from Faster R-CNN, also known as the Region Proposal Network (RPN), receives the created map as input and generates the region proposals (the regions in the image more likely to contain an object). In order to do so, a small network slides over the input feature map. At each position, it applies transformations and filters to this window, resulting in a vector that is sent to two layers. The first one, the box-classification
layer ("cls"), outputs two scores for each prediction: the probability of being an object and the probability of not being an object. The second layer, the box-regression layer ("reg"), outputs the four coordinates of a bounding-box surrounding the predicted object.

The algorithm parametrizes region proposals relative to "k" reference boxes centered at the sliding window in question, called "anchors". Each one is associated with a scale and an aspect ratio.

In order to understand how we can calculate the RPN loss, it is necessary to explain some concepts before. An Intersection over Union (IoU) overlaps measures how close a given bounding-box is to another bounding-box. It calculates the area of overlap over the area of union between the two boxes. Initially in Faster R-CNN, it is calculated the IoU between the anchors and the ground-truth boxes (the annotated boxes involving the real objects), and the result is used to classify the anchors as positive, negative or neutral, depending on its value.

The method calculates the RPN loss as a combination of the classification loss and the regression loss. The first one takes into account the predicted probability that an anchor is an object and the ground-truth label calculated as described above. The regression loss depends on the real and predicted coordinates.
4.3.2 Second Stage

Before the execution of the object detection, the algorithm sends the generated proposals to a layer that performs a down-sampling over them, called the pooling layer. In this step, the corresponding "Regions of Interest" (ROIs) of non-uniform sizes are extracted from the feature map created by the head CNN, generating small feature maps of fixed size. The system sends these square feature maps through a neural network, that uses the proposals classified as foreground examples to find the correct object class and to adjust the bounding box around it.

In a similar way to the RPN loss, the Global Loss consists of the Classification Loss and the Bounding-Box Regression Loss. While the RPN layer loss deals with only two classes (foreground and background), the global loss deals with all the object classes plus the background. The classification loss takes into account the probability distribution over all the classes and the ground-truth class, while the localization loss, calculated only for the object classes, receives as input the regression offsets predicted for the real class and the real regression offsets for this class.

At the moment that the process is training the classification layer, the error gradients propagate to the RPN network as well, given that the ROI box coordinates used during the pooling are outputs from the RPN network. During the back-propagation, the error gradient propagates until the RPN layer, through the pooling layer.
5 EXPERIMENTS AND ANALYSIS

5.1 Experimental Setup

All the experiments performed in this work were executed on a machine with a Tesla K80 GPU, 6 CPU cores, and 56 GB of RAM. In order to run the experiments, it was first necessary to prepare and set up the environment.

The Faster R-CNN algorithm used in this work was the one referenced in the by Girshick et al. [8], implemented using the programming language python by the own author, Ross Girshick (available in [23]). Before using this code, it was necessary to install some dependencies, like CUDA, for using the GPU power, and Caffe, an open deep learning framework. Then, the Faster R-CNN code itself was compiled and modified according to the experiments.

It was also implemented the evolution strategy algorithm, also using the python language, where the passed arguments are the evolution strategy parameters. These parameters are fitness function, individual length, initial individual, number of generations, population size, target fitness value, and the information whether the fitness value should be maximized or minimized.

The primary dataset used in the experiments is the panoramic database SUN360 (available in [24]), composed of thousands of images of up to 9104x4552 pixels resolution covering 360x180-degree views. These images belong to different environments, from both indoor and outdoor situations. However, the ones used in this work, all of them representing indoor contexts, were: living room, airplane interior, bedroom, museum, office, and others that could not be classified in any of the previous ones. It was used a dataset containing 506 manually annotated images from the five different environments cited above, that was shuffled and then divided into train and test dataset. The training dataset is composed of 380 images, and the test dataset is composed of 126 images.

Likewise, at some of the experiments, it was used the PASCAL VOC 2007 database (available in [25]), composed by 9963 annotated images containing 20739 images of 20 classes. These classes, well-known in the domain and extensively used, are: aeroplane, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train and TV monitor.

Before using the images from the SUN360 dataset in the training and test stages, it
was necessary to annotate the objects present at them. In order to do these annotations, it was used the software LabelImg (available in [26]), that allows the creation of rectangular annotations, generating a file for each annotated image. In this work, the images on the SUN360 dataset were manually annotated using this software. The PASCAL VOC dataset, in contrast, is already annotated, so this step is not necessary.

