UNIVERSIDADE DO VALE DO RIO DOS SINOS — UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA
NÍVEL MESTRADO

UÉLISON JEAN LOPES DOS SANTOS

BRAIN ACTION: A BRAIN-INSPIRED HUMAN ACTION RECOGNITION MODEL BASED ON OBJECT RECOGNITION

São Leopoldo
2020

S237b Santos, Uélison Jean Lopes dos.

Brain action : a brain-inspired human action reognition model based on object recognition / por Uélison Jean Lopes dos Santos. – 2020.

94 f. : il. ; 30 cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, RS, 2020.

“Orientador: Dr. Rodrigo da Rosa Righi”.

1. Reconhecimento de ações humanas. 2. Reconhecimento de objetos. 3. Reconhecimento de padrões. 4. Aprendizado de máquina. I. Título.

CDU: 001.85

Catalogação na Publicação (CIP):
Bibliotecário Alessandro Dietrich - CRB 10/2338

ACKNOWLEDGEMENTS

First, I thank God for all that I have overcome and achieved in my life. I thank my wife because, without her, none of this would be possible and also my family, who has always been with me. Without them, I wouldn’t get to where I am. Also, I thank Prof. Dr. Rodrigo da Rosa Righi, my advisor, who provided me with knowledge, commitment, and guidance at all times I needed. I also acknowledge the postgraduate colleagues from both UNISINOS and FAU, who in many ways, helped me in my academic development. Finally, I thank UNISINOS for the scholarship that allowed me to continue my academic education.

ABSTRACT

A great variety of applications require HAR (Human Action Recognition) information as input. A topic that is applied in such various areas becomes particularly prevalent recently because of their explosively emerging real-world applications. Many works tried to understand actions by only observing the actor's poses, introducing methods to model human appearances and poses, trying to more robust features. However, several actions may be performed with comparable postures, and these strategies ignored features, which made them less applicable to recognize complex actions. Although several proposals have already been submitted, HAR is a problem that is still far from having a definitive solution. The solutions focus on exploring different techniques for getting features and enabling machine learning algorithms to identify actions. However, the variety of possible human actions, the small number of dataset examples, and the complexity of the task mean that several studies are still required to reach a final solution. Simultaneously, as we try to make a computer understand actions in videos, neuroscientists are trying to understand how the human brain recognizes activities. The analysis shows that object recognition is a hard task, even to the brain. Although, studies suggest that the brain's algorithm is relatively simple and most likely processes the visual input only once. In this thesis, we explore what we know so far about how the human brain recognizes actions to simulate this same behavior on a computer. A model that proves to be robust can serve as the basis for developing solutions in the most varied branches. For this, we studied the Neuroscience and Physiology areas for information about how the human brain works. From this information, we developed the Brain Action model to simulate this behavior and introduced an algorithm workflow to implement this model on a computer. During the development of the research, we tried to understand how other proposals with similar methods solve the same problem, as well as solutions that explore other different techniques. We have gathered this knowledge to propose a model that can explore techniques that are already accepted in state of the art with the human mind's recognition of actions. This proposal aimed to develop a model that has as input RGB videos, and by identifying the positioning and movements of the elements in the scenes, and using only the relationship of this information, be able to recognize human actions, targeting applications in various domains. We followed this research by implementing the model in a challenging surgical operations HAR task, evaluating it with the state-of-the-art metrics. We built our surgical dataset with seven different classes during this process, tested the model with three different machine learning classification methods, and achieved 44.1% of correctly classified actions applying cross-validation. Our contributions are threefold: (I) A new biological inspired HAR model, (II) a new movement feature extraction design, and (III) a HAR implementation for surgery action recognition scenario.

Keywords: Human Action Recognition. Object Recognition. Pattern Recognition. Machine Learning.

RESUMO

Uma grande variedade de aplicações de software que requerem informações de reconhecimento de ações humanas (HAR) como entrada. Um tópico que é aplicado em diversas áreas se tornou prevalente recentemente por conta de aplicações do mundo real que emergem rapidamente. Muitos trabalhos acreditavam que as ações humanas podiam ser entendidas apenas observando as poses dos atores. Vários autores introduziram métodos para modelar aparências e poses humanas, produzindo características mais robustas para utilizar no reconhecimento. No entanto, várias ações podem ser executadas com poses comparáveis, e essas estratégias ignoram esta característica, o que os torna menos aplicáveis ao reconhecimento de ações complexas. Embora várias propostas já tenham sido concebidas, HAR é um problema que ainda está longe de ter uma solução definitiva. As soluções concentram-se em explorar diferentes técnicas para obter recursos e permitir que algoritmos de aprendizado de máquina identifique as ações. No entanto, a variedade de ações humanas possíveis, o pequeno número de exemplos de conjuntos de dados e a complexidade da tarefa significam que vários estudos ainda são necessários para se chegar a uma solução final. Simultaneamente, enquanto tentamos fazer um computador entender as ações em vídeos, os neurocientistas estão tentando entender como o cérebro humano reconhece as atividades. A análise mostra que o reconhecimento de objetos é uma tarefa difícil, até mesmo para o cérebro. No entanto, estudos sugerem que o algoritmo do cérebro é relativamente simples e provavelmente processa a entrada visual apenas uma vez. Nesta tese, exploramos o que sabemos até agora sobre como o cérebro humano reconhece ações para simular esse mesmo comportamento em um computador. Um modelo que se mostra robusto pode servir de base para o desenvolvimento de soluções nos mais diversos ramos. Para isso, estudamos as áreas de Neurociência e Fisiologia para obter informações sobre o funcionamento do cérebro humano. A partir dessas informações, desenvolvemos o modelo Brain Action para simular esse comportamento e introduzimos um fluxo de trabalho de algoritmo para implementar esse modelo em um computador. Durante o desenvolvimento da pesquisa, procuramos entender como outras propostas com métodos semelhantes resolvem o mesmo problema, bem como soluções que exploram outras técnicas distintas. Reunimos esse conhecimento para propor um modelo capaz de explorar técnicas que já são aceitas no estado da arte com o reconhecimento das ações pela mente humana. Esta proposta teve como objetivo desenvolver um modelo que tenha como entrada vídeos RGB, e por meio da identificação do posicionamento e movimentos dos elementos nas cenas, e utilizando apenas a relação dessas informações, seja capaz de reconhecer ações humanas, visando aplicações em diversos domínios. Seguimos essa pesquisa implementando o modelo em uma tarefa HAR de operações cirúrgicas desafiadoras, avaliando-o com as métricas de última geração. Construímos nosso conjunto de dados cirúrgicos com sete classes diferentes durante esse processo, testamos o modelo com três métodos de classificação diferentes de aprendizado de máquina e alcançamos 44,1 % de ações classificadas corretamente aplicando validação cross-fold. Nossas contribuições são três: (I) Um novo modelo HAR de inspiração biológica, (II) um novo projeto de extração de informações de movimento e (III) uma implementação de HAR para o cenário de reconhecimento de ação cirúrgica.

Keywords: Reconhecimento de Ações Humanas. Reconhecimento de Objetos. Reconhecimento de Padrões. Aprendizado de Máquina.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

CNN	Convolutional Neural Network
DNN	Deep Neural Network
FAU	Friedrich-Alexander-Universitat
FPN	Feature Pyramid Network
GPU	Graphic Processing Unit
HAR	Human Action Recognition
HOI	Human Object Interaction
IoU	Intersection over Union
KNN	K-nearest neighbors
LOP	Linear Ordering Problem
LSTM	Long Short-term Memory
NLP	Neuro-linguistic Programming
NN	Neural Network
R-CNN	Region-based Convolutional Network
ReLU	Rectified Linear Unit
RoI	Region of Interests
SGD	Stochastic Gradient Descent
SSD	Shot MultiBox Detector
SVM	Support Vector Machine
mAP	Mean Average Precision

CONTENTS

1 INTRODUCTION

Video cameras have become more accessible over the last years, and now there are a large number of videos recorded with much different information. With this advent, obtaining actions or activities of humans in videos is undergoing intense study in computer vision research. Depending on the complexity, human movements can be classified into four types: gestures, actions, interactions, and grouped activities. Gestures are basic movements of a person's body part and are the atomic elements representing a person's movement. Raising a leg, or moving the head are examples of gestures. An action is a sequence of human body moves that may involve diverse body parts concurrently in an organized manner.

Interactions are activities that involve two or more persons and/or objects, as a fight between individuals, or someone stealing a wallet from another. Lastly, grouped activities are the activities produced by groups composed of multiple persons and/or objects like a group of people dancing, or a football team playing (AGGARWAL; RYOO, 2011). In this work, we will focus on the last three, since they are more complex and essential to a Human Action Recognition (HAR) System (ANN; THENG, 2014; ZHANG et al., 2019).

Knowing the difference between the human movements, we can understand what an action recognition system should do (MOESLUND; HILTON; KRÜGER, 2006; GOLESTANI; MOGHADDAM, 2020): It needs to obtain the action data; understand the human movement and interaction; classify that action within a predefined category. The nature of the human movement is extensive and intricate. It can be a quick hand gesture or a complex action, including multiple members with a long duration. We also have to consider the interaction between various humans or with complex objects. This broad domain makes HAR a challenging task.

1.1 Motivation

A great variety of software applications require HAR information as an input: Visual Surveillance, Human-Computer Interaction, sentiment analysis, Video Analytics, Ambient Intelligence, Video Indexing, Robotics, Driver-assisting, Health care and so on (HOLTE et al., 2012; YANG; RAMANAN, 2013; SHOTTON et al., 2011; SUN; KOHLI; SHOTTON, 2012; TOSHEV; SZEGEDY, 2014; DANTONE et al., 2014; SARAFIANOS et al., 2016; CHEN et al., 2017; HU et al., 2019; JACOBY et al., 2018). A topic that is applied in such various areas becomes particularly prevalent recently because of their explosively emerging real-world applications. On the other hand, it is very challenging to infer actions from still images without any motion cues, which are found to be useful in recognizing activities (ZHU; HU; ZHENG, 2018). Many works believed that actions could be understood by only observing the actor's poses. Several authors (YAO; KHOSLA; FEI-FEI, 2011; HU et al., 2013, 2015; KHAN et al., 2013; MAJI; BOURDEV; MALIK, 2011; SHARMA; JURIE; SCHMID, 2013; LUDL; GULDE; CURIO, 2019; KIM et al., 2019) introduced methods to model human's appearances and poses, producing more robust features. However, several actions may be performed with comparable postures, and these strategies ignored features, which made them less applicable to recognize complex actions (ZHU; HU; ZHENG, 2018).

Simultaneously, as we try to make a computer understand actions in videos, neuroscientists are trying to understand how the human brain recognizes activities. Taylor (TAYLOR et al., 2014) studied and reviewed the construction of contour detection and combination in humans, while Kosilo (KOSILO et al., 2013) describes new experiments intended to disconnect the effects of color and contrast. Schneider's (SCHNEIDER, 1969) suggests that two different paths support visual orientation toward an object. However, object recognition is a hard task, even to the brain, since half of the non-human primate neocortex is dedicated to visual processing (FELLEMAN; VAN, 1991). The brain distinguishes among different actions, and it does so in a manner that is invariant to variations in 3D viewpoint (ISIK; TACCHETTI; POGGIO, 2017). However, when form or motion information is removed from the stimulus set, we perceive a reduction and delay. Although we have several studies in the area and discovered a lot, it is still uncertain how the brain recognizes objects and actions.

How the human body biologically works is studied by the Computer Science area to develop solutions. Studies made, for example, in how the human vision detects edges(SHAPLEY; TOLHURST, 1973) enable the development of new edges detectors in the computer vision area (XIAO; MA; XIA, 2013). The Neural Networks (MAASS, 1997) is another example of how the exploration of biological strategies could also apply to problem solutions for computer science, and today is one of the main tools for the development of artificial intelligence applications. Although several proposals have already been submitted, HAR is a problem that is still far from having a definitive solution. The solutions focus on exploring different techniques for getting features and enabling CNN to identify actions. However, the wide variety of possible human actions, the small number of dataset examples, and the enormous complexity of the task mean that several studies are still needed to reach a final solution.

A model that proves to be robust can serve as the basis for developing solutions in the most varied branches. In this paper, we explore what we know so far about how the human brain recognizes actions to simulate this same behavior on a computer. For this, we study the area of Neuroscience and Physiology for information on how the human brain works. From this information, we seek to develop a model to simulate this behavior and then propose a workflow with algorithms to implement this model in a computer. During the development of the research, we try to understand how other proposals with similar methods solve the same problem, as well as solutions that explore other different techniques. We have gathered this knowledge to propose a model that can explore techniques that are already accepted in state of the art with the human mind's recognition of actions.

This research also aims to support a going on a project between Unisinos SoftwareLab, Siemens, and FAU. The surgical workflow project developed a prototype for monitoring activities into a medical scenario, and a system for recognition of actions is needed. In this environment, overlapping multiple people and members is a significant challenge to be solved. Pose solutions face several difficulties in this scenario, where there is significant complexity in identifying each physician's upper limbs. Besides, the use of multiple RGBD cameras ultimately increases the computational cost, making real-time recognition with current processing power impossible. Finally, very similar poses for entirely different actions make the approach using pose inaccurate, which motivated us to perform object recognition and use this information along with scene movement, applying the techniques that the human brain uses to infer the action being performed.

1.2 Research Question and Hypotheses

The research question that the proposed model seeks to answer is as follows:*How a model inspired by the functioning of the human brain, with the support of machine learning, can recognize and understand the actions of human beings?*

This proposal aims to develop a model that has as input RGB videos, and based on how the human brain detects actions perform the recognition of human activities. We intend to achieve this by identifying the positioning of the elements in the scenes and using the relationship of this information to recognize human actions, targeting applications in various domains. This proposal also intends to implement this model within a real-world problem, evaluating it with the metrics used by state of the art to validate the model.

Based on that, this research's hypothesis can be presented as follows: *A model of HAR based on how the human brain work is capable of recognizing actions better than the Zero Rule benchmark.*

To understand that the model can recognize actions, we understand that the proposed model should be better than the Zero Rule benchmark. This method is used to evaluate classification problems, where the most frequently occurring class is set as the output. For example, if 80% of the data in a dataset have the same class, the benchmark will be better than 80%.

1.3 Objectives

This work has a general objective: *Develop a model for recognizing human actions based on biological perception interpreted by the brain with the support of machine learning.*

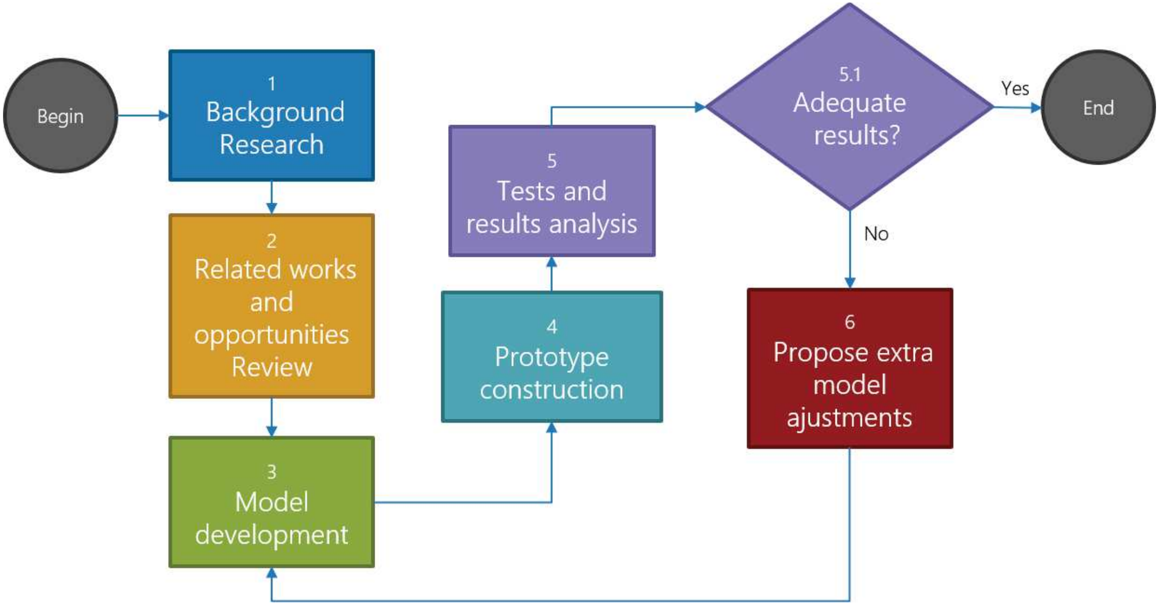
To achieve the overall objective, the following specific objectives were defined:

(i) Investigate the functioning of the human brain regarding the recognition of human actions;

(ii) Conduct a study by compiling research already developed in the HAR area;

(iii) Develop a HAR model based on the information collected with the support of machine learning;

Figure 1 – Research workflow steps. The tasks 3,4,5 and 6 were done more than once, to implement improvements into the model.



Source: Elaborated by the author.

(iv) Contribute to the academic community by sharing data sets and algorithms, as well as scientific publications regarding the possible applications of the proposed model.

1.4 Research Development

The research development is taking place according to the flowchart presented in figure 1. In the flowchart, there are five steps, as follows: (1) Background; (2) Related Works; (3) Model; (4) Proposal Assembly; (5) Results and Analysis.

Initially, a study of the theories involved in the research theme was conducted to form the theoretical framework, referred to as background. Then, the survey phase of the works related to the research theme occurred. This step aimed to find works with similar objectives to the present research, thus identifying possible gaps. The third step consisted of proposing and developing a model that would fill the differences found in the related works, answer the research question, and meet the objectives of the work. The following steps consist of implementing the proposed model, testing and evaluating the implementation, and obtaining the results. The testing and analysis step may generate modifications to the proposed model as needed.

1.5 Text organization

The proposal is organized into five sections. Initially, chapter 2 presents fundamental concepts for understanding the rest of the work. Then, the chapter 2 presents the works related to the themes of this research, demonstrating a comparison with this proposal. This chapter aims to present what already exists in state of the art, as well as to identify where there are gaps to be filled. Chapter 4 presents the model proposed in this paper that aims to fill the gaps identified in the previous chapter, as well as to achieve the objectives of this work. Chapter 5 presents the implementation proposition of the model and the evaluation experiments. In this chapter, we present the modifications done to the model. Finally, chapter 6 presents the conclusions obtained and the expected conclusions of this research.

2 BACKGROUND

In this section, some concepts for understanding the proposal will be discussed. These concepts will serve as the basis for the rest of the work. First, we will present some subtopics from Machine Learning that are fundamental. Then, we will focus on topics that are specific from Neural Networks that were key to this work. However, given the complexity of the issues, the intention in this section is to provide basic knowledge covering the proposal.

2.1 Supervised/Unsupervised learning

In the learning task, we compute a function that measures the error between the output scores and the desired scores. The algorithm then modifies its internal parameters to try to reduce this error (KOTSIANTIS; ZAHARAKIS; PINTELAS, 2007). These adjustable parameters are often referred to as weights and are real numbers that can be comprehended as controls that define the input-output function of the machine (LECUN; BENGIO; HINTON, 2015).

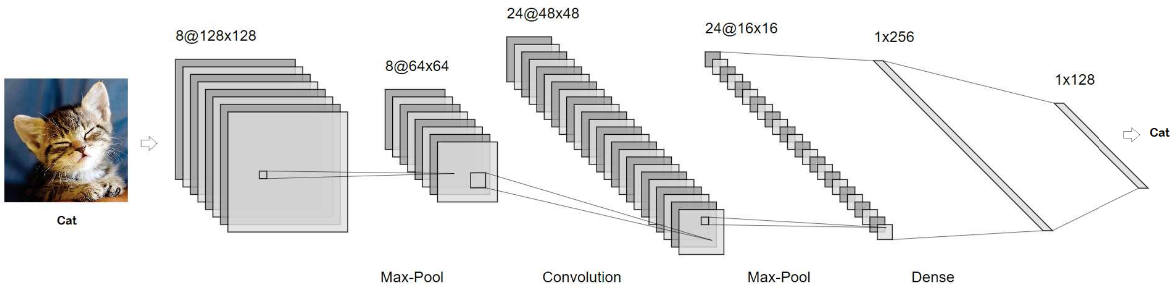
Machine learning algorithms can be either supervised or unsupervised (ALLOGHANI et al., 2020). The difference between the two main classes is the presence of labels in the training dataset for supervised learning. Simultaneously, the unsupervised involves the use of pattern recognition techniques without the involvement of a target attribute to find patterns in the data.

The most popular class of machine learning, deep or not, is supervised learning (LECUN; BENGIO; HINTON, 2015). For an image classification task, one usual approach is to collect a large dataset of images with labeled categories, as illustrated on 2. Then, during the training part, the algorithm will explore the attributes of the image. The output is a score vector for each category (ALLOGHANI et al., 2020). Technically, supervised techniques perform analytical tasks first using training data and subsequently construct contingent functions for mapping a new instance of the attribute. To be able to do this task, the algorithms require presuppositions of maximum settings for the desired outcome and performance levels (LIBBRECHT; NOBLE, 2015).

2.1.1 Classification task

Classification is a type of supervised learning, and a significant part of machine learning tasks are classification tasks. Those tasks require a model, that can understand high-dimensional data with complex structure and summarize it with a category or label (GOODFELLOW; BENGIO; COURVILLE, 2016). In classification, a model is trained on seem and known examples, and for new observations, it predicts a category. There are different kinds of classifiers that use different techniques to classify the data. When classifying data, it is difficult to predict by just seeing the data what kind of classifiers to use. As a good practice, in this work, we will use different techniques and compare their results using a Linear classifier, Decision Trees, and

Figure 2 – An extensive set of labeled images is used in the training task to adjust the network weights. In this process, the network learns what attributes are essential to classify an image correctly by adjusting its weights and seeing the labels' labels.



Source: Elaborated by the author.

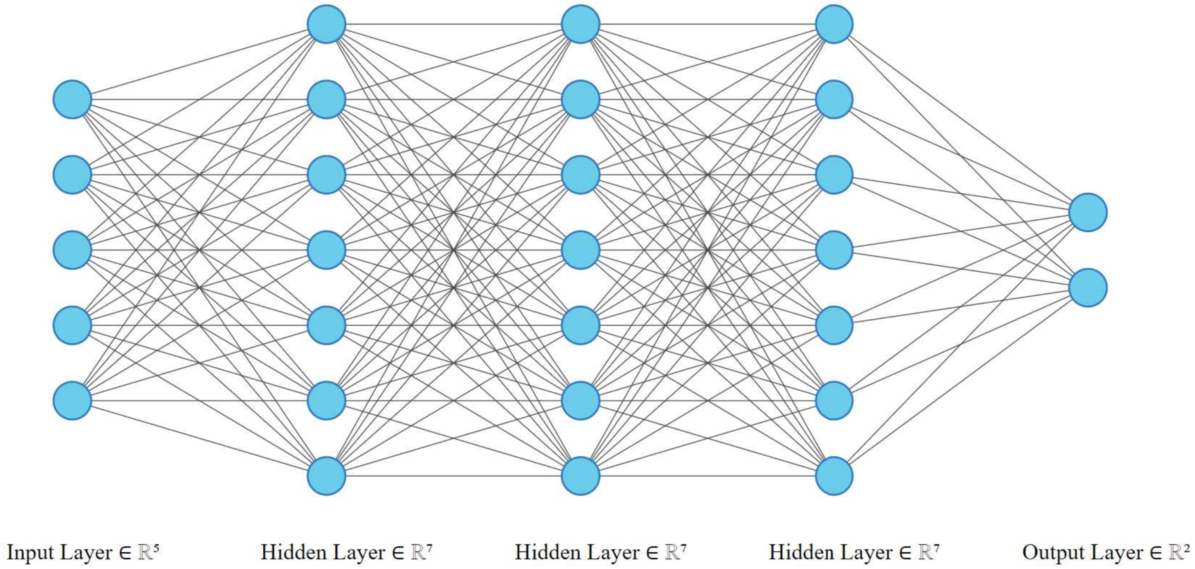
Neural Networks.

Linear classifiers, such as Naive Bayes, do the classification based on the linear value of characteristic combinations. Those divisions in data created different sections, where each section has a label. Decision trees learning uses a decision tree as a predictive model to classify the conclusions about the observed data. The leaves in the tree represent the category labels that we want to predict, and the branches represent the logical conjunctions on data. Neural Networks are models based on artificial neurons that receive data, process it, and send to connected neurons the values. Each neuron typically has a weight that changes its weight at a connection (HARRINGTON, 2012). When training a Neural Network, we search for the best weights that can represent better the data.

2.2 Deep Learning

It is challenging for a computer to recognize the meaning of raw input data, such as pixel values, or who is represented by an image. The function to map this collection is very complex, insurmountable if tackled directly. Deep Learning resolves this obstacle by dividing this problem into smaller pieces, less difficult to solve (GOODFELLOW; BENGIO; COURVILLE, 2016). It allows computational models that are composed of multiple processing layers to learn representations of data with various levels of abstraction (LECUN; BENGIO; HINTON, 2015). The input data is presented to the visible layer, so-called, because it includes the variables that we can observe. Then a series of hidden layers extract from the image abstract features. They are called hidden because their values are not given in the data; the model determines them. Those layers will explore edges, corners, contours, gradients, and other possible image features. The result is then given to the last layer, which is responsible for based on the outcome of previous layers, identify objects in the image. In the end, the description of the image is given based on the recognized object (GOODFELLOW; BENGIO; COURVILLE, 2016). In figure 3, we present an underlying DNN architecture.

Figure 3 – Deep Neural Network Example. In a Neural Network (NN), there will always be an input and output layer, while we can have zero or more hidden layers. The main difference between Neural Network and a Deep Neural Network (DNN) is the number of hidden layers, in a DNN, it must be two or more. The learning process of a NN is performed with the layers. Hidden layers reside in-between input and output layers, and the larger the number of hidden layers in a neural network, the longer it will take to produce the output and the more complex problems it can solve.



Source: Elaborated by the author.

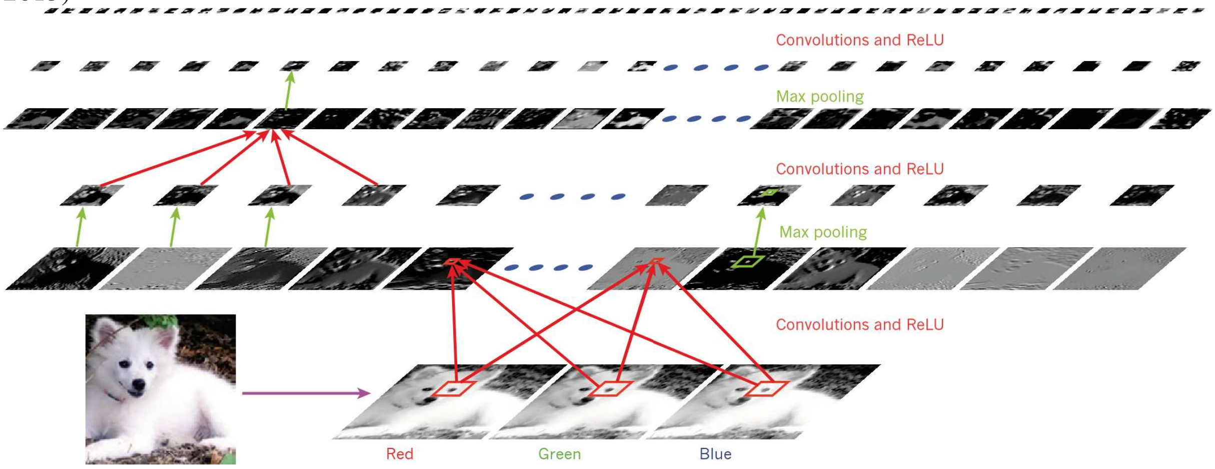
2.2.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a subtype of the discriminative deep architecture (AREL et al., 2010) designed to process data that come in the form of multiple arrays. For example, an RGB color image composed of three 2D arrays containing pixel intensities (LECUN; BENGIO; HINTON, 2015).

The animal visual cortex organization inspires the architecture of CNNs. In the 1960s, Hubel and Wisel (HUBEL; WIESEL, 1962) introduced a theory called receptive fields. They remarked that intricate arrangements of cells were included in the animal visual cortex in the administration of light detection in overlapping and small sub-regions of the vision area. Moreover, the model Neocognitron was proposed in (FELZENSZWALB et al., 2009) with hierarchically arranged image transformations.

A typical CNN architecture (see figure 4) is structured as a sequence of stages. The first few stages are frequently constituted of convolutional and pooling layers. Units in a convolutional layer are organized in feature maps, within which each unit is connected to local patches in the feature maps of the former layer through a set of weights called a filter bank. The result of this locally weighted sum is then passed through an activation function such as a ReLU (LECUN; BENGIO; HINTON, 2015). There are four critical ideas behind CNNs that benefit from the properties of natural signals: local connections, shared weights, pooling, and the use of many

Figure 4 – Example of a Convolutional Neural Network. Deep Convolutional networks have brought about breakthroughs in processing images, video, speech, and audio.(LECUN; BENGIO; HINTON, 2015)



Source: Adapted from (LECUN; BENGIO; HINTON, 2015).

layers (LECUN; BENGIO; HINTON, 2015).

Notwithstanding the attractive qualities of CNNs and the relative effectiveness of their local architecture, they have still been prohibitively expensive to apply on a large scale due to high-resolution images. Fortunately, modern GPUs, paired with a highly-optimized implementation of 2D convolution, are powerful enough to promote the training of interestingly-large CNNs (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Nowadays, CNNs are successfully applied to handwriting recognition, face detection, behavior recognition, speech recognition, recommendation systems, image classification, and NLP (LIU et al., 2017a).

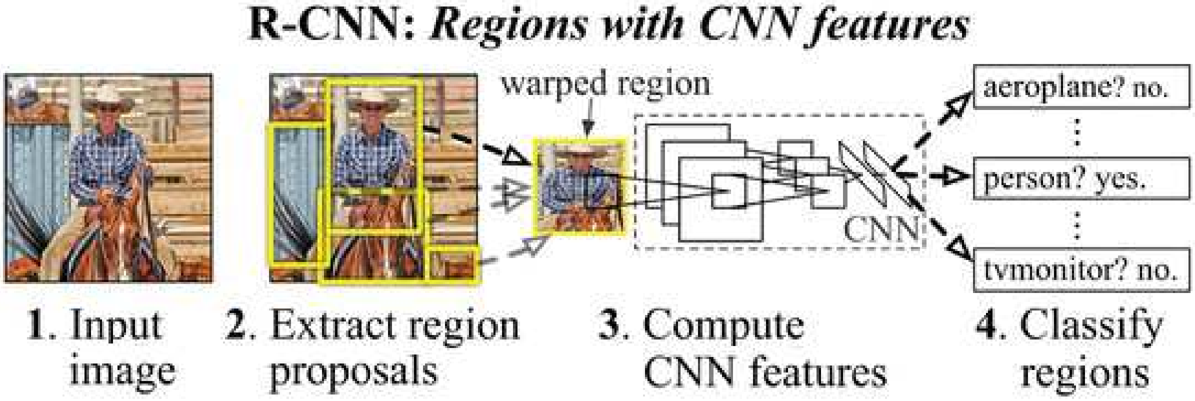
2.2.2 R-CNN

Region-based Convolutional Network (R-CNN) is a particularly successful class of techniques based on bounding-box detection in recent times (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). It takes an image as input and outputs a set of rectangular object suggestions, each with an object detection score (REN et al., 2015). It is composed of a two-stage approach where the first step suggests some candidate's Region of Interests (RoI), and the second achieves object classification. Region-wise features can be fast taken from shared feature maps by an RoI pooling procedure. Feature sharing speeds up instance-level detection and allows recognizing higher-order interactions, which would be computationally infeasible (REN et al., 2015). In figure 5, we present the approach used on R-CNN to classify the objects into an image.

2.3 Object Recognition

Object recognition is a challenging task in computer vision. It is a term to define several related computer vision tasks that involve identifying objects in digital pictures. The definition of

Figure 5 – How a R-CNN architecture deals with the object recognition task.



Source: Adapted from (RAO et al., 2017).

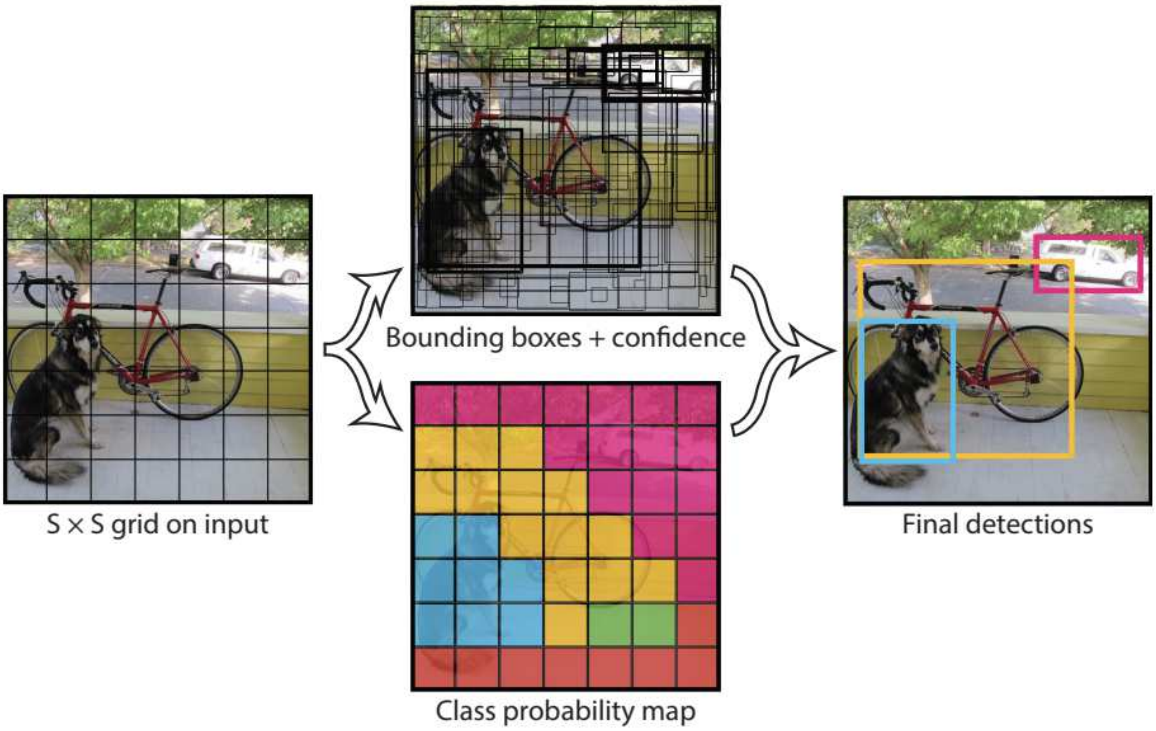
object recognition is the capacity to designate labels to particular objects, varying from precise tags, defined as identification to significant labels, defined as categorization (DICARLO; ZOCCOLAN; RUST, 2012). More specifically, it is the capacity to complete such tasks over a range of identity preserving transformations (such as changes in object's position, size, pose, and background context), without knowing any cue about the objects or their locations (DICARLO; ZOCCOLAN; RUST, 2012).

Each contact with an object is almost unique because of identity-preserving image transformations in the real world. Precisely, the immense collection of images produced by objects that should obtain the same label results from the variability of the environment and the observer. Every object can be encountered at either position on the retina (position variability), at a variety of distances (scale variability), at numerous angles relative to the observer (pose variability), at a range illumination conditions (illumination variability), and in new visual contexts (clutter variability). Furthermore, some objects as bodies and faces are deformable in shape. This frequently requires group varying three-dimensional shapes into a common category to deal with intraclass variability. In summary, each contact of the same object stimulates an entirely distinct response pattern. The task is to establish the equivalence of these response patterns somehow and, at the same time, not confuse any of them with perceptions of all other potential objects (DICARLO; ZOCCOLAN; RUST, 2012).

2.3.1 YOLO

YOLO (REDMON et al., 2016) is an end-to-end single convolutional neural network that detects objects based on class probabilities and bounding boxes prediction. YOLO trains on full images and directly optimizes detection performance. The network divides the input image into grids, and each grid predicts bounding boxes and corresponding confidence. The predicted probabilities weight these bounding boxes. To ensure that the object detection algorithm only detects each object once, they apply non-max suppression, then outputs recognized objects

Figure 6 – YOLO model



Source: Image obtained from (REDMON et al., 2016).

together with the bounding boxes.

Intersection over the union (IoU) is used to calculate the confidence of the prediction. While R-CNN methods look to parts of the image, YOLO looks at the complete picture to make the predictions. Since it is trained with the full images, it implicitly encodes contextual knowledge about classes and their representation. Figure 6 shows the model idea.

YOLO predicts the coordinates of bounding boxes directly using fully connected layers on top of the convolutional feature extractor (REDMON; FARHADI, 2017). The last version of YOLO, YOLO v3 (REDMON; FARHADI, 2018), is a full convolution network that uses the front 52 layers of the darknet-53 (REDMON, 2013) and a lot of residual hopping connections. It employs Leaky ReLU as the activation function and batch normalization as a method of regularization.

2.4 Action Recognition

Video cameras have become more accessible over the last years, and now there are a large number of videos recorded with considerably different information. With this advent, obtaining actions or activities of humans in videos is undergoing intense study in computer vision research. A great variety of applications require this information as an input: Visual Surveillance, Human-Computer Interaction, sentiment analysis, Video Analytics, Ambient Intelligence, Vi-

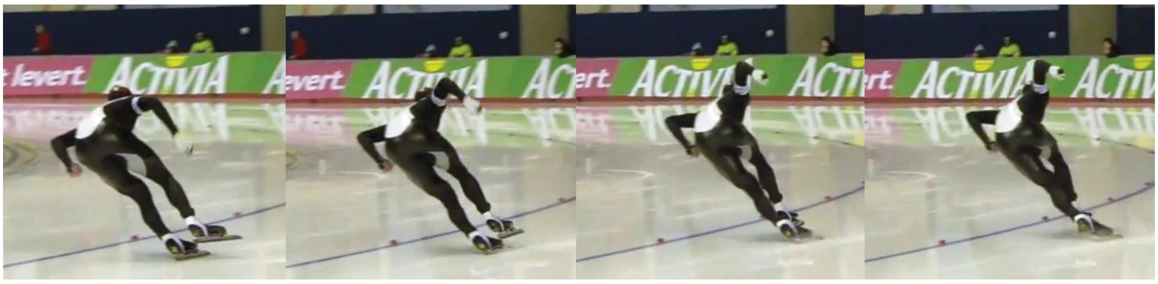


Figure 7 – An action is a group of human body moves and may involve diverse body parts concurrently in an organized way, such as Running, Waving, and Punching. The figure shows an example of a sequence of orderly movements due to skating on ice.

deo Indexing, Robotics, Driver-assisting, Health care and so on (HOLTE et al., 2012; YANG; RAMANAN, 2013; SHOTTON et al., 2011; SUN; KOHLI; SHOTTON, 2012; TOSHEV; SZEGEDY, 2014; DANTONE et al., 2014; SARAFIANOS et al., 2016; CHEN et al., 2017). Depending on the complexity, human movements can be classified into four types: gestures, actions, interactions, and grouped activities (AGGARWAL; RYOO, 2011). Gestures are basic movements of a person's body part and are the atomic elements representing the movement of a person. Raising a leg, or moving the head are examples of gestures. An action is a sequence of human body moves and may involve diverse body parts concurrently in an organized. In figure 7 have an example of skating on ice. Interactions are activities that involve two or more persons and/or objects, as a fight between individuals, or someone stealing a wallet from another. Lastly, grouped activities are the activities produced by groups composed of multiple persons and/or objects like a group of people dancing, or a football team playing (AGGARWAL; RYOO, 2011).In this work, we will focus on the last three, since they are more complex and essential to a Human Action Recognition (HAR) System (ANN; THENG, 2014).

Knowing the difference between the human movements, we can understand what should an action recognition system do (MOESLUND; HILTON; KRÜGER, 2006): It needs to obtain the action data; understand the human movement and interaction; classify that action within a predefined category. The nature of the human movement is extensive and intricate. It can be a quick hand gesture or a complex action, including multiple members with a long duration. We also have to consider the interaction between various humans or with complex objects. This broad domain makes HAR a challenging task since the movement data can be obtained in diverse ways.

3 RELATED WORK

The HAR challenge is being a target of study for a long time, and with the advance of new accessible depth sensors, it became an active area of computer vision research. Recent Deep Learning techniques are capable of remarkable achievements in the field and are evolving at a fast pace. When we review the past solutions, compared with current ones, evolution is evident. While the first works have around 30% accuracy (LIU; YUAN, 2018), current ones are close to 100%. This fast-paced evolution requires that researchers stay updated on breakthroughs. In the next sections, we will present the used survey methodology to review the related works as well as the revision of the selected works.

3.1 Survey methodology

In this section, we describe the study protocol, which details selected methods and highlights the subsequent choices. The presented study introduces a systematic literature review designed to answer defined topics about activity recognition based on the objects' field. This review verifies whether research efforts exist on the studied topic, and presents quantitative evidence (BRERETON et al., 2007; BEELMANN, 2006). The purpose of this work is to review the technology regarding HAR based on objects and recognize promising trends. Considering this, we followed widely acknowledged empirical guidelines (BRERETON et al., 2007; BEELMANN, 2006) for literature reviews, to design and control systematic mapping studies. Furthermore, to decrease threats to validity, we followed the well-documented study rules written in work by Biolchini et al. (BIOLCHINI et al., 2005) and Qiu et al. (QIU et al., 2015).

3.1.1 Research Questions

The definition of research questions is one of the most relevant parts of a systematic review, according to Kitchenham and Charters (KITCHENHAM et al., 2010) and Petticrew and Roberts (BEELMANN, 2006). Consequently, we investigate to catalog and classify the methods correlated to activity recognition, the characteristics, difficulties, challenges, and solutions that are being investigated, and the research opportunities that exist or are emerging. General research questions have been polished into more specific topics to provide a precise organization and thematic examination, as well as to denote encouraging research trends for additional investigation. Two categories were proposed: general questions and specific questions. Next, we listed all the research questions investigated, the search strategy, and composed search strings. We will answer then in section 3.3.

3.1.1.1 General questions

- **GQ1** What are the current action recognition strategies?
- **GQ2** In what context is action recognition being applied?
- **GQ3** What problems are still open?

3.1.1.2 Specific Questions

- **SQ1** What are the major recognition strategies using objects?
- **SQ2** What are the leading workflows to recognize actions using objects?
- **SQ3** What are the main Datasets applied to evaluate the solutions?

3.1.2 Search Strategy

One of the major steps is to obtain a complete set of works related to our research questions. This process required the designation of search keywords and the definition of search scope (BEELMANN, 2006). In the phase of constructing the search keywords, we look up the ideas and topics that define our content, to obtain accurate search results. Kitchenham and Charters (BRERETON et al., 2007), in their report, suggest splitting the research question into distinct phases as investigation units, where their synonyms, acronyms, abbreviations, and alternative spellings are all covered and joined by Boolean operators. Petticrew and Roberts (BEELMANN, 2006) propose the PICOC (population, intervention, comparison, outcome, and context) criteria, which are guidelines to define research units accurately. The main objective was to refine and answer specific research questions acquired from the general ones, by constraining their scope.

3.1.3 Search String

To find the relevant articles for this research, we defined some search strings for the article's databases. They are based on the Key points, Context, and Objective below. Using those strings, we built our Permutations, which means that for each combination of Key Point, Context, and Objective, we created a query and searched for it. Also, to review the current state of the art, we limited the search to the last five years, 2015 - 2019.

Key Point: K= Human Activity, Human Action
Context: C= Feature, Object, Context, Cues

Objective: O= Recognition, Detection, Understanding, Estimation
Permutations: C(n) + K(n) + O(n)

3.1.4 Corpus Selection

To research the state of the art with our search queries, we selected two databases: Databases from Scientific publishers and databases from search engines.

Databases from scientific publishers:

Publisher	Results	After Refinement
IEEE [1]	87	25
Elsevier [2]	16	4
ACM [3]	14	1
Springer[4]	11	6

Databases from search engines:

Publisher	Results	After Refinement
Scopus [5]	27	11
Google Scholar [6]	19	17

In each of those databases, we used the permutations of our search query. In some of them, we could use the advanced search and create a single search, while others required the generation of all the permutations and further yielded results combinations. We refined the articles eliminating the ones who are not into the topic of this research. We obtained 64 articles from all the selected sources, where 9 were duplicates. The final set of reviewed articles comprises 55 works.

3.1.5 Corpus Refinements

All the 55 Articles were reviewed, where six were discarded since their proposal was too different from this work. The other 8 works exhibit surveys that were considered and used to enhance the background but can not be compared to this proposal. Thirteen works presented relevant proposes to this work and will be carefully reviewed in the next sections.

3.2 Selected Works Review

Nie et al. proposed in (NIE et al., 2018), a system based on object recognition and human action recognition to detect dangerous behavior for a child caring robot. To achieve this task, they acquired the images thought a Kinect and used the skeleton data to apply action recognition, the depth images to get the human and objects position, and the RGB images to recognize the objects. For object detection, they employed YOLOnet2 (REDMON; FARHADI, 2017), but since they trained just with images of dangerous objects, they removed the last classification layers. To Human Action Recognition, they applied a CNN based on the Euclidean distances between body joints. They claim that according to their studies in the area, body structure information is more critical than the joint's dynamic variation. To evaluate the CNN, they used NorthWestern-UCLA Dataset and achieved results close to the other previous methods. To evaluate the proposed solution, they prepared a real ambient, reaching an 81.8% success rate.

In (JACOBY et al., 2018), Jacoby et al. proposed a human activity classification for learning environments on RGB videos. They explored the use of color to achieve object recognition in conjunction with the contextualization of objects interaction and applying a separate KNN classifier for each human activity that they are evaluating. In this paper, they explored the color-based segmentation to obtain the objects of interest. Once detected, they extracted motion vectors, that combined with context-based rules, can identify the objects. Based on the recognized object, they checked for full objects in the image. For example, if a keyboard is detected, they check around the identified image for hands. To evaluate the solution, they applied K-fold validation as well tested with CNN. However, they experimented with the solution just for three different actions and with separated classifiers, which can not be a solution for significant problems and can overfit easily.

In (WANG et al., 2019) the authors proposed a context-aware framework to detect human-object interactions. This framework consists of two stages: The first one is responsible for localizing the objects, and they used the FPN paradigm, applying ResNet-50-FPN (LIN et al., 2017) backbone. It's responsible to object bounding box predictions and was pre-trained using ImageNet (DENG et al., 2009) and then trained in the used datasets: V-COCO(GUPTA; MALIK, 2015), HICO-DET (CHAO et al., 2018) and HCVRD(ZHUANG et al., 2018). For the prediction, they fuse scores from a human and an object context-attention module. The final prediction is with the sum of the object predictions and the contextual streams. To evaluate, they applied the Precision (mAP) metric to the detection task and intersection-over-union (IoU) between the humans and objects with a threshold of 0.5 or higher for the action prediction. With this proposal, they achieved slightly better results than to the works that were compared.

In (CHAO et al., 2015), Chao et al. introduced a benchmark for recognizing human-objects interactions. In this works, they explored the use of semantic features for the HOI recognition, as well as proposed a new dataset. The dataset is composed of manually annotated pictures obtained from Flickr. Then, they explored a few approaches to identify the correct labels of the images. They applied some existing methods (Random Forest(YAO; KHOSLA; FEI-FEI, 2011), Fisher Vectors (SÁNCHEZ et al., 2013), Deep CNN (KRIZHEVSKY; SUTSKEVER; HINTON, 2012)) and a proposed Human-Object CNN. The last one is composed of object detection and Human Pose estimation, where both results are feed into a CNN to classify the HOI categories. To evaluate, they trained the proposes on the MS-COCO dataset and got the mAP of all proposed architectures compared. The best approach was the DNN, which was composed of features from Alex's Net and an SVM to select the categories. In the end, they compared different types of DNN feeding additional semantic information to the network and having better recognition results, especially for uncommon categories.

Hu et al. (HU et al., 2019) also applied semantic information, proposing an indoor action recognition method using Kinect data and exploring the semantic information on the scene. They introduced a trajectory clustering algorithm to combine the people's characteristics such as spatial localization, movement direction, and speed. Based on the algorithm result, they can extract an area of interest and combine it to color, depth data, motion history, and semantic context of the scene they feed an SVM to recognize the human action. To experiment with the idea, they used images from their own environment and recorded six types of actions. The semantic features are two different mapped areas. In this experiment, they were able to recognize the actions in 93.3% of the time. A second experiment was done in the HuDaAct dataset (NI; WANG; MOULIN, 2011) with 12 indoor actions. In this case, the accuracy rate was 66.17%. In their final results, they conclude that the use of semantic features was superior to methods that only use motion features to recognize the actions.

Meng et al. (MENG et al., 2015) explored the distance between objects and skeleton joints to recognize human-object interaction. To recognize the skeletons, they applied the same method as Shotton et al. (SHOTTON et al., 2011) that can predict the position of body joints using just a depth map without temporal information. The object detection is made using an adopted LOP algorithm (YU; LIU; YUAN, 2014). Having the Joints and the object position, they calculate a feature vector between then, feeding a Random Forest using 50 trees to have the classification. To evaluate the solution, they applied the approach to the ORGBD dataset (YU; LIU; YUAN, 2014). This dataset contains video records of seven types of actions between humans and objects. With this approach, they were able to achieve a recognition rate of 75.8% using cross-validation. This study concluded that the distance between the object and the person's joints is more relevant than the object position.

Likewise, combining body information with objects information Yan et al. (YAN; GAO; LIU, 2019) proposed an approach combining body motion, hand motion, digital gloves, and object recognition. They applied an RGBD camera to get the general movements of the body and the images to recognize the objects, digital gloves are responsible for giving refined hand movements and they used YOLOv3 (REDMON; FARHADI, 2018) in order to detect the scene objects. All the data is used as input to a Deep Neural Network with Softmax responsible for giving the correct interaction label. To experiment with the idea, they implemented their own dataset with eight actions and six objects. They started with the information about the Gloves and Skeleton and improved by adding the object's information. They struggle to synchronize the data as they came from different sources but were able to recognize the actions with 93% of accuracy.

Applying YOLO to get the regions of interest, Xin et al. (XIN et al., 2017) argue that frameworks based on skeleton data can easily make mistakes since similar body positions can be found in very different actions. So they proposed an architecture based on YOLO that identifies two boxes in still images. The A-box is responsible for finding the region that contains human bodies while the C-box is accountable for finding essential objects. To make the recognition, they applied two loss functions based on the squared distances between the coordinates of the objects. An LSTM gives the final classification. The evaluation of the framework they tested on the PASCAL VOC Action Dataset (EVERINGHAM et al., 2010) has ten actions with the COCO dataset (LIN et al., 2014) they applied the framework as an embedding generator, without the detection layer. They achieved good results with the PASCAL VOC (90.6 mAP) as well in the COCO dataset, comparable to the state-of-the-art.

Zhu et al. (ZHU; HU; ZHENG, 2018) also proposed a framework to apply action recognition in still images. They argue that since still images lack movement information, it is a more challenging task. The framework separates the human body into many parts and models the interaction between those parts with the objects. Instead of combining the cues by concatenation, they proposed a hierarchical propagation model from the smaller pieces to bigger pieces, dividing the human body into three levels and propagating it in a bottom-up way. Then they trained a ResNet-50 (HE et al., 2016) for each body part with the corresponding patches. All the networks are pre-trained on ImageNet (DENG et al., 2009) and then fine-tuned for each piece. An SGD is applied to update the model parameters of the propagation network. The team tested the model on HICO (CHAO et al., 2018) that has 600 action classes with 29.3mAP and PASCAL VOC Action dataset (EVERINGHAM et al., 2010) that has ten different actions with 86.4mAP.

Shen et al. (SHEN et al., 2018) introduced a model for HOI detection to categories with distributions far from the head, called long-tail distributions. Their model applies a zero-shot learning approach. In zero-shot learning, the identification of previously unseen classes is performed by information acquired from other classes' training data. To tackle the problem, they introduced a model consisting of both shared neural network layers as well as independent verb and object networks. The model is trained simultaneously in a multi-task fashion but produces disengaged verb and object networks that can be used at test moments to identify new verb-object pairs based on previously seen occurrences of the verb or object. The full joint model is trained end-to-end to produce predicted bounding boxes and scores for all verb and all object classes. During the test, they calculate scores for all blends of verb-object prediction pairs to produce the final HOI prediction, where the verb and object are tightly localized. To test the solution, they experimented on the HICO dataset (CHAO et al., 2018), where they were able to achieve 5.63% mAP.

Applying Action Recognition to the context of agriculture, Vasconez et al. (VASCONEZ; SALVO; AUAT, 2018) proposed a model to detect objects that can provide semantic information for human action recognition. Based on the recognized items in the image, they estimate the action. To do this, they applied a Single Shot MultiBox Detector (SSD) as the meta-architecture and MobileNet as a feature extractor. They trained the model to recognize nine different objects in 2D still images of the avocado harvesting process. They argue that the MobileNet feature extractor model was adopted because it is faster than Inception V2. That supports obtaining faster computing to be implemented in a robot or a cell-phone in future works. They tested the solution with their dataset and achieved 41% to 80% depending on the task. As future work, they pretend to apply their action detection model with semantic representations in a co-robot for future human-robot applications in agriculture.