5.2 Performance Measurements

Before demonstrating the results from the experiments, it is necessary to explain some theory related to the metrics used to evaluate the results. The training metric from the Faster R-CNN algorithm is the global training loss from the type L1, and the inference metrics include the Intersection Over Union and the Average Precision. These metrics are used to evaluate the performance of the object detection algorithm through the evolution strategy.

5.2.1 Training Loss

In the Faster R-CNN algorithm [8], the multi-task loss calculated during training is defined as the addition of the classification loss and the regression loss, as shown in equation 5.1. The classification loss measures the error related to the object’s class. The regression loss measures the error related to the predicted bounding-box position when compared to the correct object location.

\[ L(\{p_i\}; \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^+) + \lambda \frac{1}{N_{reg}} \sum_i p_i^r L_{reg}(t_i, t_i^+) \] (5.1)

5.2.2 Average Precision

From each experiment and each analyzed group of pictures (each environment), it was computed some results, in order to calculate the experiment performance:

- **True Positive (TP)** = The correct detection, or when a detection corresponds to the actual object.
- **False Positive (FP)** = When a detection was performed incorrectly, meaning that the system detected an object where there was none.
By using the true positives and false negatives results, the implemented algorithm calculated the precision, one of the most used performance measurements in object detection. It measures the accuracy of the predictions, or, in other words, the percentage of correct responses. Equation 5.2 describes how to calculate the precision.

\[
Precision = \frac{TP}{TP + FP}
\]  

(5.2)

On the other hand, recall is the ability of a model to find all the relevant cases (all the ground truths). As shown in the Equation 5.3, it can be calculated as the percentage of true positives detected among all ground truths.

\[
Recall = \frac{TP}{TP + FN}
\]  

(5.3)

Using the precision and recall measurements, it is possible to calculate the Average Precision (AP), which is the precision averaged across all recall values between 0 and 1.

5.3 **Experiment 1 - Faster R-CNN Inference only**

The first performed experiment involves only the Faster R-CNN algorithm, not using Evolution Strategy, in order to generate data for comparison. It used a model trained over 9963 planar images from the PASCAL VOC 2007 database. The trained model was used to run an inference over the equirectangular images from the SUN360 dataset.

In order to reduce the experiment space, it was chosen only one class to perform the detection: “person”. The input from the experiment was the test dataset created for this work, along with the pre-trained model. The dataset is composed of 126 manually annotated panoramic images and was created as described in section 3. The output was the same images with the bounding box over the object detections.

The mean Average Precision obtained in this experiment was 68.67%, a result compatible with the results obtained in the literature. It states that the problem of object detection over panoramic images is currently challenging, because of the geometric distortions present on panoramic images. Figure 10 illustrates an example of output from the first experiment, where the algorithm was able to detect objects from the class "person" correctly.
Figure 10: Positive detection example on Experiment 1

Figure 11, however, shows an example of an output image from the first experiment where the detection was not very accurate because of the distortions present in the image. In this image, it is possible to notice several people that the algorithm is not able to detect.

Figure 11: Negative detection example on Experiment 1
5.4 **Experiment 2 - Evolution Strategy with Faster R-CNN algorithm**

The second performed experiment involves the hybrid system using the Evolution Strategy and the Faster R-CNN algorithm. It implemented an Evolution Strategy where the individual consists of the training parameters from the Faster R-CNN algorithm. In the first step, the population is initialized using a determined individual, containing the training parameters used in the previous experiment. At each generation from the implemented strategy, after the mutation of the individuals, the Faster R-CNN model is trained over 380 panoramic images from the SUN360 using the new parameters, and this model is used to run the inference over 126 images the SUN360 dataset. The evolution strategy uses the average precision value obtained with this inference as the fitness value.

The configuration from the Evolution Strategy used in this experiment is:

- Fitness value: Mean Average Precision
- Individual length: 4
- Individual contents:
  - RPN positive overlap: It is the minimum percentage of the Intersection over Union so that the training example is considered positive.
  - RPN batch size: It is the number of images to be used in a batch of the Region Proposal Network stage from the algorithm.
  - Batch size: Number of regions of interest (ROIs) generated per image at each batch. This value states the number of ROIs to work through before updating the internal model parameters. Used in the detection stage.
  - Bbox threshold: Overlap required between an ROI and ground-truth box so that the algorithm uses the ROI as a bounding-box regression training example.
- Initial individual:
  - RPN positive overlap: 0.7
  - RPN batchsize: 256
  - Batch size: 128
  - Bbox threshold: 0.5
- Number of generations: 14
- Population size: 1
- Target fitness value: 100%

The training dataset for the Faster R-CNN algorithm was composed of 380 manually annotated images from different environments of the SUN360 dataset, containing objects from the class "person". The test set was composed of other 126 annotated images, also from the SUN360 dataset. The output from each generation of the evolution strategy was the same images with the bounding box over the object detections.