Monteiro et al. (MONTEIRO et al., 2018) proposed a two-stream architecture, where two convolutional neural networks run in parallel, that considers movements and context scene to recognize actions. They used two VGG-16 networks (SIMONYAN; ZISSERMAN, 2014), one for scene stream and the second one for the action. The first one was trained using weights of a CNN pre-trained using the Places365 (ZHOU et al., 2017) dataset, which contains only images with scenes. For the action, they used pre-trained weights on the ImageNet dataset (ZHOU et al., 2017). In the end, an SVM concatenate both streams for the final classification. They evaluated the solution in two datasets. The DogCentric Activity contains ten different actions performed by four dogs and the UCF YouTube Action containing 11 actions. They were able to achieve 53% and 75% of accuracy, respectively. As future work, they plan to change the action stream to recognize actions taking into account the video's temporal aspect.

Zhu et al. (ZHU et al., 2018a) also applied the YOLO object detector to help in action recognition. They propose a framework for action recognition in untrimmed videos with two detectors. The first one is responsible for sequentially regresses candidate bounding boxes using Recurrent Neural Network for long-term temporal contexts. The second is based on YOLO and produces bounding boxes using cues in every single frame. Once the boxes where detected, they apply a path and trimming process. It consists of verifying an action score, confidence score, and background score. Based on the scores, they can find the boxes with where is happening the action. To experiment with the idea, they trained the model on ImageNet, and then they fuse the corresponding outputs of the detectors in an LSTM to obtain the final action proposals. They then tested on UCF-101, UCF-Sports, and JHMDB datasets, looking for the Intersection over Union (IoU) from the ground truth to the recognized box. With this framework, they were able to get impressive results on the tested datasets.

3.3 Discussion

In section 3.1, we proposed our study design as a systematic literature review. This approach supports us to verify whether research pieces of evidence exist on the studied topic and present quantitative evidence (BRERETON et al., 2007; BEELMANN, 2006). In this section, we return to the proposed questions where we discuss and seek to answer them. In order to do this, we bring insights and quotes from the revised works and explore the ideas to identify the patterns. The table 1 organize the comparison between the selected works.

Table 1 – Related works

Article	Main Application	Architecture	Evaluated Datasets (%)	Input type	Frames /s
(NIE et al., 2018)	Chield Caring Robot	Euclidean Distance Matrix, ResNet	Northwester-UCLA (81.80%)	RGB, RGBD, Skeleton Data	0.5 f/s
(JACOBY et al., 2018)	Learning Environments	KNN, CNN	Self Made (95)	RGB Videos	Offline
(WANG et al., 2019)	General	FPN, CNN	V-COCO, HICO-DET, HCVRD	RGB	Offline
(CHAO et al., 2015)	Dataset publishing, General	Deep CNN	HICO (49.98)	RGB	Offline
(HU et al., 2019)	Indoor security	Motion Trajectories, SVM	Self Made (93.33)	RGBD	Offline
(MENG et al., 2015)	General	RF, LOP	ORGBD (75.80)	RGBD Videos	Offline
(YAN; GAO; LIU, 2019)	General	CNN	Self Made (93)	RGBD, Sensors	Offline
(XIN et al., 2017)	General	CNN, LSTM	PASCAL VOC (90.60)	RGB	Offline
(ZHU; HU; ZHENG, 2018)	General	CNN, SGB	HICO, PASCAL VOC(86.40)	RGB	Offline
(SHEN et al., 2018)	General	CNN	HICO-DET (7.12mAP)	RGB	Offline
(VASCONEZ; SALVO; AUAT, 2018)	Agriculture	CNN, SSD	Self Made (86)	RGB	Offline
(MONTEIRO et al., 2018)	General	CNN, SVM	UCF YouTube (75), DogCentric (75)	RGB Videos	Offline
(ZHU et al., 2018a)	General	CNN, LSTM	UCF-101, UCF-Sports, JHMDB (99.31 recall)	RGB Video, Flow	Offline

Source: Elaborated by the author.

3.3.1 What are the current action recognition strategies?

Strategies, in general, differ significantly, as there are several ways to input and get features to recognize actions. Hu et al. Hu et al. (HU et al., 2019) introduced a trajectory clustering algorithm to combining the people characteristics such as spatial localization, movement direction, and speed. (NIE et al., 2018) Explored the use of skeleton data with object recognition cues to deal with the problem as well as (CHAO et al., 2015) that combined pose with semantic features and feed into a CNN to classify the HOI categories. To recognize actions, similar to what is proposed in this work, (MENG et al., 2015) used the distance between objects and skeleton joints.

Following the line of recognizing objects, Jacoby et al. (JACOBY et al., 2018) explored the color Object recognition with context-based rules to solve the HAR task. The most similar propose to this work (XIN et al., 2017) uses two boxes in still images. The A-box is responsible for finding the region that contains human bodies while the C-box is responsible for finding important objects. Wang et al. (WANG et al., 2019) also, localize the objects and then fuse with scores from a human and an object context-attention modules.

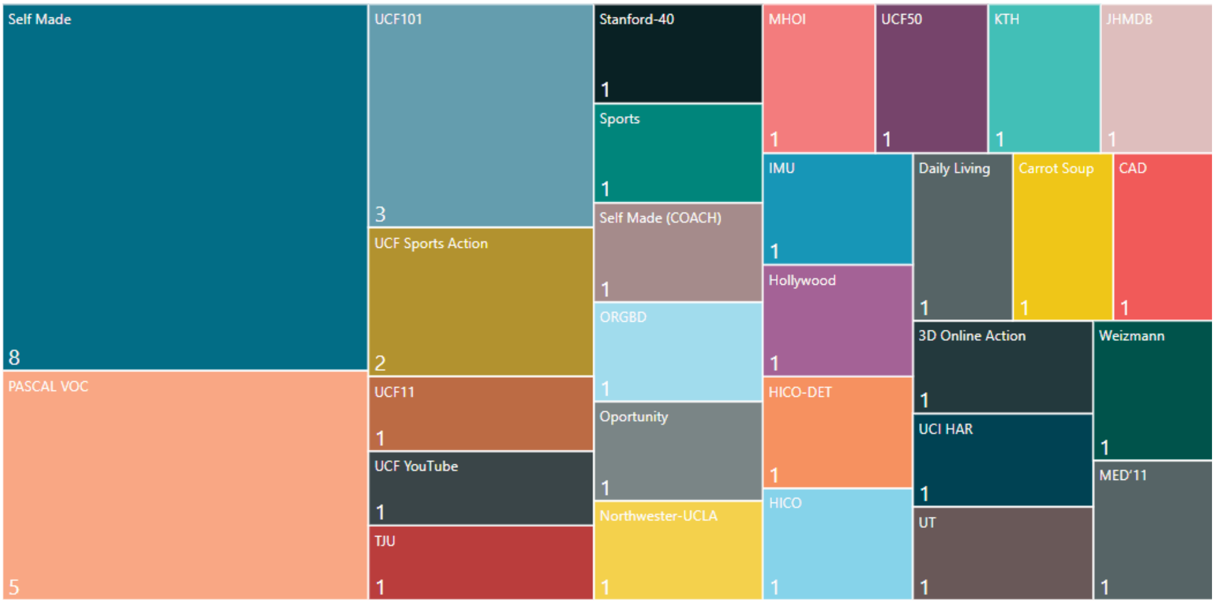
3.3.2 In what context is action recognition being applied?

Most of the works (JACOBY et al., 2018) (WANG et al., 2019) (CHAO et al., 2015) (MENG et al., 2015) uses general datasets without applying in specific scenario. On the other hand some works applied their research to particular cases, as we can cite child caring robot (NIE et al., 2018), indoor action recognition (HU et al., 2019), agriculture (VASCONEZ; SALVO; AUAT, 2018), dog actions (MONTEIRO et al., 2018), smartphone application (CAO et al., 2018) and learning environment (JACOBY et al., 2018).

3.3.3 What are the major recognition strategies using objects?

To recognize objects, the works explored different features based on the input type. Working with sensors data, (YAN; GAO; LIU, 2019) examined the data with camera data to identify objects. Dealing with RGB images, (WANG et al., 2019) explored the FPN paradigm to recognize objects. With videos, (ZHU et al., 2018a) (XIN et al., 2017) applied YOLO to recognize the objects, as well as (NIE et al., 2018) that applied YOLOv2 and (YAN; GAO; LIU, 2019) used YOLO v3. Similar to YOLO (VASCONEZ; SALVO; AUAT, 2018) used the Single Shot MultiBox Detector to do the task. Also to detect the objects in RGB videos, (JACOBY et al., 2018) used color-based segmentation while (MENG et al., 2015) LOP algorithm.

Figure 8 – Reviewed articles by dataset.



Source: Elaborated by the author.

3.3.4 What are the leading workflows to recognize actions using objects?

The most used approach for works that use object recognition cues to deal with action recognition applied CNN variations as YOLO (ZHU et al., 2018a) (XIN et al., 2017) (NIE et al., 2018) or SSD (VASCONEZ; SALVO; AUAT, 2018) as input and then exploring motion and semantic signals. Similar to this (MONTEIRO et al., 2018) combined two VGG-16 networks, one for scene stream and the second one for the action. But there are also some variants as (WANG et al., 2019) that localize the objects and then fuse scores from a human and an object context-attention modules as well as (JACOBY et al., 2018) that uses object recognition with context-based rules. Skeleton based approaches also were tested combined with object detection cues, as (MENG et al., 2015) that combined with body joints using depth map and (ZHU; HU; ZHENG, 2018) that combined with a model that divided the human body into parts.

3.3.5 What are the main Datasets applied to evaluate the solutions?

A large number of works apply in their own dataset. This is due to the exploration of work-specific cues, which is why researchers end up applying them in a closed dataset. But this is a problem as there is no way to compare solutions because datasets are often unpublished or too limited, which discourages further research on this data. As we look at the datasets used, as can be seen in Figure 8, it is evident that there is a wide range of different datasets being used, but the vast majority have many limitations. Featured in the analysis are the PASCAL VOC (EVERINGHAM et al., 2010), UCF101 (SOOMRO; ZAMIR; SHAH, 2012) and UFC Sports (RODRIGUEZ; AHMED; SHAH, 2008) datasets.

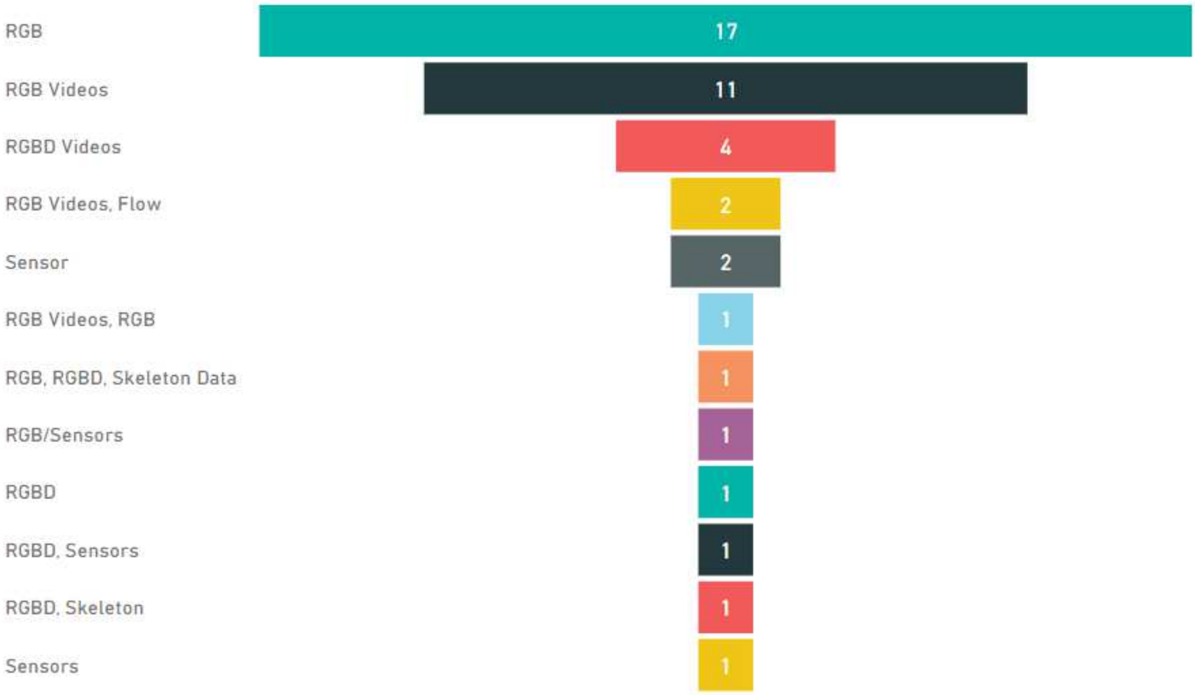
3.3.6 What problems are still open?

Many important milestones have been achieved over the last decade, but hard challenges remain. The complexity of human nature allows for the same action to be executed as different pose sequences (SHAHROUDY et al., 2015). Additionally, there is intra-class similarity of motion and appearance among actions, e. g. "waving"and "reaching for"may differ only at the very end of their pose sequences (DAS et al., 2019) (LI; TONG; TANG, 2018). In this way, without clear boundaries between action in a continuous video stream, activity recognition is a considerably harder task and is currently unsolved. This may be due to the limitations of current publicly available datasets, which are used for training the models.

The classes in the real-world also remains a challenging problem due to large intra-class variation (LI; TONG; TANG, 2018). Although there are many publicly available datasets, the span of available action classes is still constrained. All of the datasets are composed of a limited set of labeled actions and well-defined collections of frames per action instance. Having a restricted set of action classes makes the models biased towards such actions, reducing their generalization, i.e., models tend to overfit to the given scenes and correlate actions to specific objects (ZHU et al., 2018b). Even variety on the types of objects and the environment interfere with model performance on unseen scenarios. On the other hand, handling all these imitations results in the unfeasible task of mapping the space of all possible actions for unconstrained videos (ZHU et al., 2018b).

Although many HAR methods have achieved excellent performance, they are still far from being able to express practical visual information for efficient high-level interpretation (LI et al., 2017). Recognizing actions from untrimmed sequences is also a new field for exploring (LIU; LIU; CHEN, 2017). In contrast to RNN-based approaches, how to adequately represent the 3D skeleton data and feed it into deep CNN models is still an open and critical problem (LI et al., 2018). Also, another problem can be seen by observing the collected data. From all works reviewed, just four (NIE et al., 2018; ZHU; VIAL; LU, 2017; JEAN-BAPTISTE; MIHAILIDIS, 2017; FLORES-VÁZQUEZ; ARANDA, 2016) have online approaches, while only one has resulted in more than 5fps (ZHU; VIAL; LU, 2017). Also, most of the reviewed works still operate with RGB data, without motion cues, as seen in Figure 9.

Figure 9 – Reviewed articles by input type.



Source: Elaborated by the author.

4 PROPOSAL: THE MODEL BRAIN ACTION

This chapter aims to describe the Brain Action model, an action recognition method based on what we know about how the human brain recognizes objects. This chapter is divided into three sections. The first will detail how the human brain recognizes actions, how other works accomplish HAR using objects, and the action recognition problem that motivated this approach. Then the next section presents the model proposal, and the third section offers the proposed prototype to implement the model. The last section introduces the metrics to evaluate the solution.

4.1 Model Foundation

The comprehension of how the brain makes action recognition is the first move to approach a complete understanding of human intelligence and conduct the evolution of better AI systems (ISIK; TACCHETTI; POGGIO, 2017). Here we will seek to understand how the human brain recognizes objects and how other works deal with action recognition problems to gather knowledge to propose the model.

4.1.1 How the Human Brain Recognize Objects and Actions?

Understanding words in a book, some keys on a desk, or a person entering a car seem all so easy. The apparent simplicity of our visual perception abilities refutes the computational dimension of this achievement. People can easily detect and classify objects from huge different possibilities (BIEDERMAN, 1987), and we can do so in less than a fraction of a second (POTTER, 1976; THORPE; FIZE; MARLOT, 1996), even with the massive disparity in appearance that each object can produce on our eyes (LOGOTHETIS; SHEINBERG, 1996). From an evolutionary viewpoint, our recognition techniques are not unexpected. Our daily activities (reading, interacting socially, looking for food, etc.), and consequently, our lives depend on our reliable and fast extraction of object identification from photons' patterns on our retina (DICARLO; ZOCCOLAN; RUST, 2012).

Some experiments in this research area present signs of the current status of how people do object recognition. Taylor (TAYLOR et al., 2014) examines the construction of contour detection and combination in individuals, correlating the functional trajectory from infancy to adolescence to the expanding range of horizontal connectivity within regions V1 and V2 through the same age. Kosilo et al. (KOSILO et al., 2013) then report new tests intended to disconnect the consequences of color and contrast to the ability to identify object features on the control of the eye's rapid movements between fix points (FIELDS, 2016).

For more than a century, the question about how object detection and action detection is made and a general view has started to appear just in the past two decades (FIELDS, 2016).

Schneider's (SCHNEIDER, 1969) suggests that two different paths support visual orientation toward object features and locations was a watershed event in this increasing perception (GOODALE; MILNER, 1992). Research beginning from this approach has connected object recognition to the experiences of space, time, and persistence over time (SCHOLL, 2007; FIELDS, 2012). Outwardly a spacetime container and individual time-persistent objects, motion causation cannot be determined. Consequently, object recognition also carries these experiences (FIELDS, 2016).

Even though the human brain is capable of recognizing objects in 200 ms (ISIK; TACCHETTI; POGGIO, 2017), there are pieces of evidence that half of the non-human primate neocortex is dedicated to visual processing (FELLEMAN; VAN, 1991). This fact addresses the complexity of object recognition, showing that object recognition is a hard task also to the brain. The speed of action recognition that humans have reveals to us that the brain's algorithm is relatively simple and most possible processes the optical information just once. This evidence means that there is no requirement to memorize the video to recognize the action or process it on various occasions. (ISIK; TACCHETTI; POGGIO, 2017).

The brain signs can globalize over the various views. In addition to recognizing different actions quickly, the brain also does in a way that can disconnect differences in perspective. In the time that the brain recognizes actions, it can ignore differences in viewpoint. It is impressive considering that discrimination and generalization are conflicting goals but happens at the same time (ISIK; TACCHETTI; POGGIO, 2017). This finding suggests implementing discrimination and generalization at the same stage in a computer algorithm (ISIK; TACCHETTI; POGGIO, 2017).

The brain distinguishes among several actions, and when it does so in a method that is invariant to changes in 3D viewpoint (ISIK; TACCHETTI; POGGIO, 2017). Though, if either form or motion knowledge is excluded from the stimulus collection, experiments observe a reduction and delay in invariant action decoding. Investigations propose that the brain identifies actions and makes invariance to complex transformations simultaneously and that both form and motion knowledge are crucial for fast, invariant action recognition (ISIK; TACCHETTI; POGGIO, 2017).

4.1.2 How Other Works Deals With Action Recognition?

A thorough review of the related works was conducted on 2, where we compare the methods and highlight their differences and similarities. Our focus is to understand the different approaches used to provide action recognition as well as problems related to other techniques that motivated and sustain the propose.

There are sensor-based approaches and video-based approaches (HUSSAIN; SHENG; ZHANG, 2019). According to Wang et al. (WANG; ZHOU, 2015) sensor-based action recognition can be classified into three principal categories, based on the type of sensor's deployment: Wearable (wearing clothes with sensors), object-tagged (sensors into objects), and dense sensing(sensors into the environment). But sensor approaches are invasive or need at least a previously prepared environment, and we will not discuss here. Since the human brain mainly uses images to process the recognition of actions, we are more interested in video approaches.

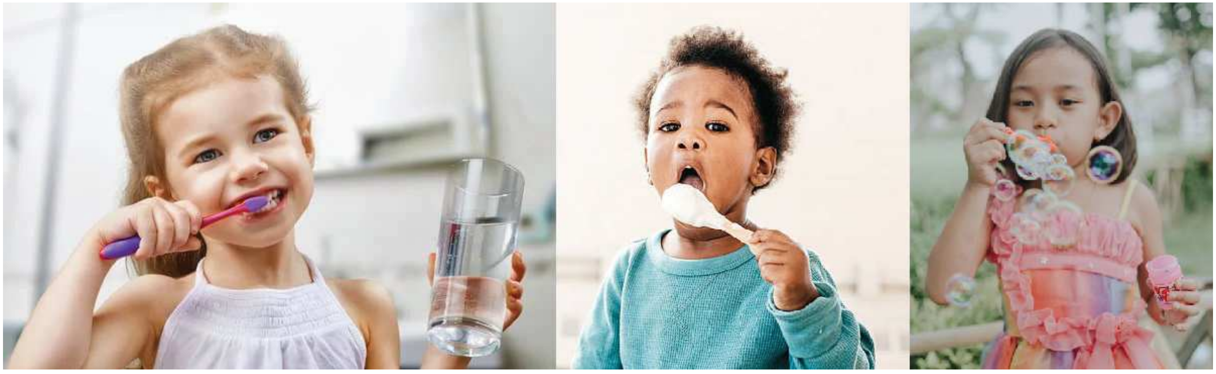
On video approaches, we have procedures that use pose information and strategies that use semantic information. Regarding pose approaches, they could use RGB, RGBD, or Skeleton Data, but they all seek to get the pose positions and movements to infer action. With RGB data, there are two strategies: Works that use static images, and works that use video streaming or video frames as input. Techniques that use static images (LUVIZON; PICARD; TABIA, 2018) as information has an overall performance worse than those that use video(LIU; YUAN, 2018) since they do not have the time constraints and the skeleton movement and miss the Spatio-temporal correlation.

Depth sequences were the second generation of HAR strategies. They combine the RGB features with Depth information of the scene. With this new information, algorithms were capable of distinguishing better the classes and achieve better results than just with RGB data. However, this approach is being replaced in the last years by the Skeleton Data, since its information is more robust. In the studied works, we could observe the use of depth data to extract body points (LIU; YUAN, 2018; SHAHROUDY et al., 2015) to get the feature descriptors combined with Color Maps (LIU; YUAN, 2018), Regression Framework (SHAHROUDY et al., 2015), and CNN (LIU; YUAN, 2018).

Skeleton strategy is the new trend in the HAR area. Along with the studied works, we can notice a division by three different input types: Skeleton Pose (DAS et al., 2019; LI et al., 2018), that uses on pose estimation strategy, and then apply the action recognition algorithm; Skeleton Joints (LI et al., 2017), that receives the points of specific skeleton joints to apply the classification task to action recognition; and Skeleton sequences(PHAM et al., 2018, 2019; KE et al., 2018; LIU et al., 2017b; JIANG; XU; SUN, 2019; LIU; LIU; CHEN, 2017), that combines the Skeleton Poses or Joints into a collection, so the algorithm has to deal with multiple inputs to one classification task.

Pose-based action recognition is unaffected by a complex foreground and background. However, the pose keeps only a small amount of information, and its representative ability is limited to (LI; LIU; ZHANG, 2019). When dealing with single person action recognition and limited actions, we can see great achievements with pose strategy for action recognition (PHAM et al., 2019, 2018; SHAHROUDY et al., 2015; ZHU et al., 2018b; DAS et al., 2019; LIU; YUAN, 2018; LIU et al., 2017b; KE et al., 2018; JIANG; XU; SUN, 2019; LI et al., 2018, 2017; LIU; LIU; CHEN, 2017; LUVIZON; PICARD; TABIA, 2018; LI; TONG; TANG, 2018). However, when dealing with multi-person scenes with occlusions, as well when dealing with similar poses to different actions, similar poses can be associated with different action (YAN et al., 2017; LI; LIU; ZHANG, 2019; BARADEL; WOLF; MILLE, 2017), as we can see in the figure 10. It's also important to highlight that tests on small datasets with few actions and few

Figure 10 – How to deal with similar poses that can be associated with different actions? Merely the pose is not sufficient to determine an action.



Source: Elaborated by the author with pictures from unsplash.com.

videos, make the algorithms be suspected of overfitting (BARADEL; WOLF; MILLE, 2017).

Other works already stated this problem. Li et al. (LI; LIU; ZHANG, 2019) did experiments and concluded in their tests that image-based representations show superior performance, and those image-related representations are more powerful to depict human actions. Liu et al. (LIU; KUIPERS; SAVARESE, 2011) investigated the use of using high-level semantic concepts, also called attributes, to symbolize human activities from videos and demonstrate that characteristics facilitate the production of more detailed models for human action recognition. They claim that an action attribute-based description is further expressive and discriminative for action recognition than conventional methods. They further confirm that attribute-based action representation can be efficiently used to create a recognition system for classifying new classes for which no training examples are available.

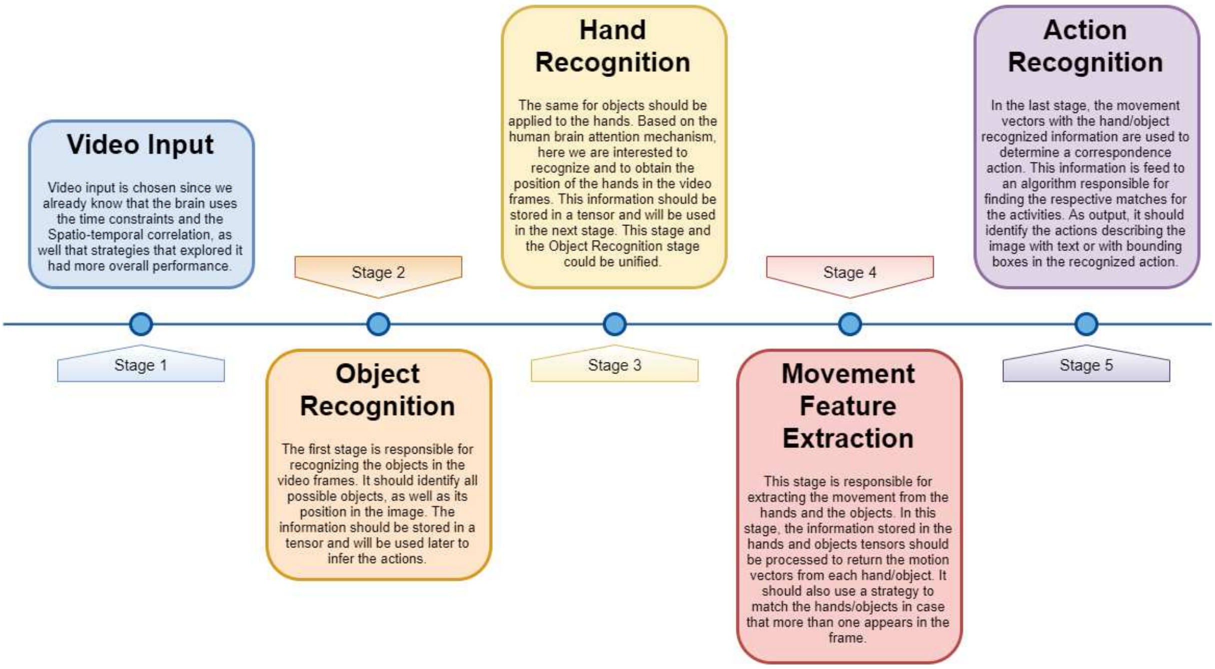
4.2 Model Proposal

After reviewing related works and other works that deal with the action recognition task, we can infer that even though many researchers proposed solutions to action recognition, it is still an open problem. Here we want to explore the same known approach used by the human brain to solve the action recognition problem.

4.2.1 Design Principles

The model architecture is divided into stages, as shown in figure 11. In this section, we will describe each step and what constraints it should care when implementing inspired in the human brain.

Figure 11 – In the Brain Action architecture model, we highlight the principal characteristics to create a recognition system inspired by the human brain.



Video Input

Video input is chosen since we already know that the brain uses the time constraints and the Spatio-temporal correlation, as well that strategies that explored it had more overall performance.

Object Recognition

The first stage is responsible for recognizing the objects in the video frames. It should identify all possible objects, as well as its position in the image. The information should be stored in a tensor and will be used later to infer the actions.

Hand Recognition

The same for objects should be applied to the hands. Based on the human brain attention mechanism, here we are interested to recognize and to obtain the position of the hands in the video frames. This information should be stored in a tensor and will be used in the next stage. This stage and the Object Recognition stage could be unified.

Movement Feature Extraction

This stage is responsible for extracting the movement from the hands and the objects. In this stage, the information stored in the hands and objects tensors should be processed to return the motion vectors from each hand/object. It should also use a strategy to match the hands/objects in case that more than one appears in the frame.

Action Recognition

In the last stage, the movement vectors with the hand/object recognized information are used to determine a correspondence action. This information is feed to an algorithm responsible for finding the respective matches for the activities. As output, it should identify the actions describing the image with text or with bounding boxes in the recognized action.

Source: Elaborated by the author.

4.2.2 Video Input

Based on the study of the human brain, when the motion information is removed, there is a decrease and delay in invariant action decoding (ISIK; TACCHETTI; POGGIO, 2017). This evidence suggests that the brain recognizes actions considering motion and Spatio-temporal data. To be able to simulate the human brain, the input used must include movement and Spatio-temporal information. This information determines the discarding of the possibility of using only contextual information, as well as just images.

4.2.3 Object Recognition

It's still unclear how the human brain recognizes objects, but researchers give us clues about how about implement it inspired in the human brain. It should focus on contour detection and explore color and contrast to be able to identify different objects. It should generalize across the different views, ignoring changes in viewpoint. We can recognize the objects independent of the perspective and classify new objects never saw before. And last, it should process the visual input only once.

4.2.4 Hand Recognition

Human observation centers selectively on elements of the scene to obtain knowledge at particular spots and events, as shown by the neuroscience community (JONIDES, 1983). Based on this, Attention mechanisms (ITTI; KOCH; NIEBUR, 1998) were proposed in computer vision to deal with object detection. Combining attention can potentially guide increased overall efficiency, as the system can concentrate on portions of the data, which are most important to the task (BARADEL; WOLF; MILLE, 2017). Motivated by this, this model focuses attention on the hands considering that humans perform most of their activities using their hands.

4.2.5 Movement Feature Extraction

Motion information has great importance in action recognition (ISIK; TACCHETTI; POGGIO, 2017). This information should be extracted from the video frames to make movement recognition more reliable. Here other problems referent to action should be given attention: The motion process could be done in a few frames and for more than some seconds. In this step, the duration of the movement should be processed, and to the next step, just the motion vector and the labels should be given. Keep track of the elements in the image is another challenge. The element could be gently moved in the image, as well as move faster. A strategy to keep track of each element should also be given attention and consider more than one element with the same label moving around the environment.

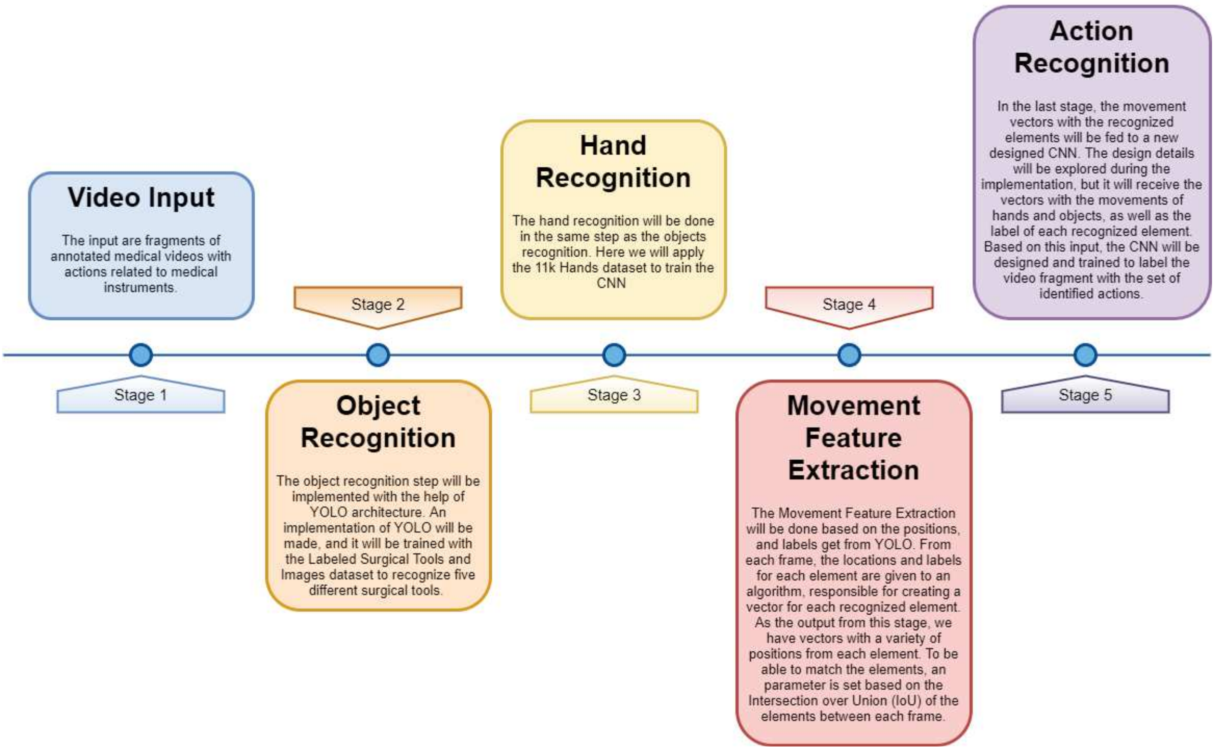
4.2.6 Action recognition

The last stage is responsible for outputting the recognized actions. Here, an algorithm should receive the identified elements and the motion vectors and output the perceived actions. The strategy should be robust enough to understand the known actions correctly, as well as be able to classify never saw actions into known actions based on given previous knowledge.

4.3 Model Implementation Proposal

In this section, we describe our version of the model Brain Action. Here, details of how we pretend to build the model are discussed, and we will present details of our implementation. In the next section, we present some decisions made to build the model that will delimit the scope of the implementation. In section 4.4, we present all details related to the implementation proposal.

Figure 12 – Brain Action implementation proposal. Here we highlight the principal characteristics of our implementation of the Brain Action model.



Video Input

The input are fragments of annotated medical videos with actions related to medical instruments.

Object Recognition

The object recognition step will be implemented with the help of YOLO architecture. An implementation of YOLO will be made, and it will be trained with the Labeled Surgical Tools and Images dataset to recognize five different surgical tools.

Hand Recognition

The hand recognition will be done in the same step as the objects recognition. Here we will apply the 11k Hands dataset to train the CNN

Movement Feature Extraction

The Movement Feature Extraction will be done based on the positions, and labels get from YOLO. From each frame, the locations and labels for each element are given to an algorithm, responsible for creating a vector for each recognized element. As the output from this stage, we have vectors with a variety of positions from each element. To be able to match the elements, an parameter is set based on the Intersection over Union (IoU) of the elements between each frame.

Action Recognition

In the last stage, the movement vectors with the recognized elements will be fed to a new designed CNN. The design details will be explored during the implementation, but it will receive the vectors with the movements of hands and objects, as well as the label of each recognized element. Based on this input, the CNN will be designed and trained to label the video fragment with the set of identified actions.

Source: Elaborated by the author.

4.4 Prototype

In this section, we present our version of the Brain Action model. Here we detail how we will implement each step, and a complete diagram of the system could be visualized on figure 12.

4.4.1 Project Decisions

Some decisions were made in order to align the model constraints with the current necessities and also to be able to have a closed scope. Here, we list all the decisions made.

1. **Surgical environment** - The SOFTWARELAB [1] is developing a project in partnership between Unisinos, Siemens, and FAU, where a system for recognition of actions in the surgical environment is required. As a collaboration between teams, this implementation will focus on the surgical environment.

2. **Open Datasets** - As long as possible, the implementation will use open datasets. This is motivated by two aspects: (a) Public datasets are available to other researchers compare

[1] http://www.unisinos.br/softwarelab/en/

and evaluate between proposed solutions, and (b) annotating dataset requires a considerable amount of structured work and is not the focus on this work.

3. **Annotation Tool** - When necessary annotating videos, a widely used open tool should be applied, as the Cvat [2].

4. **Known solutions** - Wherever it is possible well-known solutions that match the model requirements should be employed to evaluate known solutions as well as improve the overall software quality and productivity.

4.4.2 Video Input

One of the proposes of this study is to aid SOFTWARELAB in their research on surgical rooms. So we will be applying videos from surgery as input on this work. The videos will be annotated with the CAT annotation tool (LENZI; MORETTI; SPRUGNOLI, 2012), mapping the x and y positions of surgical items, as well as their labels. Each action will be a video fragment, and each video will have a label based on the executed activity.

4.4.3 Object Recognition

The first stage of the Brain Action model is object recognition. Attending to the model requirements, we will employ YOLO on this proposal. YOLO is is an end-to-end single convolutional neural network that detects objects based on bounding boxes prediction and class probabilities, and we present more details in the 2.3.1 section.

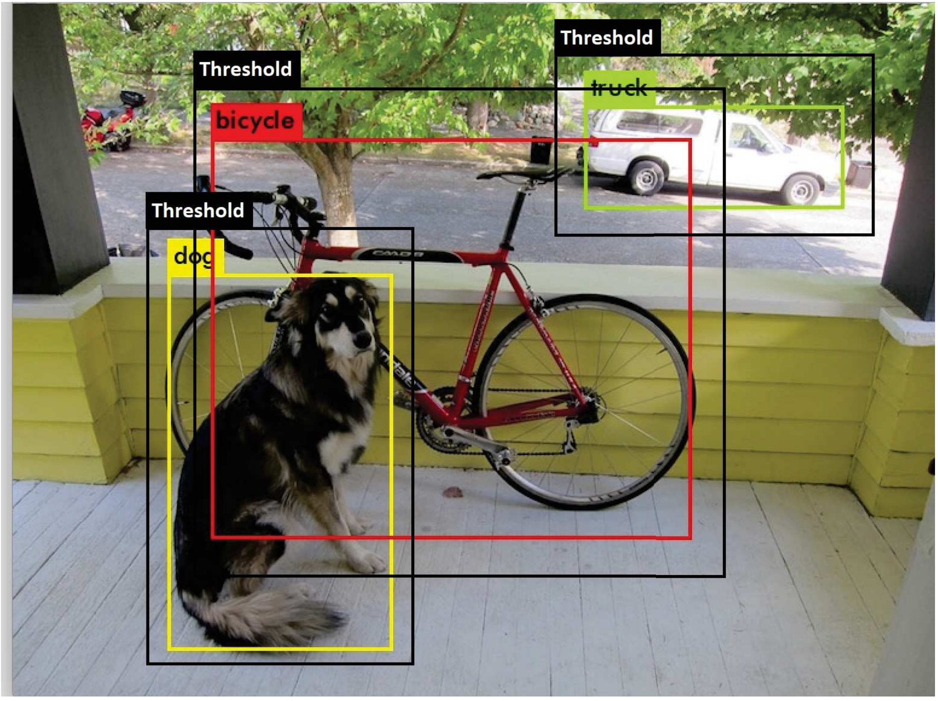
Will be applied a full YOLO implementation on a widely-used programming language to return the predicted objects and their positions on each frame. This information will be managed by a method responsible for getting the data from each frame and store on and Tensor container that will be the output of this stage. A parameter will be added to define the minimum change registered on the tensor. If the objects didn't change a minimum value, it will not be registered. This is a way to prevent that the time variance on the same type of actions impacts the action recognition in the next stages. The value will be on pixels on X and Y and will be configured during the training phase. On figure 13 we represent how the threshold should work.

During the training phase, the YOLO must receive pictures from medical instruments to be able to recognize them. The images that will be used are from the Surgical Tools and Images [3]. The dataset contains 3009 images and the respective labels classifying the objects as Scalpel, Straight Dissection Clamp, Straight Mayo Scissor, or Curved Mayo Scissor. Those instruments will be the ones that we will be able to recognize the actions on videos.

[2] https://github.com/opencv/cvat

[3] https://www.kaggle.com/dilavado/labeled-surgical-tools

Figure 13 – To prevent that the time variance on the same type of actions impacts the action recognition, we will consider an area where we will not record tiny objects' movements in the next frames.



Source: Elaborated by the author.

4.4.4 Hand Recognition

For hand recognition, the same approach used to object recognition will be used. During the training stage, surgical objects and hands will be given to YOLO to recognize both categories in images. By recognizing hands, as same as objects, we rever as identify the item on the video frame, detaching with a bounding box where the hands are and given a correct label. The surgical hand's environment differs from the available datasets. In surgery, all doctors are using the same gloves. Such a situation could be beneficial to CNN recognize based on RGB color, as well as be challenging to define the bounding boxes. Thinking about this, we selected and examined three public available different datasets. We will start the training with the first one, moving on in case of a lower recognition rate. In the next paragraphs, we introduce them.

The 11k Hands dataset (AFIFI, 2019) is a collection of 11,076 hand images of 190 subjects, with ages varying between 18 and 75 years old. Each subject was asked to open and close his fingers of the right and left hands. Each hand was photographed from dorsal and palmar sides with a uniform white background in the same distance from the camera.

The **EgoHands** (BAMBACH et al., 2015) contains 48 different videos of egocentric interactions with pixel-level ground-truth annotations for 4,800 frames and more than 15,000 hands. It includes 48 Google Glass first-person interactions between two people. It has annotations that make possible distinguish between the observer's hands and someone else's hands, as well as left and right hands.

In Mittal et al. (MITTAL; ZISSERMAN; TORR, 2011) hand images collected from 6 different public image data. A total of 13050 hand instances are annotated. In the collected data, no restriction was imposed on the pose or visibility of people, or any constraint imposed on the environment. The annotations consist of a bounding rectangle oriented concerning the wrist.

4.4.5 Movement Feature Extraction

The movement feature extraction is the most complex part of the model and one of the most significant contributions. Based on how the human brain works, it extracts the movement information from the recognized items on the video's frames to posterior recognition. To our implementation, we are focusing on the integrity of the movement data while keeping the algorithm as less intricate as possible. Since the videos will be a fragment with an action to simplify the implementation, we propose to extract the movement from each recognized item. The movement extraction is done by keeping track of each object until the end of the video input. In each frame, the position of every detected element is saved. In the next step, the movement vector is extracted.

4.4.6 Objects Match

More than one object with the same label could be recognized in a frame. If this happens, we need to know what object is the same as the frame before. Our propose to match the same items is to calculate the overlap between the recognized bounding boxes. Given a Box1 and a Box2 in different frames, we want to know if they are the same element. So we will determine the area of intersection between the two rectangles, defined by $area(Box\_1 \cup Box\_2)$. If they intersect and have a value inside a specified threshold, we consider them the same object. In the list 4.1, we outline the implementation of the proposed solution.

Listing 4.1 – Python code to calculate boxes intersection

```
Input: box1 and box2 coord
Output: interArea

def intersection_over_boxes(box1, box1):
    # get (x, y) coordinates
    xA = max(box1[0], box2[0])
    yA = max(box1[1], box2[1])
    xB = min(box1[2], box2[2])
    yB = min(box1[3], box2[3])

    # compute the intersection
    intersectionArea = max(0, xB - xA + 1) * max(0, yB - yA + 1)

    return intersectionArea
```

4.4.6.1 Objects Movement Measurement

When the first frame is recognized, the coordinates of the objects are just stored. From the second frame and going on, the object's movement measurement proposed here takes in count in the object match value described above. If the value is beyond a threshold, then we consider that the object didn't move in that frame, not updating its movement values. This is done to check that two same movements that are done at different times are recognized as the same.

If the value is below the threshold, then we calculate the movement vector. We take the center of the previously recognized box and the center of the new recognized box, as given by the equation $((x1 + x2)/2, (y1 + y2)/2)$. With the boxes centers, we calculate the Euclidian distance between the boxes. The Euclidean distance between points p and q is given on equation 4.1.

A variable is responsible for storing the movement vectors from all detected objects. With the new information coming from the next frames, we update the vector and output it to the next stage.

$$d(p, q) = \sqrt{\sum_{i=1} (p_i - q_i)^2} \tag{4.1}$$

4.4.6.2 Output Action Recognition

The output is responsible for recognizing the action. It's defined by a deep CNN that receives the tensor with the object's movement vectors and the object labels. It will be trained based on recognized objects, and the objects' movements to predict the action label. CNN was chosen to be a strategy that could solve the problem. Hyperparameters would be used to test different learning configurations. Also, different layer configurations and different activators should be tests to define the best version. Details of this CNN will be defined in the implementation phase, where different CNN configurations will be tested to the propose.

5 MODEL EVALUATION

The Brain Action model seeks tips on how the human brain works to simulate its behavior to be able to recognize actions. This chapter presents the methodology used to evaluate the Brain Action model. For this, an application was developed to perform stock assessments within a hospital environment during cardiac surgeries. To make this possible, a surgery dataset, an image recognition application, and an action recognition application based on the positioning of the images were developed. In the next sections, the details of how each step was carried out are described.

5.1 Evaluation methodology

In this section, we will define the metrics that should be evaluated and compared with the state-of-the-art to assess the proposal. K-fold cross-validation should be used, and the best output used to define the confusion matrix. The confusion matrix is the matrix where actual and predicted values are displayed and gives information to all other metrics be calculated.

5.1.1 Performance

To evaluate the entire workflow performance, we will assess the time spent to execute the whole framework for each frame in each stage. In other words, we will measure the time spent to recognize the items, evaluate the objects, and calculate the distances and the time spent to evaluate the frame and the time to acknowledge the action.

5.1.2 Accuracy

Accuracy is given by the number of correct predictions divided by the total number of predictions, multiplied by 100 as given on equation 5.1. We want to assess the Accuracy of the action recognitions.

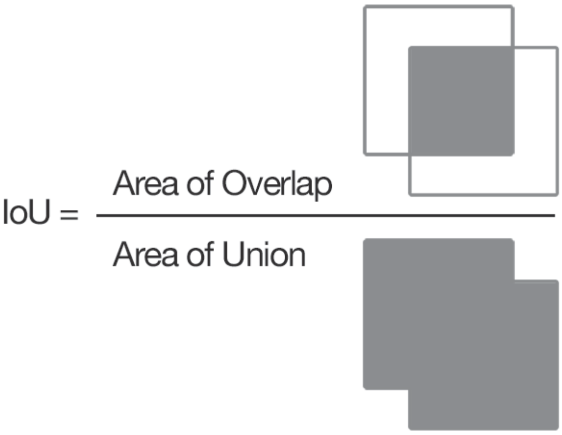
$$Accuracy(\%) = TP/(TP + FP + TN + FN) * 100 \tag{5.1}$$

5.1.3 Precision

The precision is defined as the number of true positives divided by the sum of true positives and false positives, as given on equation 5.2.

$$Precision = TP/(TP + FP) \tag{5.2}$$

Figure 14 – IoU equation



Source: Elaborated by the author.

5.1.4 Recall

The recall is defined as the number of true positives divided by the sum of true positives and false negatives, as given on equation 5.3. Note that the sum is just the number of ground-truths, so there's no requirement to include the false negatives.

$$Recall = TP/(TP + FN) \tag{5.3}$$

5.1.5 Mean average precision (mAP)

When dealing with object detection or action recognition, there are usually K > 1 classes available to recognize. The Mean average precision (mAP) is defined as the mean of AP across all K classes and could be calculated as given on equation 5.4.

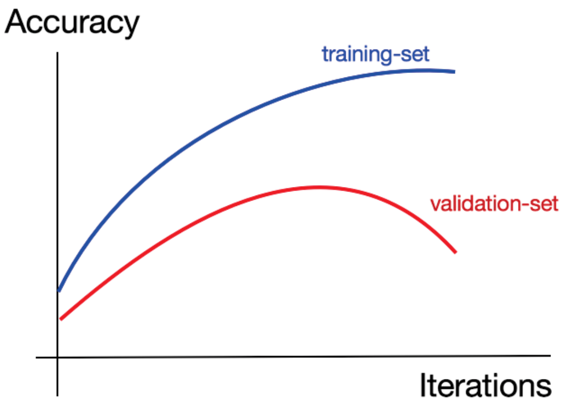
$$mAP = \frac{(\sum K_{i=1}(AP_i)}{K} \tag{5.4}$$

5.1.6 IoU thresholds

Intersection over Union (IoU) seen in figure 14 is an evaluation metric used to measure the accuracy of an object detector on a particular dataset. It's a popular metric across object detection challenges.

The ground-truth bounding boxes, for example, a labeled bounding box of a hand, are compared with the bounding boxes proposed by the object detector. The IoU is defined as the division between the union of the boxes and the intersection of the boxes. In equation 5.5 is defined the method to determine it.

Figure 15 – Overfitting problem



Source: Elaborated by the author.

$$IoU = \frac{area(B_p \cup B_g t)}{area(B_p \cap B_g t)} \tag{5.5}$$

5.1.7 Training strategy

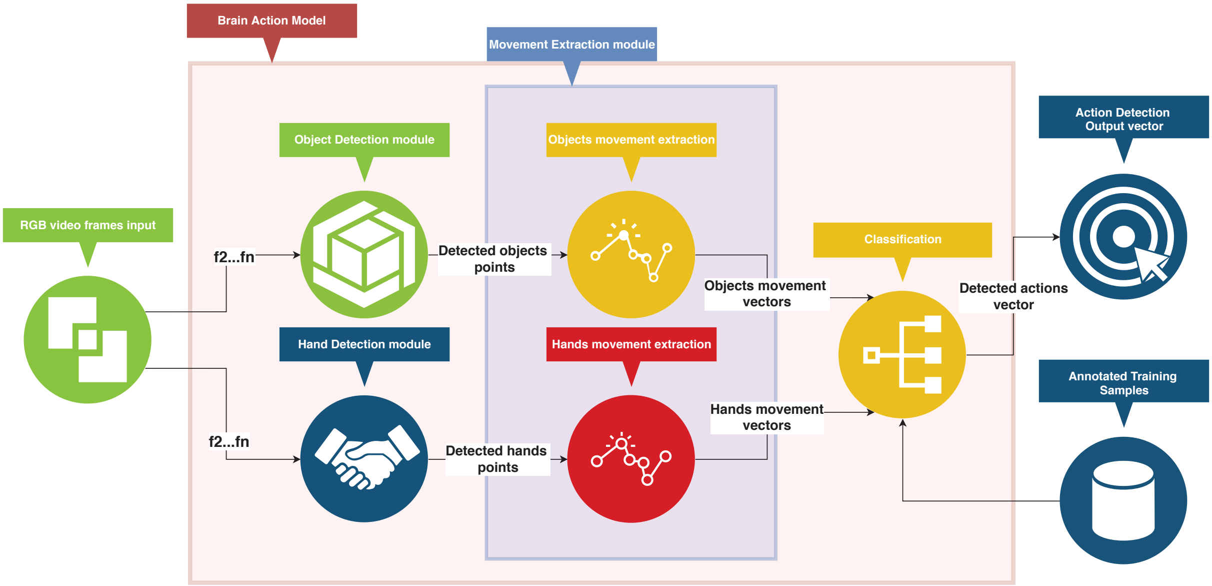
When training a Neural Network (NN), usually the objective is to have the optimal generalization performance. But a common problem when dealing with Neural Networks training is overfitting. While the network appears to get more reliable and the error on the training dataset drops, at a spot while training, it occurs to get worse again, and the mistake in unseen samples increases (PRECHELT, 1998). We draw the situation in figure 15. One widely strategy to avoid this problem is called Early Stop. While training, the model is assessed on a holdout validation dataset after each epoch. If the model's performance on the validation dataset starts to deteriorate, then the training process is stopped (BISHOP, 2013). That is the strategy that will be applied.

How to define the separation between the training dataset and the testing dataset is also essential. K-fold Cross-validation is a way of ensuring robustness in the model, as the expense of computation. It divides the data into k equal parts, uses k-1 parts to training, and then evaluates it. The next execution separates another k part to training, doing this until all data is used to training as well used to test (DANGETI, 2017). In the end, the error will be the average of all errors. To evaluate the solution, at least three different k values will be used, and the one with the lower error will be used.

5.2 Prototype Development

The prototype development was based on the proposed implementation, detailed in section 4.3. Figure 16 represents the developed prototype, and each module will be specified in the sections below.

Figure 16 – The Brain Action prototype architecture overview.



Source: Elaborated by the author.