At each generation, the neural network was trained using the new parameters for 10000 iterations. At each iteration, the Faster R-CNN algorithm [8] computes the global loss defined in equation 5.1. At this experiment, the evolution strategy ran for 14 generations, and, for each one, it was computed the boxplot for the training loss. Figure 12 shows the evolution of the training loss through the generations.

![Results Training Loss](image)

Figure 12: Results of the Training Loss for Experiment 2
Based on the results, it is possible to realize that the generation that produced the smallest loss was number 12, and the generation that had the biggest loss was the number 6. Figure 13 shows the loss evolution during generation 6, and Figure 14 shows the loss evolution during generation 12.

![Figure 13: Loss per Iteration at Generation 6](image)

After each training performed in the evolution strategy, the Faster R-CNN inference computed the mean Average Precision. The final obtained mean Average Precision was 69.37%. The result is very similar to the one obtained in experiment 1, which shows that the proposed solution using the evolution strategy could not evolve as much as we wanted.

This result can be explained by the number of generations in the evolution strategy. Due to the long time necessary for each training, it was possible to execute the implemented algorithm for only 14 times, which was not enough for the strategy to evolve and find a configuration for the object detection algorithm that improved the fitness value significantly. Another possible reason for the low obtained result is the step value for performing the reproduction at the population. The used value may not be enough to
explore the search space efficiently. Moreover, it is possible that the training parameters chosen in this situation were not the best ones. There is the possibility that there are other parameters from Faster R-CNN that may be used in this strategy and generate better results at the object detection task over panoramic images.
6 CONCLUSION AND FUTURE WORKS

An essential task in the Machine Learning domain is the object detection task, which includes locating the object and classifying it. However, when applied over different types of images, this algorithm may present different results. For example, the panoramic images present geometric distortions that usually prevent state-of-the-art object detection algorithms from obtaining good results. In order to overcome these distortions, one possibility is to use different training parameters when working with this kind of image.

The presented work proposes a hybrid system for object detection at panoramic images, composed of an evolution strategy and an object detection algorithm. The proposed solution had the objective of using an evolution strategy to discover the best training parameters for the object detection algorithm when applied to panoramic images. In this work, it was chosen the Faster R-CNN algorithm to perform the object detection task. The results obtained with the experiments were analyzed and compared to the use of a state-of-the-art object detection algorithm. The present work represents a method that explores the search space to find the best solution. This exploratory work may be adjusted in order to be applied in different situations and may be improved in order to achieve better results.

For future works, there are many possibilities for improving the developed solution. One possibility is to run the evolution strategy through a more significant number of generations so that it has more time to discover the best solution. Another one is to change the step in the evolution strategy. This step regulates how much the mutation will change the individuals, and this solution may allow a better exploration of the search space. Besides, it is possible to try the developed method using different training parameters as an individual’s genotype.
BIBLIOGRAPHY


[12] ZHANG, M.; CIESIELSKI, V.; ANDREAE, P. A domain-independent window ap-
proach to multiclass object detection using genetic programming. EURASIP Journal

works using an evolution strategy algorithm. Tunnelling and Underground Space Tech-

#130731.

2002. ISSN 1567-7818. Available at: <https://doi.org/10.1023/A:1015059928466>.

line; accessed 23-August-2019]. Available at: <https://towardsdatascience.com/object-
detection-using-deep-learning-approaches-an-end-to-end-theoretical-perspective-
4ca27eee8a9a>.

[17] PATHAK, A. R.; PANDEY, M.; RAUTARAY, S. Application of deep learning for ob-
International Conference on Computational Intelligence and Data Science. Available at:

[18] OUAKNINE, A. Review of Deep Learning Algorithms for Ob-
jekt Detection. 2019. [Online; accessed 23-August-2019]. Available at:
<https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-
detection-c1f3d437b852>.

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Jun
2016. Available at: <http://dx.doi.org/10.1109/CVPR.2016.91>.

ence, Springer International Publishing, p. 21–37, 2016. ISSN 1611-3349. Available at:
<http://dx.doi.org/10.1007/978-3-319-46448-0\_i>. 