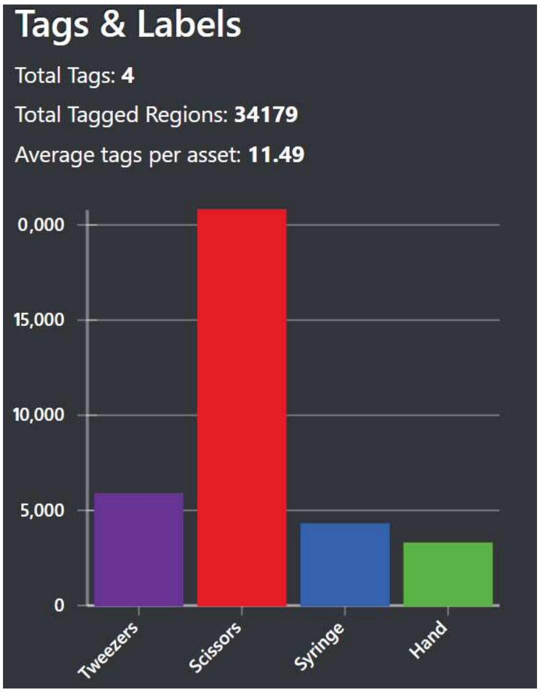
5.2.1 Dataset

The dataset was created based on the information collected in the SoftwareLab operating room workflow project. For its creation, images of five surgeries were used from the same camera position, positioned behind the operating room technician. The frames went through a manual analysis to make it possible to select the frames corresponding to the following actions:

- No action (DoNothing)
- Act of picking up the scissors (TakeScissors)
- Act of taking the tweezers (TakeTweezers)
- Act of taking the syringe (TakeSyringe)
- Act of returning the scissors (ReturnScissors)
- Act of returning the tweezers (ReturnTweezers)
- Act of returning the syringe (ReturnSyringe)

Over 100 thousand frames were reviewed during this process. A total of 3045 frames were selected, annotated with their due actions. For the annotation of the actions, frames referring to the movement performed were selected and cataloged. There is a variation of 7 to 20 frames per share. After the frame selection process, there was a second stage of annotations using the open-source tool VoTT [1] provided by Microsoft. For selected actions, the objects involved

[1] https://github.com/microsoft/VoTT/



Tags & Labels

Total Tags: **4**

Total Tagged Regions: **34179**

Average tags per asset: **11.49**

Figure 17 – Brain Action dataset tags distribution. In general surgery, there is a significant disparity in instrument type in the table, so we have a considerable difference between the number of object classes. This unbalanced situation makes this a challenging dataset to achieve excellent results in all categories.

Source: Elaborated by the author.

(scissors, tweezers, and syringes) were identified and located in each frame, in addition to the hands. All objects in the image are identified, whether they are involved in an action or not. In this process, over 34000 objects were marked and cataloged, and the object distribution can be observed in figure 17.

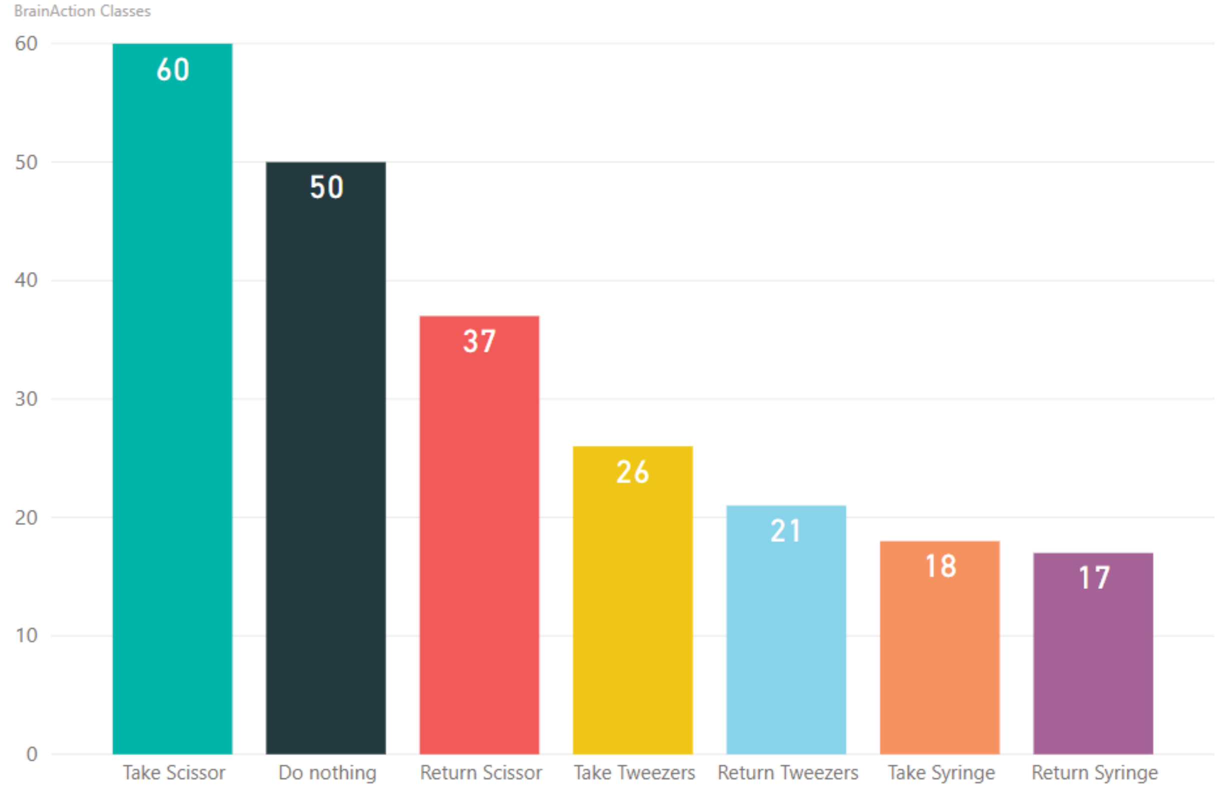
The capture of the images is performed by a Kinect 3 at one frame per second. Each image is 1920x1080 pixels. All images refer to real surgeries, with a variety of doctors involved, number of people, types of surgeries performed, and equipment used. There is also overlapping of people, occlusions, and differences in lighting. For data privacy reasons, this dataset cannot be made publicly available. This is unfortunate as it presents a real-world scenario with unique and very challenging characteristics, making it attractive to other experiments.

As a case of a real-world dataset, it has unbalanced classes. Figure 18 shows the difference in the number of shares for each category.

5.2.2 Brain Action Prototype

This section describes how the BrainAction prototype was developed. Here the design choices are reported and detailed how the application was built.

Figure 18 – The Brain Action dataset has unbalanced quantity in its action classes, a reflection of a real-world dataset.



Source: Elaborated by the author.

5.2.2.1 Image recognition

For image recognition, a Python implementation of YOLO V3 based on TensorFlow [2] and CUDA [3] was developed. This application has two main methods, one for training and a second for object identification. The training method is used to define the neural network parameters based on the objects we want to identify. Once trained with sufficient images, the model can identify the images in new frames, using the identification method.

For the training stage, the frames were divided using the K-fold Cross-Validation strategy with three parts, with two sections being used in training, and the other part was used for network evaluation. In the three possible executions, the values were similar, indicating that the dataset contains enough examples for the network to be able to identify the objects. The final setup had 100 seasons and a loss hit rate of 35.3848%

The training was performed using an Nvidia Tesla P100 video card on a computer with an Intel Xeon @ 2.00GHz CPU with one core. The entire training stage was carried out online through the Collaboratory [4] platform provided by Google. The resulting training weights were then saved to run the software locally. To train the 100 seasons, it took approximately 12 hours of software execution.

When executing the object recognition method, it has an image as input, and its output is a tensor with the position of a bounding box and a tag for each detected object. There is also the possibility of obtaining the image marked with the respective bonding boxes and classes as a result, as shown in figure 19.

The next step is the detection of actions, based on the movement of the detected objects and the movement of the hands detected by the object recognition software. This part is described in the next section.

5.2.2.2 Action recognition V1

For the recognition of actions, a Python application obtains input tensors with the positions of each bounding box for each recognized object in each frame for an action.

The number of frames per action is defined by the dataset, ranging from 7 to 20 frames. The tensors are divided into A_{obj} and A_{hand}.

The first contains all objects recognized in the image with their box positions x, y and their recognized class. The A_{hand} tensor contains all hands recognized, with positions x and y.

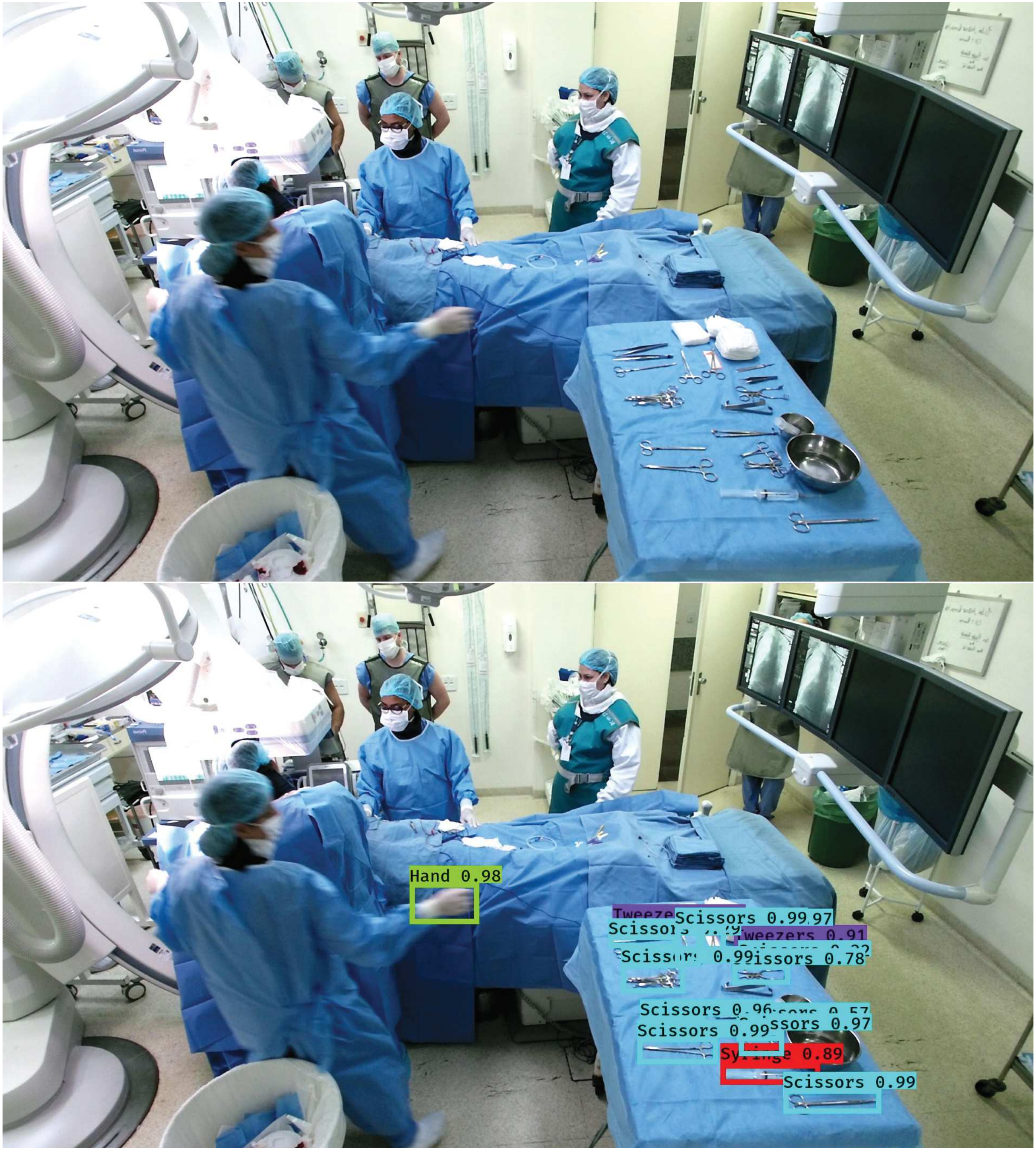
The A_{obj} is defined as $A_{obj,pos} = \begin{pmatrix} obj_1 & pos1_{x,y} & \cdots & pos4_{x,y} \\ obj_2 & pos1_{x,y} & \cdots & pos4_{x,y} \\ \vdots & \vdots & \ddots & \vdots \\ obj_m & pos1_{x,y} & \cdots & pos4_{x,y} \end{pmatrix}$ while the $A_{hand,pos}$ is

[2] https://www.tensorflow.org/

[3] https://developer.nvidia.com/cuda-zone

[4] https://colab.research.google.com/

Figure 19 – The same image before and after being an output of the object recognition step.The object recognition adds the recognized bounding boxes and their classes.



Source: Elaborated by the author.

defined as $A_{hand,pos} = \begin{pmatrix} hand_1 & pos1_{x,y} & \cdots & pos4_{x,y} \\ hand_2 & pos1_{x,y} & \cdots & pos4_{x,y} \\ \vdots & \vdots & \ddots & \vdots \\ hand_m & pos1_{x,y} & \cdots & pos4_{x,y} \end{pmatrix}$.

The first step of the algorithm consists in the match between objects through the intersection calculation. This step is essential to identify the same object between frames. For this calculation, each object is compared with the others through the intersection of union calculation. Once there was a match between the two objects, both are removed from the association list. The intersection metric was defined as greater than 0.7 to identify the correspondence to the same item.

Objects are mapped throughout the frames, and the recognized bounding box position is stored for the movement calculation. At the end of all frames processing, the resulting movement vectors are calculated through Euclidean distance between the objects in different frames. The resulting vector method calculates the object's movement simultaneously between the frames. However, it carries a noise value, resulting from the recognized bounding box position difference in each frame. The output from this phase are two tensors; $B_{obj,pos}$ with moving objects and $B_{hand,pos}$ with moving hands.

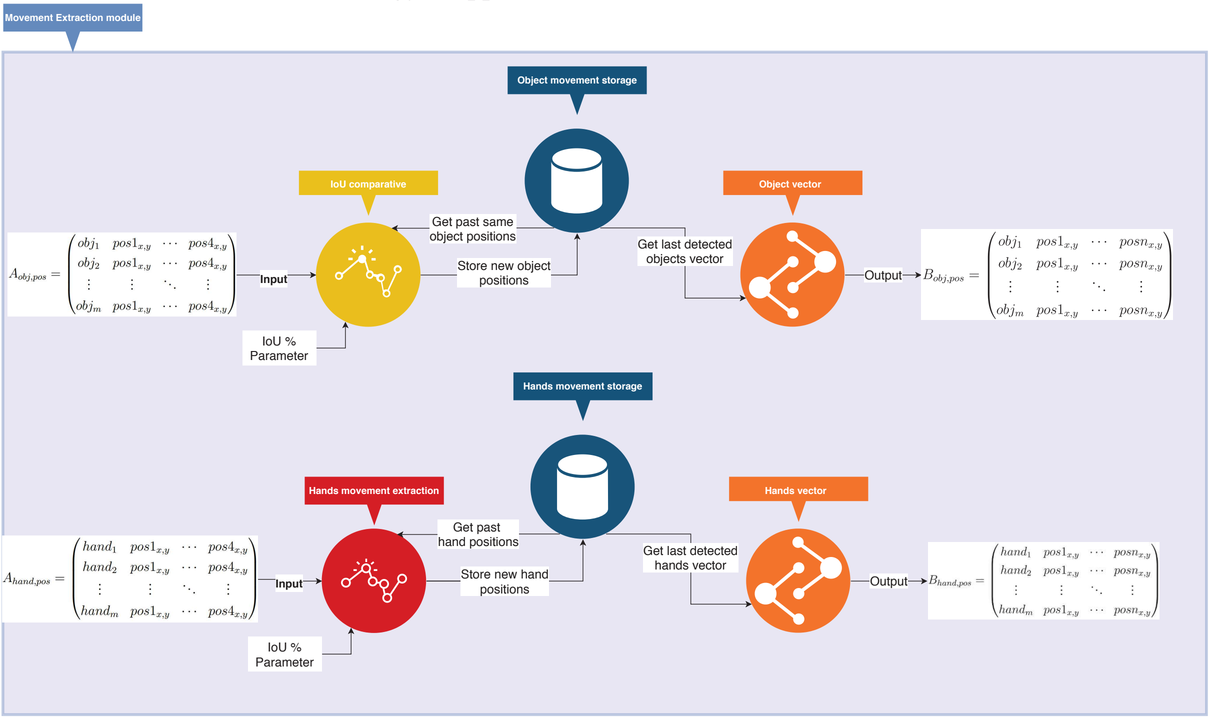
The resulting object movement matrix is defined as $B_{obj,pos} = \begin{pmatrix} obj_1 & pos1_{x,y} & \cdots & posn_{x,y} \\ obj_2 & pos1_{x,y} & \cdots & posn_{x,y} \\ \vdots & \vdots & \ddots & \vdots \\ obj_m & pos1_{x,y} & \cdots & posn_{x,y} \end{pmatrix}$,
while $B_{hand,pos} = \begin{pmatrix} hand_1 & pos1_{x,y} & \cdots & posn_{x,y} \\ hand_2 & pos1_{x,y} & \cdots & posn_{x,y} \\ \vdots & \vdots & \ddots & \vdots \\ hand_m & pos1_{x,y} & \cdots & posn_{x,y} \end{pmatrix}$ defines the resulting hand movement matrix.

In the third phase of action detection, the vectors resulting from the previous step are evaluated by a classification algorithm. Figure 20 represents an workflow of the Extraction Module. The next section will evaluate the Action Detector using different classification methods.

5.3 Preliminary Evaluation

In this section, the results obtained during the model experiments will be presented and discussed. The next sections will be divided by the analyses in each module. In the last section, there is a discussion of the results.

Figure 20 – Movement Extraction module proposal. It receives a tensor with objects and positions, using the intersection of objects between frames to identify its movements. As output, a tensor with objects and their vectors. The same strategy is applied to hands movement detection.


Movement Extraction module
Object movement storage
IoU comparative
Object vector
$A_{obj,pos} = \begin{pmatrix} obj_1 & pos1_{x,y} & \cdots & pos4_{x,y} \\ obj_2 & pos1_{x,y} & \cdots & pos4_{x,y} \\ \vdots & \vdots & \ddots & \vdots \\ obj_m & pos1_{x,y} & \cdots & pos4_{x,y} \end{pmatrix}$
Input
Get past same object positions
Store new object positions
Get last detected objects vector
Output
$B_{obj,pos} = \begin{pmatrix} obj_1 & pos1_{x,y} & \cdots & posn_{x,y} \\ obj_2 & pos1_{x,y} & \cdots & posn_{x,y} \\ \vdots & \vdots & \ddots & \vdots \\ obj_m & pos1_{x,y} & \cdots & posn_{x,y} \end{pmatrix}$
IoU % Parameter
Hands movement storage
Hands movement extraction
Hands vector
$A_{hand,pos} = \begin{pmatrix} hand_1 & pos1_{x,y} & \cdots & pos4_{x,y} \\ hand_2 & pos1_{x,y} & \cdots & pos4_{x,y} \\ \vdots & \vdots & \ddots & \vdots \\ hand_m & pos1_{x,y} & \cdots & pos4_{x,y} \end{pmatrix}$
Input
Get past hand positions
Store new hand positions
Get last detected hands vector
Output
$B_{hand,pos} = \begin{pmatrix} hand_1 & pos1_{x,y} & \cdots & posn_{x,y} \\ hand_2 & pos1_{x,y} & \cdots & posn_{x,y} \\ \vdots & \vdots & \ddots & \vdots \\ hand_m & pos1_{x,y} & \cdots & posn_{x,y} \end{pmatrix}$
IoU % Parameter

Source: Elaborated by the author.

5.3.1 Object Detector

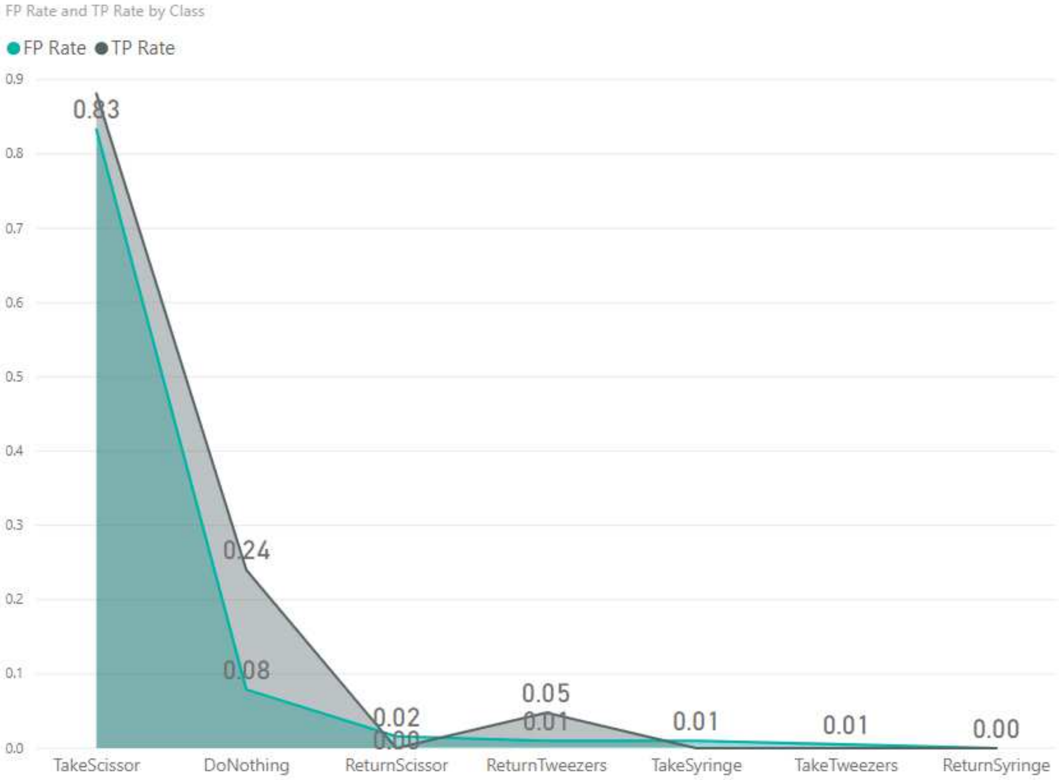
Object detection was trained on a virtual machine, using Google Colab pro, as detailed in section 5.2.2.1. The object identification method was run on a seventh-generation Intel Core I7 home computer with a GTX 270M graphics card. In this environment, the recognition algorithm runs at a rate of 1.5 images per second. The final Mean Average precision (mAP) of the network was 36.4631.

5.3.2 Action Detector

The second and final stage is the action detector. This stage is responsible to based on the transformed data from previous stages, classify into the possible classes. This is a typical classification problem, and it is possible to use different kinds of machine learning algorithms to solve. In this work, we choose to evaluate the Action Detector using three different classification methods: Bayesian Network, Random Forest (RF) and Deep Neural Network. All methods received the vectors with information about the x, y position movements, the classes of the moving objects, and the hands. All test were executed on a seventh-generation Intel Core I7 home computer using only CPU.

For each test, we measured False Positives and True Positives, Precision, and Recall. The

Figure 21 – Bayesian Network False Positive vs True Positive rate.



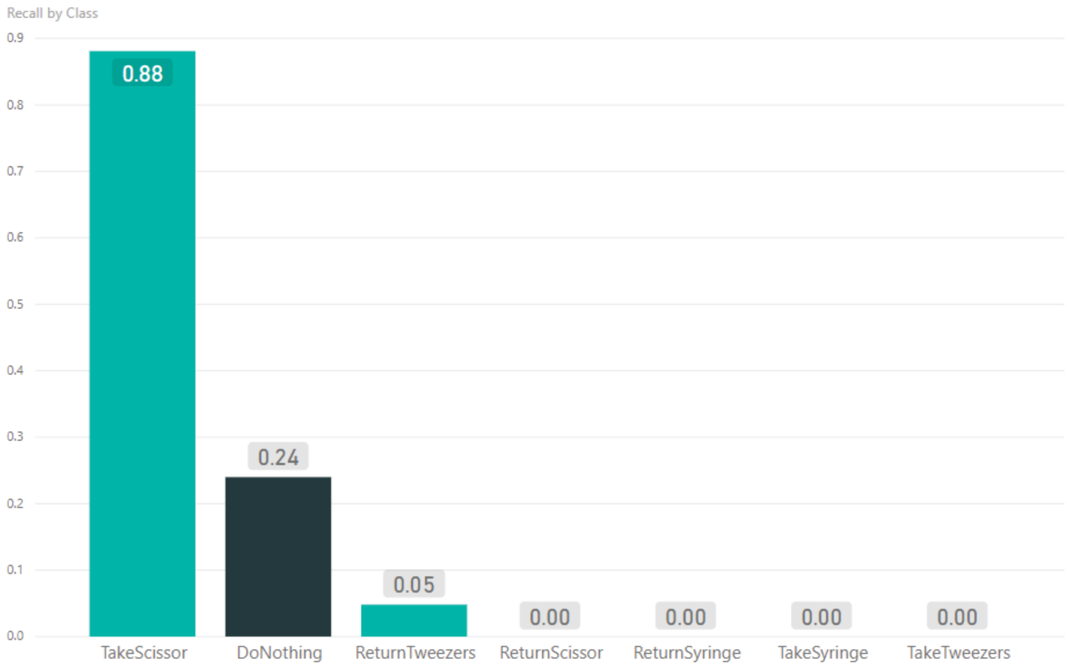
Source: Elaborated by the author.

first measure helps us verify which classes are more correctly detected, which ones are not detected, and verify the behavior. For example, we can have proper overall detection, but detecting only three classes, and having the other classes wrong detected. This graph helps us to analyze the behavior of the detection for all classes. The Recall is an excellent metric to observe how many items were correctly detected. We are also interested in false positives, so we measured Precision since it takes into account the true positives and false positives. Precision is used to understand and how good are our algorithm to classify a particular class. The next sections will detail the tests and discuss them.

5.3.2.1 Bayesian Network

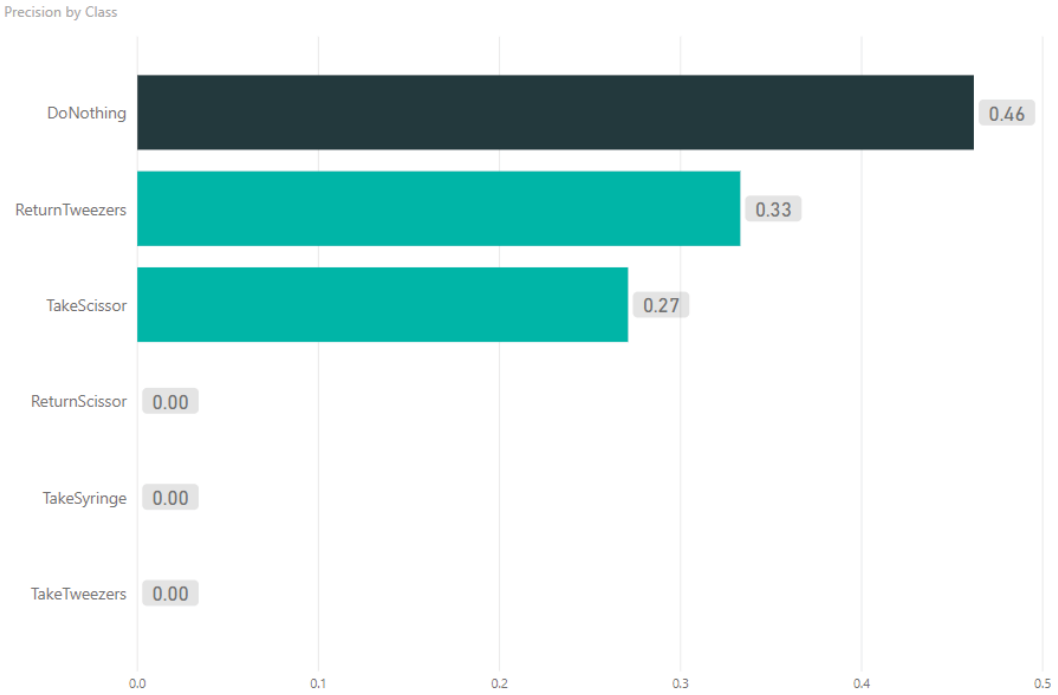
The Bayesian Network applied a simple estimator to construct the probabilities and a hill-climbing algorithm in the learning process. All process was using two decimal places. Figure 21 represents well what happened in the Bayesian test after ten cross-validation folds. It has inferior performance since it tends to classify every item as a Take Scissor label. The class has a good performance of true positives, but also a huge false positives rate. In the other classes, we can also observe inferior performance, and some of them could not be recognized, as Return Syringe and Return Scissor. The overall accuracy was 28.63% and figures 22 and 23 complements the information with Recall and Precision.

Figure 22 – Bayesian Network Recall rate.



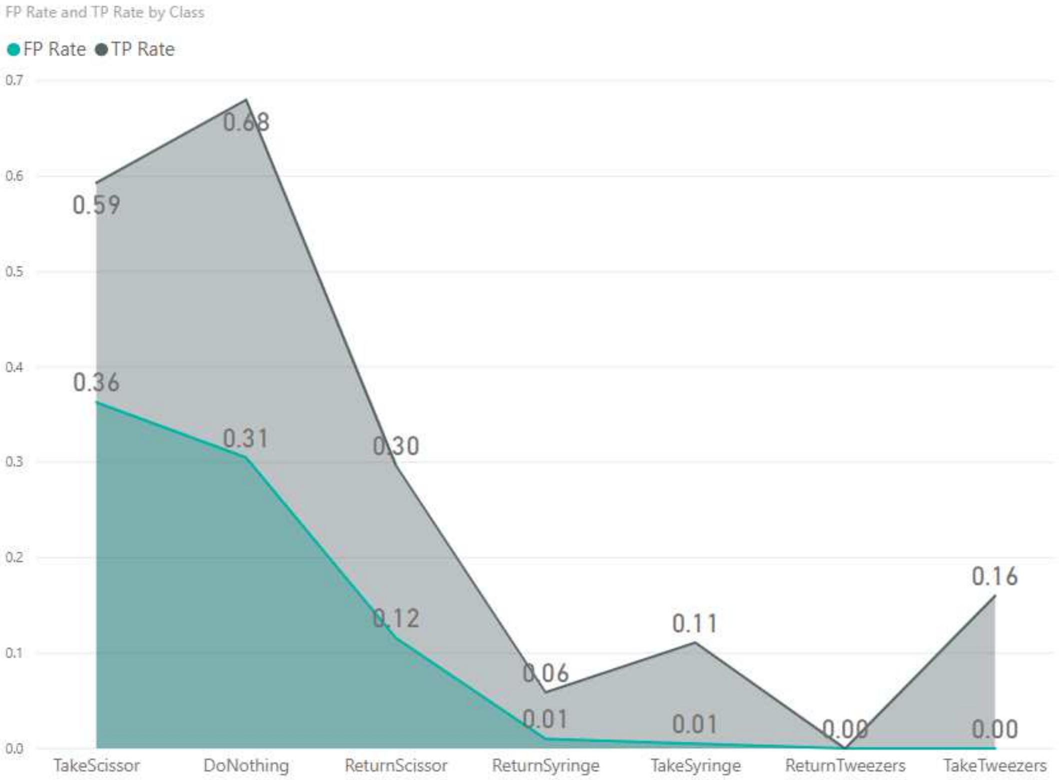
Source: Elaborated by the author.

Figure 23 – Bayesian Network Precision rate.



Source: Elaborated by the author.

Figure 24 – Random Forest False Positive vs True Positive rate.



Source: Elaborated by the author.

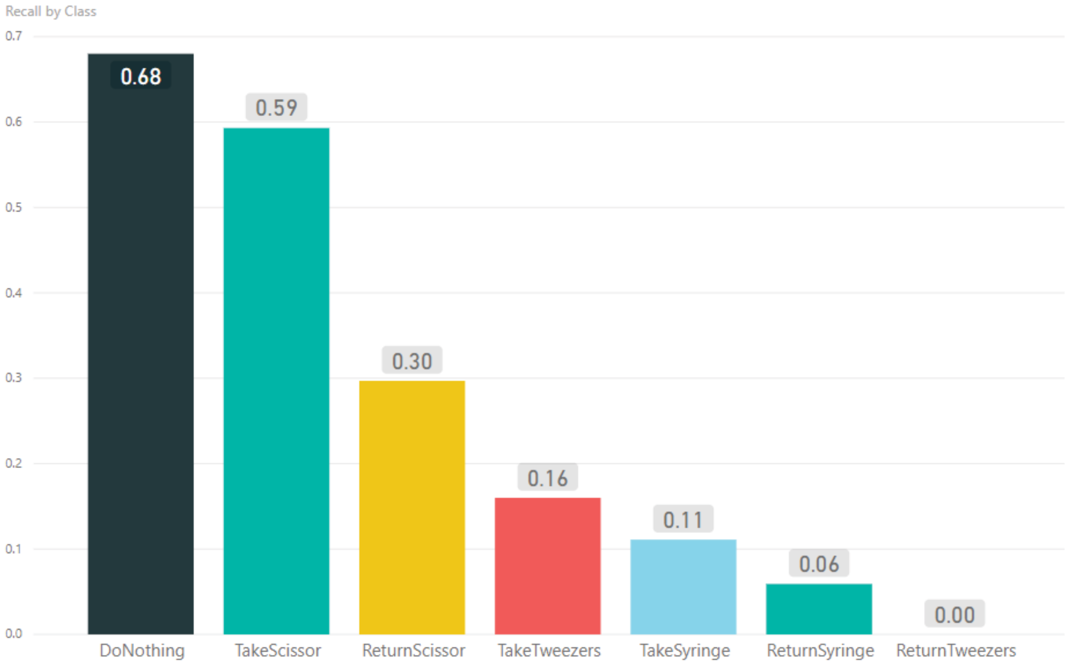
5.3.2.2 Random Forest

A basic Random Forest algorithm was used to implement the RF without depth limit and using two decimal places to the calculations and evaluated using 10 fold cross-validation. Here we can see more of the expected behavior. Three classes (Take Scissor, Do Nothing, and Return Scissor) are recognized with acceptable scores. We could have a better classification in those three classes, but the RF has a tendency to classify every item into those three classes. The recall measure in figure 25 represents a great improvement over the Bayesian network, with just class Return Tweezers without recognizing. With the random forest, we have an overall accuracy rate of 38.33%. The figures 24 and 26 complements the tests data.

5.3.2.3 Deep Neural Network

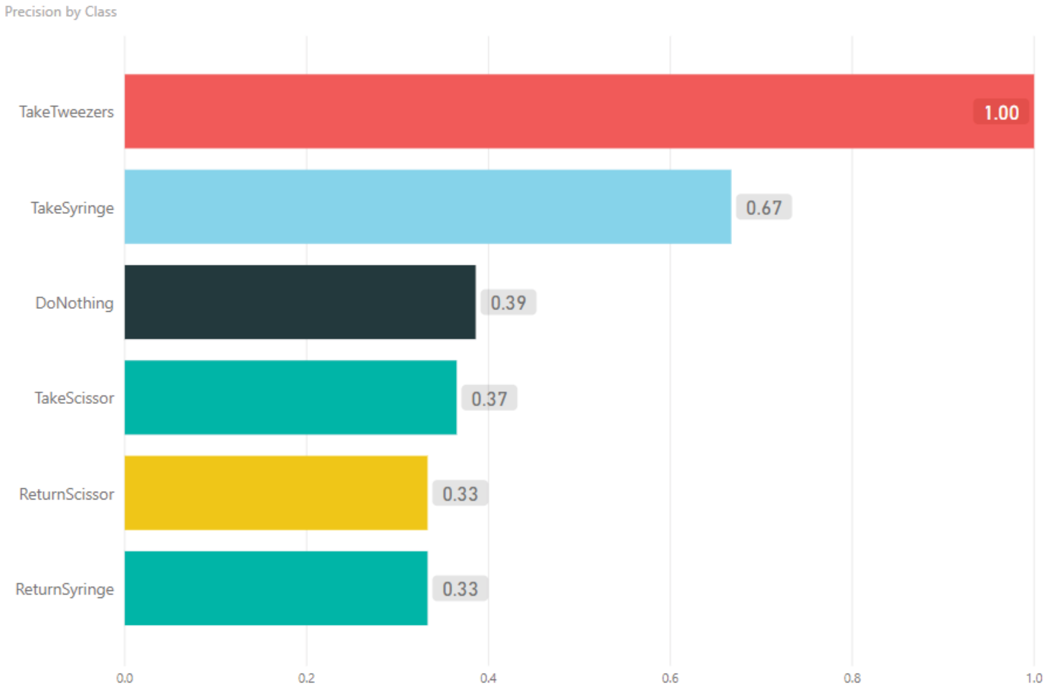
The Neural Network was trained with a 4-layer architecture, the first being 16 nodes, the second 12 nodes, the third 10 nodes, and the final with one node. The validation process used K-fold Cross-validation with ten folds. Three networks were run with random startup parameters, and the best one was used in the final process. The figures 27, 22 and 29 shows the obtained results.

Figure 25 – Random Forest Network Recall rate.



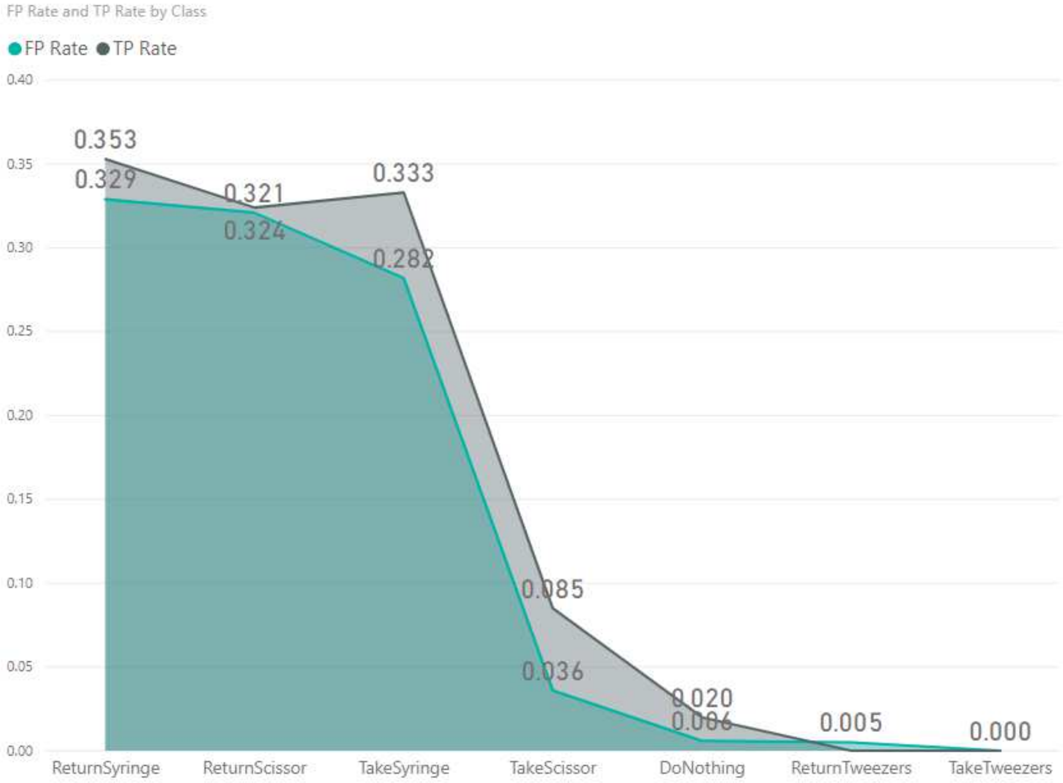
Source: Elaborated by the author.

Figure 26 – Random Forest Network Precision rate.



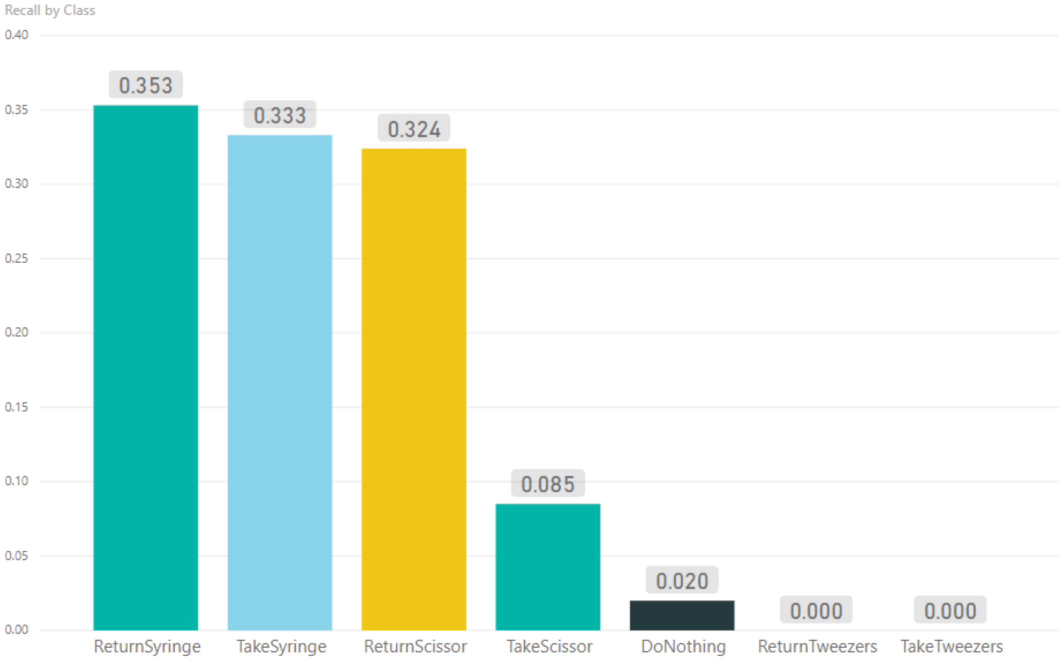
Source: Elaborated by the author.

Figure 27 – Deep Neural Network False Positive vs True Positive rate.



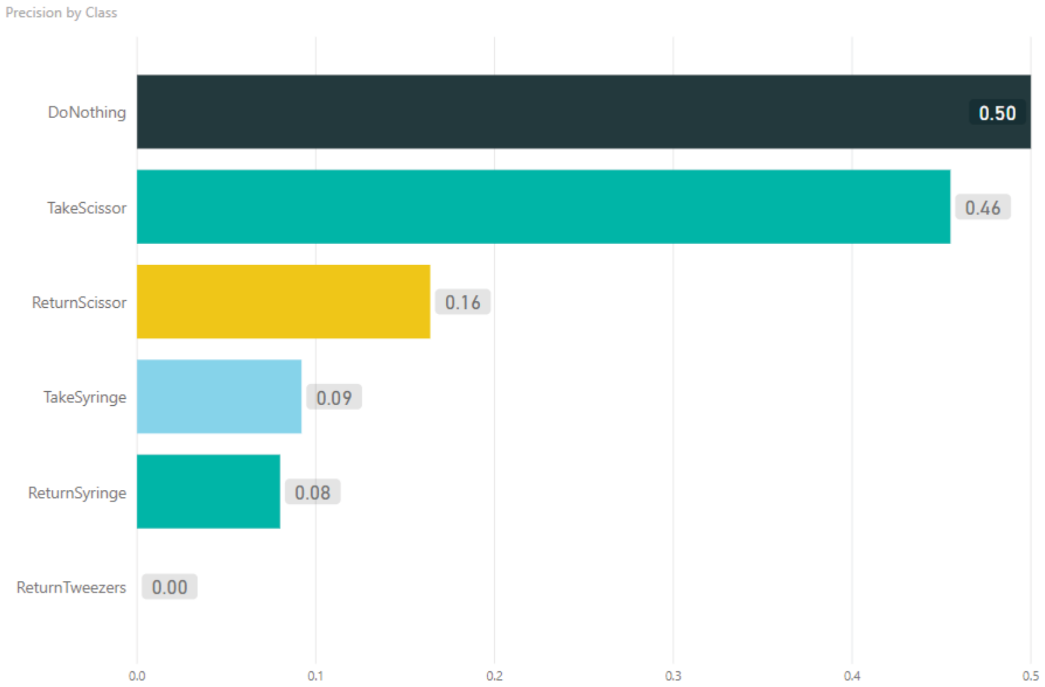
Source: Elaborated by the author.

Figure 28 – Deep Neural Network Recall rate.



Source: Elaborated by the author.

Figure 29 – Deep Neural Network Precision rate.



Source: Elaborated by the author.

5.3.3 Results Discussion

We tested the extraction with different intersection between images percentages. The best one was 50% and was the one used in the final results. The results were quite low, with the best classification in 38.33% and followed by poor performance in some classes. Some classes were not even recognized in the classification tests what brings us to two major problems: (I) The dataset was few examples, and the classification methods are not able to understand the data; (II) The data is not well-formatted to explain the class behavior. The first one is a delimitation of this work, where we want to detect this dataset's actions. However, there was space for improvements in how data was delivered to the action recognition step. This motivated the alteration of the model in a new version with a filter for such cases.

There are situations during the surgical procedure, in which many elements are on the table, but only one is being moved. This situation is quite common in the dataset ended up negatively affecting the recognition performance. The stock detection algorithm collects, processes, and sends data for all elements. When analyzing the data, it is possible to identify a large amount of information submitted to the classification algorithms with no significance in the context and do not help the action's detection. This information ends up degrading the final performance. Besides, there is unnecessary processing of elements not involved in the context of the action. Once we detected this challenge, we prepared a version to filter those elements. We present and evaluated it in the next section.

5.4 Improvements

Our research development proposal, presented in section 1.4, expected the model implementation and, based on the results, modifications to the model. This section offers the evolution of the model, based on the acquired performance results.

5.4.1 Action Recognition Redesign

For the recognition of actions, a Python application obtains input a tensor with the positions of each bounding box for each recognized object in each frame for an action. The number of frames per action is defined by the dataset, ranging from 7 to 20 frames. The first step of the algorithm is the removal of objects that have not changed their position. This is done because there are frames that contain more than 14 objects on the table. The goal is to filter only elements that have moved in the video, as these are the objects involved in the action.

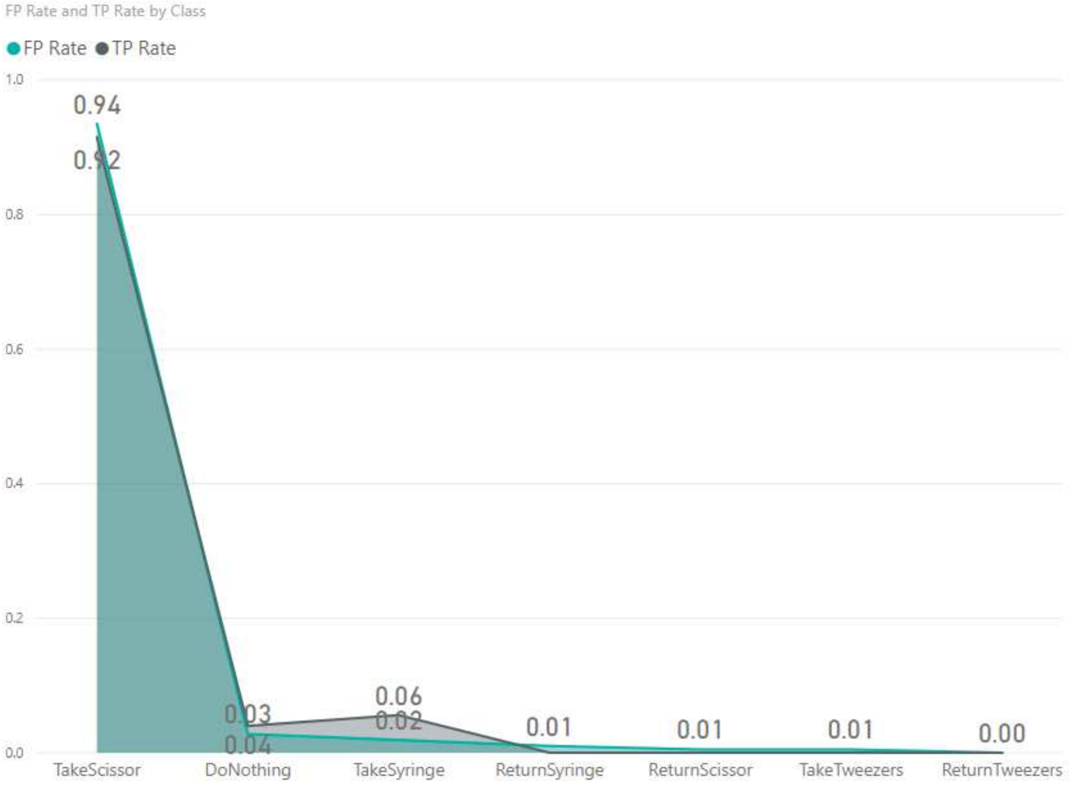
Once the tensors are obtained, the application initially compares each object's positioning in the frames by measuring the IoU overlapping within a threshold. In this comparison, there are two possible situations:

- If the object is matched for two consecutive frames and the object's movement is within the threshold, it is considered to be unmovable, and only its last position is maintained to compare.
- If there is no match for two sequential frames, that is, in the first frame in some position, the object was identified, and in the next frame not, this object is registered as a point of attention. It could be a moving object, or just an object overlapped by hands.

At the end of the last frame, there is the final measurement. We compare the objects labeled as a point of attention with the last frame. If there is an IoU match, any object identified as a point of attention with the last frame will no longer be marked. Only objects with no correspondence between frames and marked as points of attention, are kept. The objects that had matches, being thus without movement, are removed from the tensor. In the second stage, there is a match between elements. In this step, the number of elements of the same class in each frame is initially checked. If there is only one element of the same class that moved between frames, its motion vector is calculated based on the Euclidean Distance over the frames.

For cases where more than one object of the same class moved, the Euclidean Distance between frames is calculated and selected as corresponding to the shortest distance. At the end of this stage, there are vectors of movement of objects and their respective classes. Hands are treated as objects and go through the same steps, are just classified as "Hands". In the third phase of action detection, the previous step's vectors are evaluated by a classification algorithm. This algorithm is responsible for retrieving the action output.

Figure 30 – Bayesian Network False Positive vs True Positive rate for the second experiment.



Source: Elaborated by the author.

5.5 Final Evaluation

This last evaluation comprehends the modifications in the model discussed in section 5.4. The adjustments are exclusively related to the Action Detector step, considering that the object detector had adequate results.

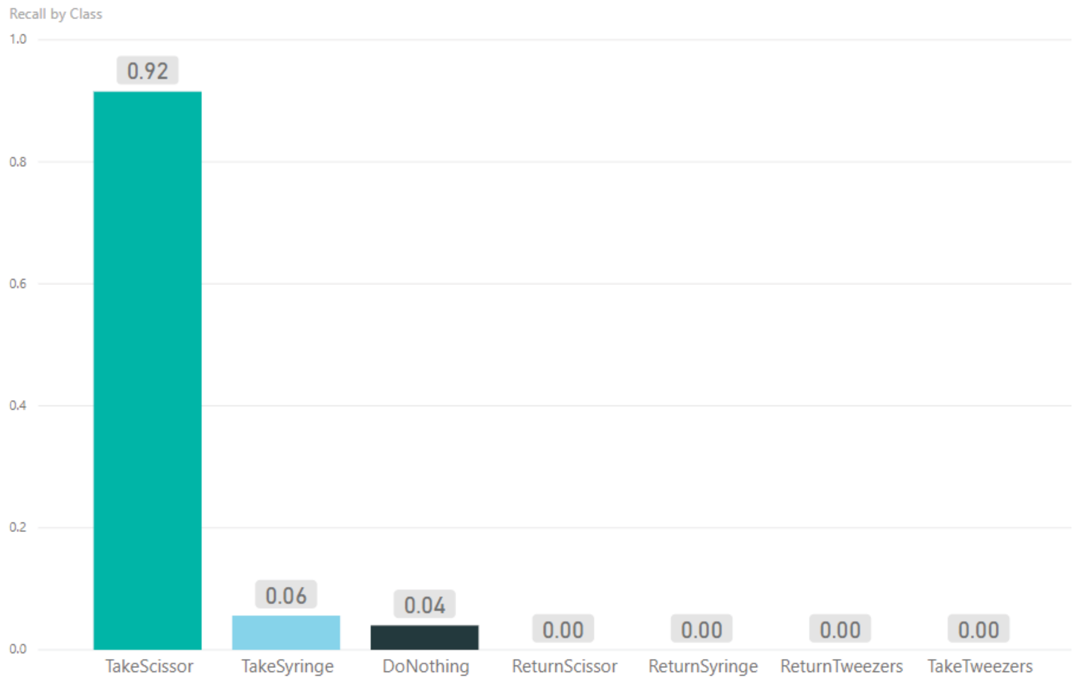
5.5.1 Action Detector

The action detector was tested in the same environment as before, to enable comparisons. All process was using 2 decimal places and ten fold cross-validation and the intersection between images was keep in 50%. Also the dataset was the same as used before. In the next sections we will present the obtained results with the new configuration, and discuss them on section 5.5.2.

5.5.1.1 Bayesian Network

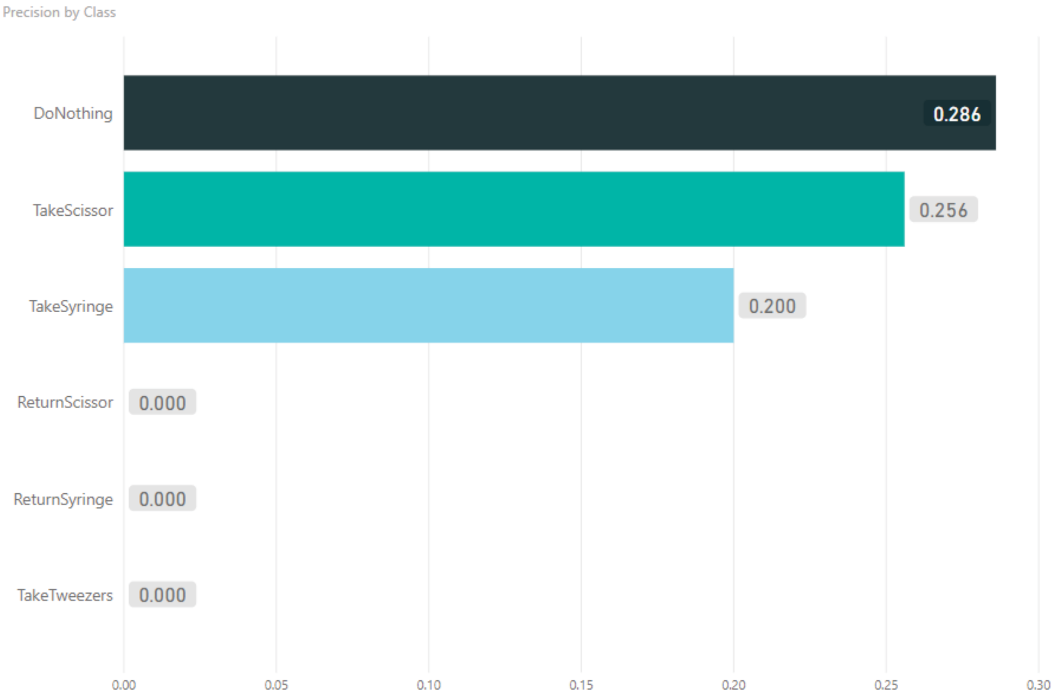
The same Bayesian Network was applied, and we can observe the same behavior as before. There was a huge tendency to classify all items as Take Scissor class. The final accuracy was even worse, with 25.1%. The figures 30, 31 and 32 shows the evaluation results.

Figure 31 – Bayesian Network Recall rate for the second experiment.



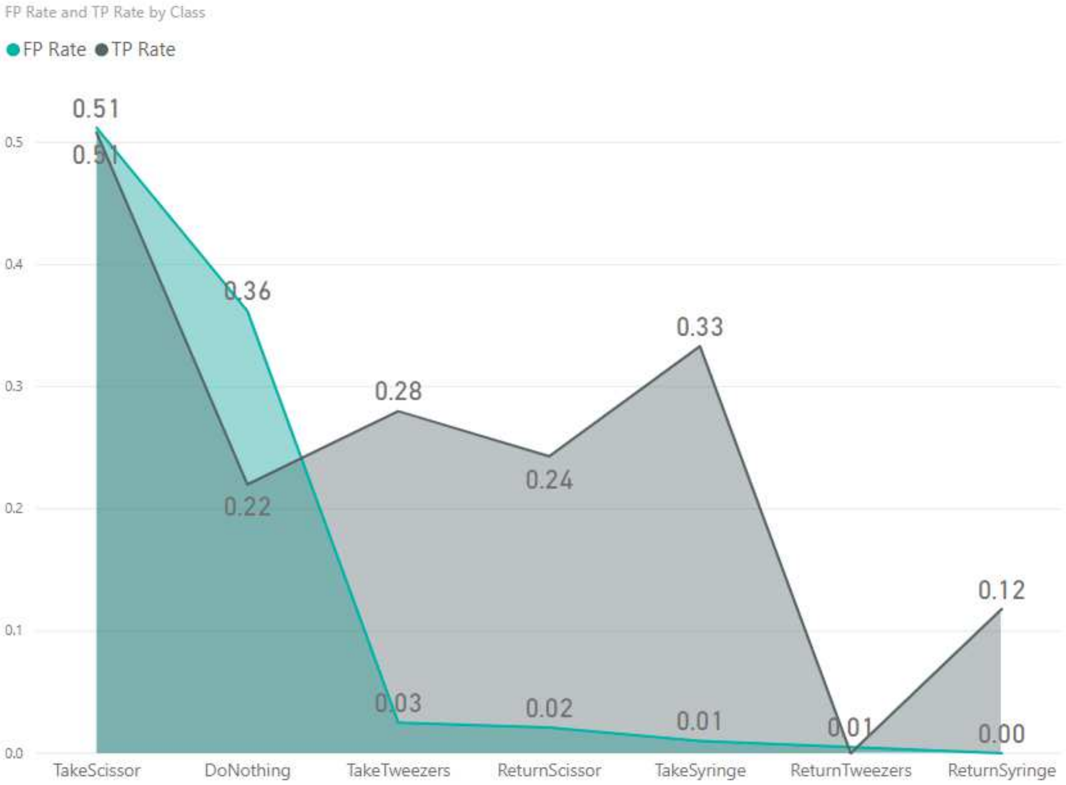
Source: Elaborated by the author.

Figure 32 – Bayesian Network Precision rate for the second experiment.



Source: Elaborated by the author.

Figure 33 – Random Forest False Positive vs True Positive rate for the second experiment.



Source: Elaborated by the author.

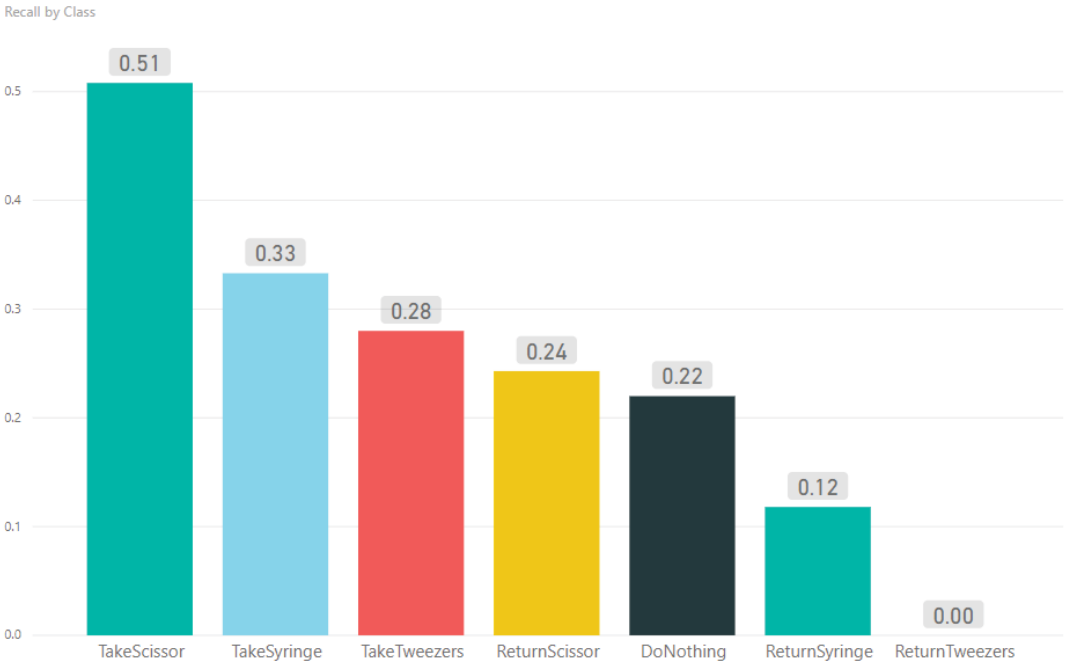
5.5.1.2 Random Forest

The Random Forest presented an overall accuracy of 28.1%, 10% lower than the first experiment. We still have bad False Positive rates into the two best-classified classes, as shown in figure 33. However, we got a lot better overall results than the first version of the action recognition model. Even if the overall rate was worse, the first version just tried to classify into three classes. In comparison, now we have better distribution results. Figures 34 and 35 complement the information about recall and precision respectively.

5.5.1.3 Deep Neural Network

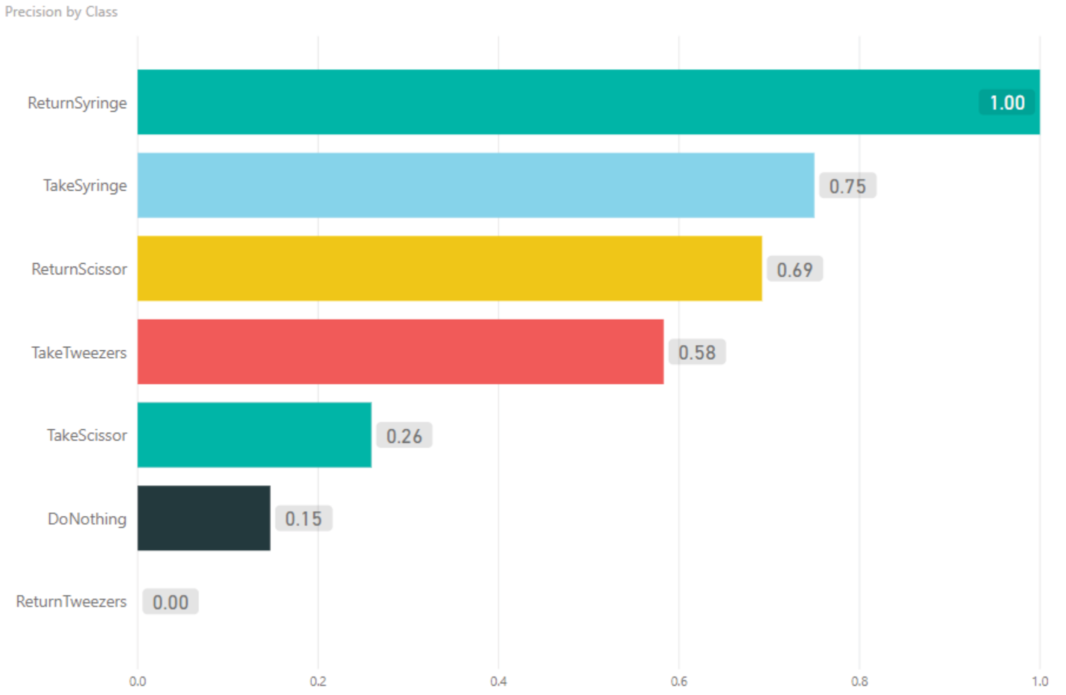
The Neural Network was trained with the same configuration as the first experiment. We got the best results with overall 44.05% accuracy, and the model was able to recognize all classes. The False Positive errors where a lot more well distributed, as we can observe in figure 36. The NN was excellent in detecting the "Do Nothing"class, the most simple one. The other classes are more complex, but the model's modifications helped the NN classify better. The figures, 31 and 38 complements the obtained results.

Figure 34 – Random Forest Network Recall rate for the second experiment.



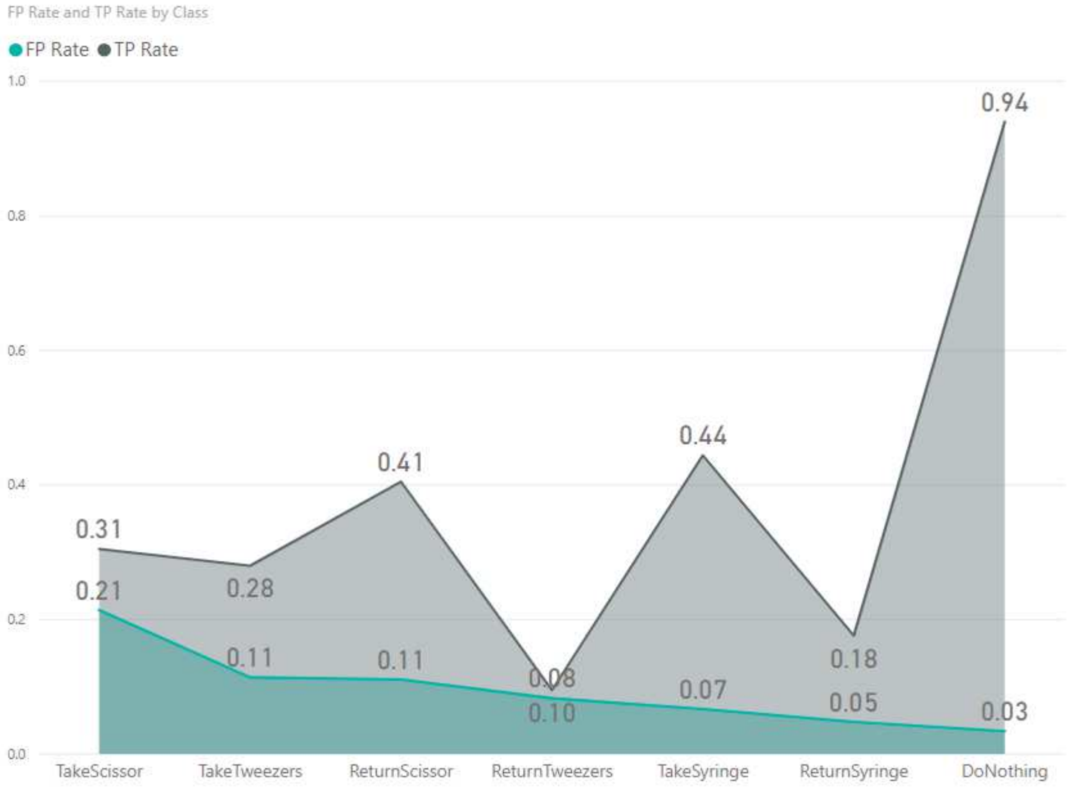
Source: Elaborated by the author.

Figure 35 – Random Forest Network Precision rate for the second experiment.



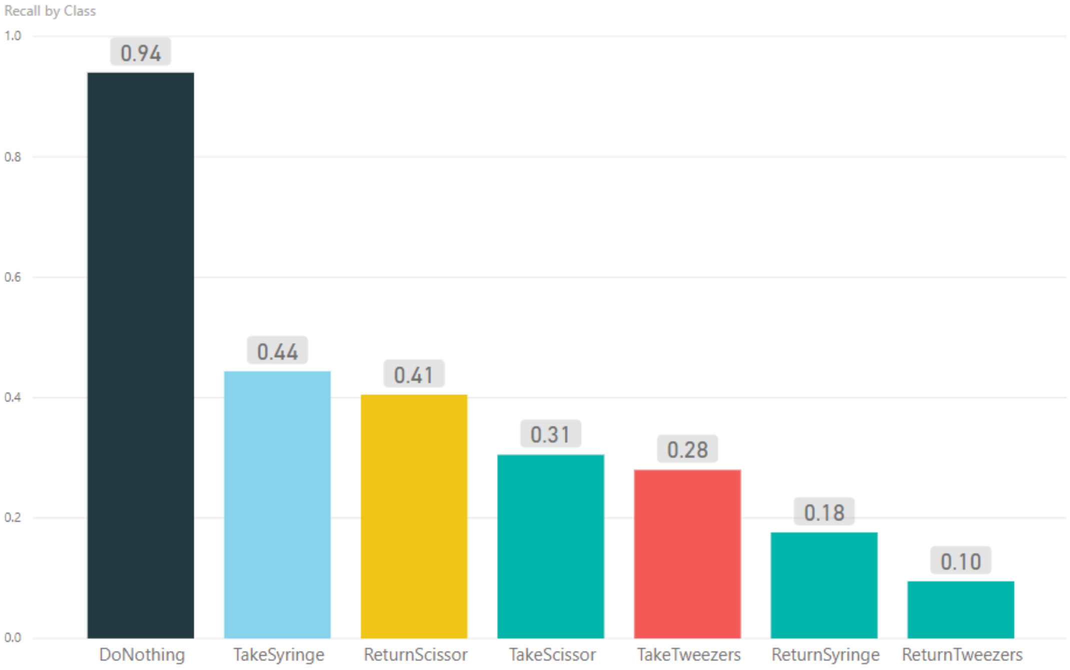
Source: Elaborated by the author.

Figure 36 – Deep Neural Network False Positive vs True Positive rate for the second experiment.



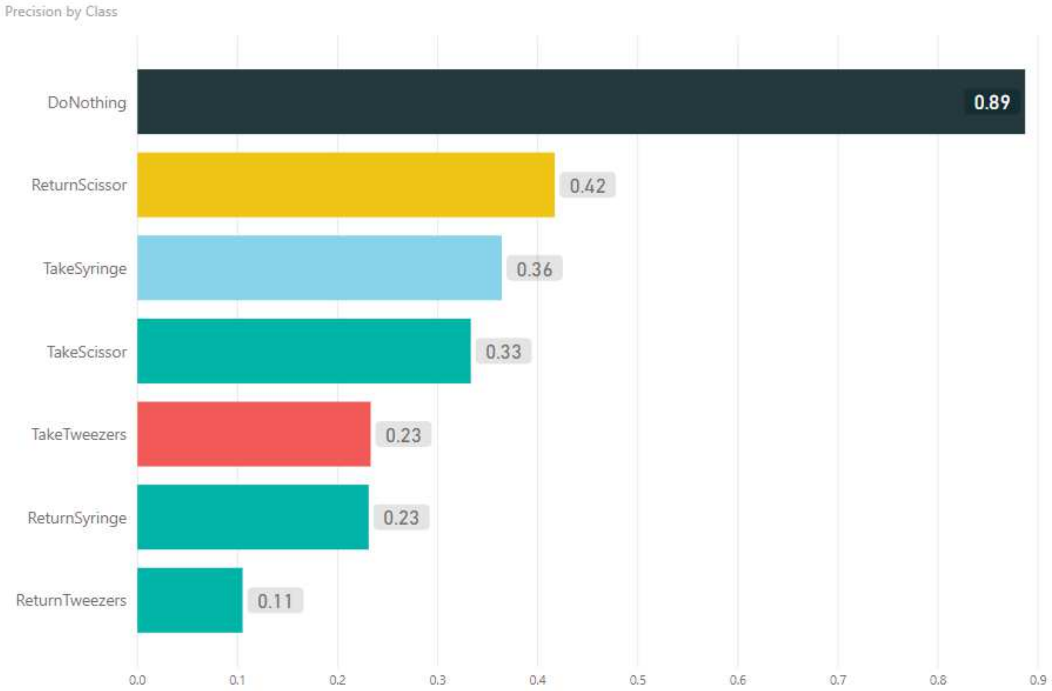
Source: Elaborated by the author.

Figure 37 – Deep Neural Network Recall rate for the second experiment.



Source: Elaborated by the author.

Figure 38 – Deep Neural Network Precision rate for the second experiment.



Source: Elaborated by the author.

5.5.2 Results Discussion

With less information in the tensors, the new version of the model showed worse results in Bayesian Networks and Random Forests, but a significant improvement using DNN. The dataset is challenging by its actions, since "Return"and "Take"actions implicate the same objects use, just changing the movement. In table 2, we present the confusion matrix of the DNN using the improved version. For example, in the "Take Scissor"class, we can observe that there is a considerable amount of confusion between "Take"and "Return"actions using the same objects. This shows us that the network is not able to understand the differences between them. Even so, we got great results with the DNN, with good and balanced False Positive rates, which shows a promising direction for this research.

Table 2 – Confusion Matrix of the final proposed model using DNN.

a	b	c	d	e	f	g	Classified as
47	1	0	0	1	0	1	a = Do Nothing
0	15	1	5	13	1	2	b = ReturnScissor
0	0	3	3	5	3	3	c = ReturnSyringe
0	4	0	2	6	4	5	d = ReturnTweezers
2	7	7	7	18	6	12	e = TakeScissor
1	2	1	0	6	8	0	f = TakeSyringe
3	7	1	2	5	0	7	g = TakeTweezers

Source: Elaborated by the author.

For the final version, we also calculated the F1-Score, presented in image 40. F1-Measures are interesting to measure the test accuracy and are calculated from precision and recall, as described in equation 5.6. Since F1-Score gives a large weight to lower numbers, we can see that just the class "DoNothing"had good results, with the others presenting results with 0.41 and above. The results show the deficiency of the model to detect the classes with both precision and Recall.

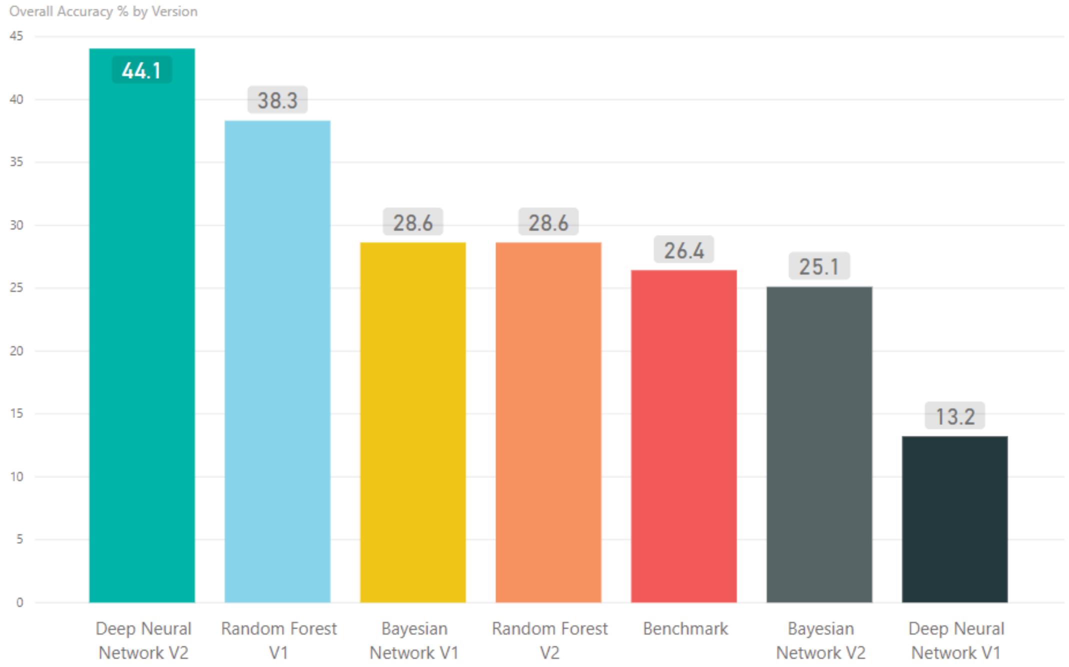
$$F1 - Score = 2.\frac{Precision.Recall}{Precision + Recall} \tag{5.6}$$

Also, to evaluate the results, we calculated the Receiver Operating Characteristic (ROC) values for each class. Its a measure of how good a method is to distinguish between the other categories. The ROC values are calculated as a curve between the false positive and true positive fractions. In figure 41 we calculated the resultant area for each class. With near to 1, the better the results are. We got great results in this metric, showing that the DNN method distinguished the classes, with a weighted average of 0.69.

Then, we understand that the DNN in the proposed solution is going in the right direction, but misses more examples to understand the difference between the classes better. The used dataset has 3045 annotated frames that were good enough to train the object detector. When classified into actions, it resulted in 227 action examples and was insufficient to have enough examples for training the classification algorithms. Also, the examples are not balanced, and this is also a challenge to the classifiers. Even though the balanced results between positive and false rates and the good overall ROC area values show that the last proposed solution presents a good strategy to detect the actions.

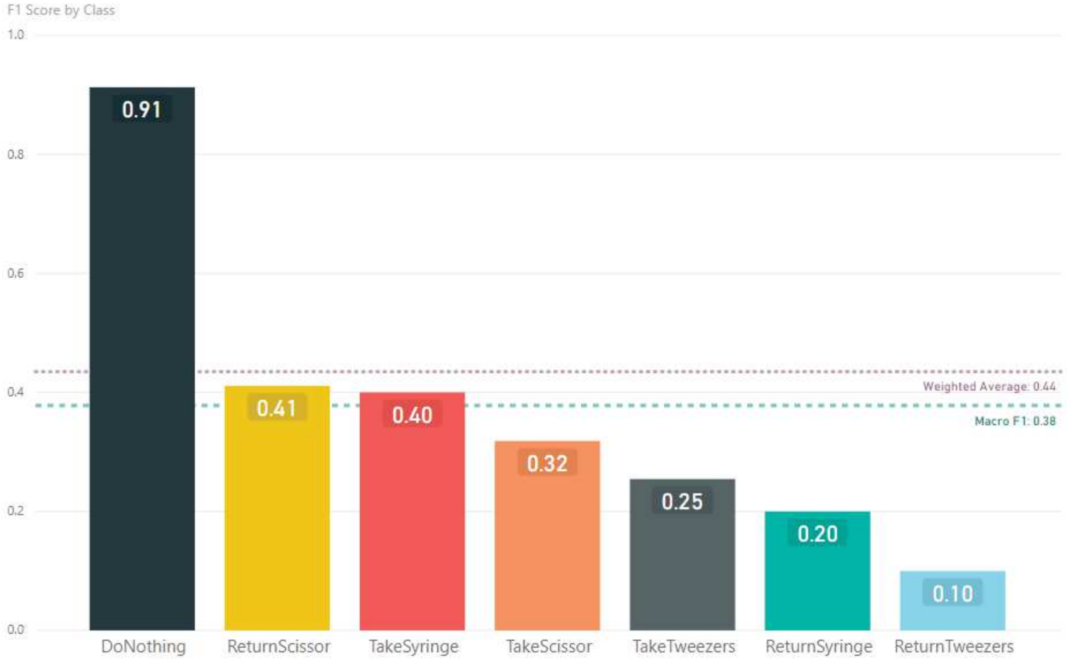
Now, we come back to evaluate our hypothesis, presented in section 1.2. We want to evaluate if the presented model is better than the Zero Rule benchmark. Figure 39 shows all tested configurations against the benchmark. We were able to obtain 44.1% accuracy, where the benchmark was 26.4%. An increment of 67.95%. This research's next natural step is to test the proposed method in a well-known dataset and evaluate it against other works. This work is already in progress, and we will present the results in a future paper.

Figure 39 – Accuracy of tested models against benchmark.



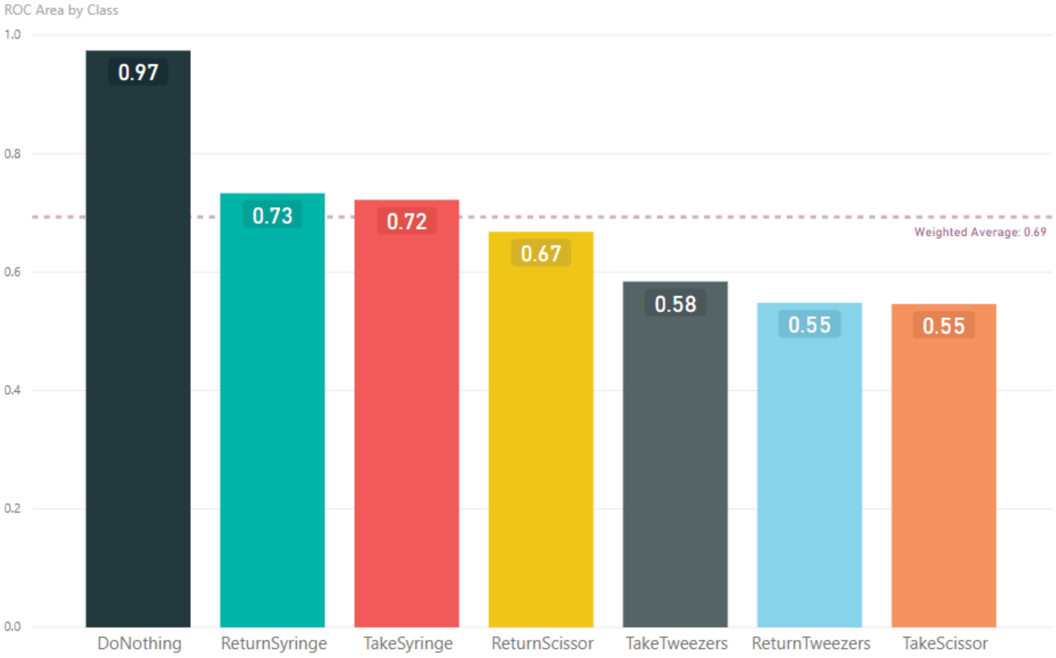
Source: Elaborated by the author.

Figure 40 – F1 score of each class for the final model.



Source: Elaborated by the author.

Figure 41 – ROC area for each class for the final model.



Source: Elaborated by the author.

6 CONCLUSION

Human action recognition can be considered one of the most critical challenges in video analysis problems due to its extensive applicability and the complexity of the motion patterns produced by body movements. In this work, our main objective was to propose a model to simulate the human brain-behavior to recognizes actions and then propose a workflow with algorithms to implement this model on a computer. In order to achieve this goal, we seek to understand what is known about the functioning of the human brain to implement in a new model of HPE. This model, named Brain Action, takes into account this study as well as the state of the art study in HAR. For the model, we proposed a new workflow divided into five steps: Input, Object Recognition, Hand Recognition, Movement Feature Extraction, and Action Recognition. For this model, a workflow implementation proposal was also presented, using as support a widely used object recognition CNN and introducing a new way to extract features to obtain the movement of the objects involved.

In this work, we annotated a new challenging RGB Video dataset, based on real-world surgery data, with information about objects, hands, and actions. This dataset comprises more than 34 thousand object annotations in 3045 frames and is divided by seven different actions resulting in 227 video fragments. To test the model, we presented a methodology to thoroughly examine the workflow using state-of-the-art metrics and two implementation versions. Each one was tested using Random Forests, Bayesian Networks, and Deep Neural Networks as final classifiers. As a final result, we evaluated six different model configurations enabling comparisons and discussions about each one. In this work hypothesis, we proposed to assess if the presented model is better than the Zero Rule benchmark. We were able to obtain 44.1% accuracy, where the reference was 26.4%—an increment of 67.95%, showing a promising research direction.

A model that proves to be robust can serve as the basis for developing solutions in the most varied branches. The development of HAR technologies has great potential to solve societal problems and be essential in the expansion of technologies such as health care, smart cities, education, security, and sports. Thus, we understand that the development of this work and research in this area will positively impact people's lives in the future. We hope that by proposing this new model and we can assist in this process.

HAR is a broad area, and we had to draw our attention to some specific points for this work. In this sense, we review only works that obtain their data through cameras, discarding valid alternatives, such as device sensors or wearables. Also, we have limited the search to the last five years, and we only study in-depth detail proposes that were using similar approaches to this work. Therefore, we understand as opportunities for future work to close those gaps. This research will continue with evaluation of the other datasets.

6.1 Contributions

During the development of this research, we tried to understand how other proposals with similar ideas solve the same problem, as well as solutions that investigate other different methods. We have gathered this knowledge to propose a model that can explore techniques that are already accepted in state of the art with the human mind's recognition of actions. With this, we expect the following contributions:

The following contributions from the Brain Action proposal can be highlighted:

1. A new HAR model inspired by how the human brain recognizes actions;
2. A new movement feature extraction design based on object movement;
3. A HAR implementation for surgical action recognition scenario based on a real-world dataset.

6.2 Limitations and Future Works

In this section, we list the limitations encountered during the development/implementation of the full-scale model or are defined by the implementation proposal.

1. The tested surgical scenario in which the model was applied is quite specific and insufficient to prove the effectiveness of the model. Future testing in other scenarios will be needed to determine its effectiveness.
2. The used dataset may be insufficient to train the model with reasonable accuracy. Other datasets should be tested in the future and compared to state-of-art values to evaluate the model. In the resulting paper of this work, we will include tests in the UCF101 (SOOMRO; ZAMIR; SHAH, 2012) dataset, to enable comparing with other works.
3. The equipment used to execute the CNN may not have sufficient processing power to perform the complete workflow satisfactorily. Testing on other equipment configurations is required to compare against different suggested workflows.
4. The Number of frames/movements needed to set up action is not considered in the implementation scenario, as videos are already clipped with the actions. This situation is an opportunity for improvement in a future proposal that should consider how to recognize actions in full-length videos.

6.3 Publications

During this research, we produced two kinds of papers: First, documents to share the systematized information obtained during the literature review. Then an article to share the proposed model’s main idea, and the findings during the carried out tests.

During the first phase, the article entitled "Understanding Skeleton Action Recognition systems: A Systematic Literature Review"was produced, containing a systematic review of 222 works in the HPE area and answering eight essential questions. In this same article, we organized the proposals in a new taxonomy to assist new researchers in the area and point out future directions and research opportunities.

The first phase of the work helped define a path to do action recognition using objects, resulting in a second systematic review. The work is entitled "Human Action Recognition in videos using objects: A systematic literature review"and contains the last five years of action recognition based on objects, answering essential questions, and presenting trending topics. At the end of this work, we started making an article comprising the model, the tests, and the results obtained.

I value scientific production as a way to recognize the work as well as share the discoveries. We are updating the first review with the latest works to publish it, but all the article construction is already done. JVIS is currently analyzing the other object-based action recognition survey, while the final article is in the construction phase.

REFERENCES

AFIFI, M. 11k hands: gender recognition and biometric identification using a large dataset of hand images. **Multimedia Tools and Applications**, [S.l.], 2019.

AGGARWAL, J. K.; RYOO, M. S. Human activity analysis: a review. **ACM Computing Surveys (CSUR)**, [S.l.], v. 43, n. 3, p. 16, 2011.

ALLOGHANI, M. et al. A systematic review on supervised and unsupervised machine learning algorithms for data science. In: **Supervised and unsupervised learning for data science**. [S.l.]: Springer, 2020. p. 3–21.

ANN, O. C.; THENG, L. B. Human activity recognition: a review. In: IEEE INTERNATIONAL CONFERENCE ON CONTROL SYSTEM, COMPUTING AND ENGINEERING (ICCSCE 2014), 2014., 2014. **Anais. . .** [S.l.: s.n.], 2014. p. 389–393.

AREL, I. et al. Deep machine learning-a new frontier in artificial intelligence research. **IEEE computational intelligence magazine**, [S.l.], v. 5, n. 4, p. 13–18, 2010.

BAMBACH, S. et al. Lending a hand: detecting hands and recognizing activities in complex egocentric interactions. In: THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), 2015. **Anais. . .** [S.l.: s.n.], 2015.

BARADEL, F.; WOLF, C.; MILLE, J. Human action recognition: pose-based attention draws focus to hands. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2017. **Proceedings. . .** [S.l.: s.n.], 2017. p. 604–613.

BEELMANN, A. **Review of systematic reviews in the social sciences. a practical guide.** [S.l.]: Hogrefe & Huber Publishers, 2006. v. 11, n. 3.

BIEDERMAN, I. Recognition-by-components: a theory of human image understanding. **Psychological review**, [S.l.], v. 94, n. 2, p. 115, 1987.

BIOLCHINI, J. et al. Systematic review in software engineering. **System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES**, [S.l.], v. 679, n. 05, p. 45, 2005.

BISHOP, C. **Pattern recognition and machine learning**. [S.l.]: Springer (India) Private Limited, 2013. (Information science and statistics).

BRERETON, P. et al. Lessons from applying the systematic literature review process within the software engineering domain. **Journal of systems and software**, [S.l.], v. 80, n. 4, p. 571–583, 2007.

CAO, L. et al. Gchar: an efficient group-based context—aware human activity recognition on smartphone. **Journal of Parallel and Distributed Computing**, [S.l.], v. 118, p. 67–80, 2018.

CHAO, Y.-W. et al. Hico: a benchmark for recognizing human-object interactions in images. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2015. **Proceedings. . .** [S.l.: s.n.], 2015. p. 1017–1025.

CHAO, Y.-W. et al. Learning to detect human-object interactions. In: 2018 IEEE WINTER CONFERENCE ON APPLICATIONS OF COMPUTER VISION (WACV), 2018. **Anais. . .** [S.l.: s.n.], 2018. p. 381–389.

CHEN, Y. et al. Cascaded pyramid network for multi-person pose estimation. **arXiv preprint arXiv:1711.07319**, [S.l.], 2017.

DANGETI, P. **Statistics for machine learning**. [S.l.]: Packt Publishing, 2017.

DANTONE, M. et al. Body parts dependent joint regressors for human pose estimation in still images. **IEEE transactions on pattern analysis and machine intelligence**, [S.l.], v. 36, n. 11, p. 2131–2143, 2014.

DAS, S. et al. Where to focus on for human action recognition? In: IEEE WINTER CONFERENCE ON APPLICATIONS OF COMPUTER VISION (WACV), 2019., 2019. **Anais. . .** [S.l.: s.n.], 2019. p. 71–80.

DENG, J. et al. Imagenet: a large-scale hierarchical image database. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2009., 2009. **Anais. . .** [S.l.: s.n.], 2009. p. 248–255.

DICARLO, J. J.; ZOCCOLAN, D.; RUST, N. C. How does the brain solve visual object recognition? **Neuron**, [S.l.], v. 73, n. 3, p. 415–434, 2012.

EVERINGHAM, M. et al. The pascal visual object classes (voc) challenge. **International journal of computer vision**, [S.l.], v. 88, n. 2, p. 303–338, 2010.

FELLEMAN, D. J.; VAN, D. E. Distributed hierarchical processing in the primate cerebral cortex. **Cerebral cortex (New York, NY: 1991)**, [S.l.], v. 1, n. 1, p. 1–47, 1991.

FELZENSZWALB, P. F. et al. Object detection with discriminatively trained part-based models. **IEEE transactions on pattern analysis and machine intelligence**, [S.l.], v. 32, n. 9, p. 1627–1645, 2009.

FIELDS, C. The very same thing: extending the object token concept to incorporate causal constraints on individual identity. **Advances in cognitive psychology**, [S.l.], v. 8, n. 3, p. 234, 2012.

FIELDS, C. Editorial: how humans recognize objects: segmentation, categorization and individual identification. **Frontiers in Psychology**, [S.l.], v. 7, p. 400, 2016.

FLORES-VÁZQUEZ, C.; ARANDA, J. Human activity recognition from object interaction in domestic scenarios. In: IEEE ECUADOR TECHNICAL CHAPTERS MEETING (ETCM), 2016., 2016. **Anais. . .** [S.l.: s.n.], 2016. p. 1–6.

GOLESTANI, N.; MOGHADDAM, M. Human activity recognition using magnetic induction-based motion signals and deep recurrent neural networks. **Nature communications**, [S.l.], v. 11, n. 1, p. 1–11, 2020.

GOODALE, M. A.; MILNER, A. D. Separate visual pathways for perception and action. **Trends in neurosciences**, [S.l.], v. 15, n. 1, p. 20–25, 1992.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT press, 2016.

GUPTA, S.; MALIK, J. Visual semantic role labeling. **arXiv preprint arXiv:1505.04474**, [S.l.], 2015.

HARRINGTON, P. **Machine learning in action**. [S.l.]: Manning Publications Co., 2012.

HE, K. et al. Deep residual learning for image recognition. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2016. **Proceedings. . .** [S.l.: s.n.], 2016. p. 770–778.

HOLTE, M. B. et al. Human pose estimation and activity recognition from multi-view videos: comparative explorations of recent developments. **IEEE Journal of selected topics in signal processing**, [S.l.], v. 6, n. 5, p. 538–552, 2012.

HU, J.-F. et al. Recognising human-object interaction via exemplar based modelling. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2013. **Proceedings. . .** [S.l.: s.n.], 2013. p. 3144–3151.

HU, J.-F. et al. Exemplar-based recognition of human–object interactions. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v. 26, n. 4, p. 647–660, 2015.

HU, T. et al. Human action recognition based on scene semantics. **Multimedia Tools and Applications**, [S.l.], v. 78, n. 20, p. 28515–28536, 2019.

HUBEL, D. H.; WIESEL, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. **The Journal of physiology**, [S.l.], v. 160, n. 1, p. 106–154, 1962.

HUSSAIN, Z.; SHENG, M.; ZHANG, W. E. Different approaches for human activity recognition: a survey. **arXiv preprint arXiv:1906.05074**, [S.l.], 2019.

ISIK, L.; TACCHETTI, A.; POGGIO, T. A fast, invariant representation for human action in the visual system. **Journal of neurophysiology**, [S.l.], v. 119, n. 2, p. 631–640, 2017.

ITTI, L.; KOCH, C.; NIEBUR, E. A model of saliency-based visual attention for rapid scene analysis. **IEEE Transactions on Pattern Analysis & Machine Intelligence**, [S.l.], n. 11, p. 1254–1259, 1998.

JACOBY, A. R. et al. Context-sensitive human activity classification in collaborative learning environments. In: IEEE SOUTHWEST SYMPOSIUM ON IMAGE ANALYSIS AND INTERPRETATION (SSIAI), 2018., 2018. **Anais. . .** [S.l.: s.n.], 2018. p. 1–4.

JEAN-BAPTISTE, E. M.; MIHAILIDIS, A. Benefits of automatic human action recognition in an assistive system for people with dementia. In: IEEE CANADA INTERNATIONAL HUMANITARIAN TECHNOLOGY CONFERENCE (IHTC), 2017., 2017. **Anais. . .** [S.l.: s.n.], 2017. p. 61–65.

JIANG, X.; XU, K.; SUN, T. Action recognition scheme based on skeleton representation with ds-lstm network. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], 2019.

JONIDES, J. Further toward a model of the mind's eye's movement. **Bulletin of the Psychonomic Society**, [S.l.], v. 21, n. 4, p. 247–250, 1983.

KE, Q. et al. Learning clip representations for skeleton-based 3d action recognition. **IEEE Transactions on Image Processing**, [S.l.], v. 27, n. 6, p. 2842–2855, 2018.

KHAN, F. S. et al. Coloring action recognition in still images. **International journal of computer vision**, [S.l.], v. 105, n. 3, p. 205–221, 2013.

KIM, S. et al. Skeleton-based action recognition of people handling objects. In: IEEE WINTER CONFERENCE ON APPLICATIONS OF COMPUTER VISION (WACV), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p. 61–70.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering–a tertiary study. **Information and Software Technology**, [S.l.], v. 52, n. 8, p. 792–805, 2010.

KOSILO, M. et al. Low-level and high-level modulations of fixational saccades and high frequency oscillatory brain activity in a visual object classification task. **Frontiers in psychology**, [S.l.], v. 4, p. 948, 2013.

KOTSIANTIS, S. B.; ZAHARAKIS, I.; PINTELAS, P. Supervised machine learning: a review of classification techniques. **Emerging artificial intelligence applications in computer engineering**, [S.l.], v. 160, p. 3–24, 2007.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 2012. **Anais...** [S.l.: s.n.], 2012. p. 1097–1105.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, [S.l.], v. 521, n. 7553, p. 436–444, 2015.

LENZI, V. B.; MORETTI, G.; SPRUGNOLI, R. Cat: the celct annotation tool. In: LREC, 2012. **Anais...** [S.l.: s.n.], 2012. p. 333–338.

LI, B. et al. 3d skeleton based action recognition by video-domain translation-scale invariant mapping and multi-scale dilated cnn. **Multimedia Tools and Applications**, [S.l.], v. 77, n. 17, p. 22901–22921, 2018.

LI, C.; TONG, R.; TANG, M. Modelling human body pose for action recognition using deep neural networks. **Arabian Journal for Science and Engineering**, [S.l.], p. 1–12, 2018.

LI, J. et al. Human action recognition based on point context tensor shape descriptor. **Journal of Electronic Imaging**, [S.l.], v. 26, n. 4, p. 043024, 2017.

LI, Y.; LIU, Y.; ZHANG, C. What elements are essential to recognize human actions? **CVPR Workshops 2019**, [S.l.], 2019.

LIBBRECHT, M. W.; NOBLE, W. S. Machine learning applications in genetics and genomics. **Nature Reviews Genetics**, [S.l.], v. 16, n. 6, p. 321–332, 2015.

LIN, T.-Y. et al. Microsoft coco: common objects in context. In: EUROPEAN CONFERENCE ON COMPUTER VISION, 2014. **Anais...** [S.l.: s.n.], 2014. p. 740–755.

LIN, T.-Y. et al. Feature pyramid networks for object detection. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2017. **Proceedings...** [S.l.: s.n.], 2017. p. 2117–2125.

LIU, J. et al. Skeleton-based human action recognition with global context-aware attention lstm networks. **IEEE Transactions on Image Processing**, [S.l.], v. 27, n. 4, p. 1586–1599, 2017.

LIU, J.; KUIPERS, B.; SAVARESE, S. Recognizing human actions by attributes. In: CVPR 2011, 2011. **Anais...** [S.l.: s.n.], 2011. p. 3337–3344.

LIU, M.; LIU, H.; CHEN, C. Enhanced skeleton visualization for view invariant human action recognition. **Pattern Recognition**, [S.l.], v. 68, p. 346–362, 2017.

LIU, M.; YUAN, J. Recognizing human actions as the evolution of pose estimation maps. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2018. **Proceedings...** [S.l.: s.n.], 2018. p. 1159–1168.

LIU, W. et al. A survey of deep neural network architectures and their applications. **Neurocomputing**, [S.l.], v. 234, p. 11–26, 2017.

LOGOTHETIS, N. K.; SHEINBERG, D. L. Visual object recognition. **Annual review of neuroscience**, [S.l.], v. 19, n. 1, p. 577–621, 1996.

LUDL, D.; GULDE, T.; CURIO, C. Simple yet efficient real-time pose-based action recognition. In: IEEE INTELLIGENT TRANSPORTATION SYSTEMS CONFERENCE (ITSC), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p. 581–588.

LUVIZON, D. C.; PICARD, D.; TABIA, H. 2d/3d pose estimation and action recognition using multitask deep learning. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2018. **Proceedings...** [S.l.: s.n.], 2018. p. 5137–5146.

MAASS, W. Networks of spiking neurons: the third generation of neural network models. **Neural networks**, [S.l.], v. 10, n. 9, p. 1659–1671, 1997.

MAJI, S.; BOURDEV, L.; MALIK, J. Action recognition from a distributed representation of pose and appearance. In: CVPR 2011, 2011. **Anais...** [S.l.: s.n.], 2011. p. 3177–3184.

MENG, M. et al. Human-object interaction recognition by learning the distances between the object and the skeleton joints. In: IEEE INTERNATIONAL CONFERENCE AND WORKSHOPS ON AUTOMATIC FACE AND GESTURE RECOGNITION (FG), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. v. 7, p. 1–6.

MITTAL, A.; ZISSERMAN, A.; TORR, P. H. Hand detection using multiple proposals. In: BMVC, 2011. **Anais...** [S.l.: s.n.], 2011. p. 1–11.

MOESLUND, T. B.; HILTON, A.; KRÜGER, V. A survey of advances in vision-based human motion capture and analysis. **Computer vision and image understanding**, [S.l.], v. 104, n. 2-3, p. 90–126, 2006.

MONTEIRO, J. et al. Using scene context to improve action recognition. In: IBEROAMERICAN CONGRESS ON PATTERN RECOGNITION, 2018. **Anais...** [S.l.: s.n.], 2018. p. 954–961.

NI, B.; WANG, G.; MOULIN, P. Rgbd-hudaact: a color-depth video database for human daily activity recognition. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION WORKSHOPS (ICCV WORKSHOPS), 2011., 2011. **Anais...** [S.l.: s.n.], 2011. p. 1147–1153.

NIE, Q. et al. A child caring robot for the dangerous behavior detection based on the object recognition and human action recognition. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND BIOMIMETICS (ROBIO), 2018., 2018. **Anais. . .** [S.l.: s.n.], 2018. p. 1921–1926.

PHAM, H.-H. et al. Skeletal movement to color map: a novel representation for 3d action recognition with inception residual networks. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2018., 2018. **Anais. . .** [S.l.: s.n.], 2018. p. 3483–3487.

PHAM, H. H. et al. Spatio-temporal image representation of 3d skeletal movements for view-invariant action recognition with deep convolutional neural networks. **Sensors 2019**, [S.l.], 2019.

POTTER, M. C. Short-term conceptual memory for pictures. **Journal of experimental psychology: human learning and memory**, [S.l.], v. 2, n. 5, p. 509, 1976.

PRECHELT, L. Automatic early stopping using cross validation: quantifying the criteria. **Neural Networks**, [S.l.], v. 11, n. 4, p. 761–767, 1998.

QIU, D. et al. Regression testing of web service: a systematic mapping study. **ACM Computing Surveys (CSUR)**, [S.l.], v. 47, n. 2, p. 21, 2015.

RAO, J. et al. A mobile outdoor augmented reality method combining deep learning object detection and spatial relationships for geovisualization. **Sensors**, [S.l.], v. 17, n. 9, p. 1951, 2017.

REDMON, J. **2016. darknet**: open source neural networks in c. 2013.

REDMON, J. et al. You only look once: unified, real-time object detection. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2016. **Proceedings. . .** [S.l.: s.n.], 2016. p. 779–788.

REDMON, J.; FARHADI, A. Yolo9000: better, faster, stronger. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2017. **Proceedings. . .** [S.l.: s.n.], 2017. p. 7263–7271.

REDMON, J.; FARHADI, A. Yolov3: an incremental improvement. **arXiv preprint arXiv:1804.02767**, [S.l.], 2018.

REN, S. et al. Faster r-cnn: towards real-time object detection with region proposal networks. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 2015. **Anais. . .** [S.l.: s.n.], 2015. p. 91–99.

RODRIGUEZ, M. D.; AHMED, J.; SHAH, M. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In: CVPR, 2008. **Anais. . .** [S.l.: s.n.], 2008. v. 1, n. 1, p. 6.

SÁNCHEZ, J. et al. Image classification with the fisher vector: theory and practice. **International journal of computer vision**, [S.l.], v. 105, n. 3, p. 222–245, 2013.

SARAFIANOS, N. et al. 3d human pose estimation: a review of the literature and analysis of covariates. **Computer Vision and Image Understanding**, [S.l.], v. 152, p. 1–20, 2016.

SCHNEIDER, G. E. Two visual systems. **Science**, [S.l.], 1969.

SCHOLL, B. J. Object persistence in philosophy and psychology. **Mind & Language**, [S.l.], v. 22, n. 5, p. 563–591, 2007.

SHAHROUDY, A. et al. Multimodal multipart learning for action recognition in depth videos. **IEEE transactions on pattern analysis and machine intelligence**, [S.l.], v. 38, n. 10, p. 2123–2129, 2015.

SHAPLEY, R.; TOLHURST, D. Edge detectors in human vision. **The Journal of physiology**, [S.l.], v. 229, n. 1, p. 165–183, 1973.

SHARMA, G.; JURIE, F.; SCHMID, C. Expanded parts model for human attribute and action recognition in still images. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2013. **Anais...** [S.l.: s.n.], 2013. p. 652–659.

SHEN, L. et al. Scaling human-object interaction recognition through zero-shot learning. In: IEEE WINTER CONFERENCE ON APPLICATIONS OF COMPUTER VISION (WACV), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 1568–1576.

SHOTTON, J. et al. Real-time human pose recognition in parts from single depth images. In: COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2011 IEEE CONFERENCE ON, 2011. **Anais...** [S.l.: s.n.], 2011. p. 1297–1304.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, [S.l.], 2014.

SOOMRO, K.; ZAMIR, A. R.; SHAH, M. Ucf101: a dataset of 101 human actions classes from videos in the wild. **arXiv preprint arXiv:1212.0402**, [S.l.], 2012.

SUN, M.; KOHLI, P.; SHOTTON, J. Conditional regression forests for human pose estimation. In: COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2012 IEEE CONFERENCE ON, 2012. **Anais...** [S.l.: s.n.], 2012. p. 3394–3401.

TAYLOR, G. et al. The development of contour processing: evidence from physiology and psychophysics. **Frontiers in psychology**, [S.l.], v. 5, p. 719, 2014.

THORPE, S.; FIZE, D.; MARLOT, C. Speed of processing in the human visual system. **nature**, [S.l.], v. 381, n. 6582, p. 520, 1996.

TOSHEV, A.; SZEGEDY, C. Deeppose: human pose estimation via deep neural networks. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2014. **Proceedings...** [S.l.: s.n.], 2014. p. 1653–1660.

VASCONEZ, J. P.; SALVO, J.; AUAT, F. Toward semantic action recognition for avocado harvesting process based on single shot multibox detector. In: IEEE INTERNATIONAL CONFERENCE ON AUTOMATION/XXIII CONGRESS OF THE CHILEAN ASSOCIATION OF AUTOMATIC CONTROL (ICA-ACCA), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 1–6.

WANG, S.; ZHOU, G. A review on radio based activity recognition. **Digital Communications and Networks**, [S.l.], v. 1, n. 1, p. 20–29, 2015.

WANG, T. et al. Deep contextual attention for human-object interaction detection. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2019. **Proceedings...** [S.l.: s.n.], 2019. p. 5694–5702.

XIAO, C. M.; MA, T. L.; XIA, R. B. An edge detection algorithm based on human visual system. In: ADVANCED MATERIALS RESEARCH, 2013. **Anais...** [S.l.: s.n.], 2013. v. 760, p. 1519–1523.

XIN, M. et al. Learning discriminative action and context representations for action recognition in still images. In: IEEE INTERNATIONAL CONFERENCE ON MULTIMEDIA AND EXPO (ICME), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p. 757–762.

YAN, S. et al. Multibranch attention networks for action recognition in still images. **IEEE Transactions on Cognitive and Developmental Systems**, [S.l.], v. 10, n. 4, p. 1116–1125, 2017.

YAN, W.; GAO, Y.; LIU, Q. Human-object interaction recognition using multitask neural network. In: INTERNATIONAL SYMPOSIUM ON AUTONOMOUS SYSTEMS (ISAS), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p. 323–328.

YANG, Y.; RAMANAN, D. Articulated human detection with flexible mixtures of parts. **IEEE transactions on pattern analysis and machine intelligence**, [S.l.], v. 35, n. 12, p. 2878–2890, 2013.

YAO, B.; KHOSLA, A.; FEI-FEI, L. Combining randomization and discrimination for fine-grained image categorization. In: CVPR 2011, 2011. **Anais...** [S.l.: s.n.], 2011. p. 1577–1584.

YU, G.; LIU, Z.; YUAN, J. Discriminative orderlet mining for real-time recognition of human-object interaction. In: ASIAN CONFERENCE ON COMPUTER VISION, 2014. **Anais...** [S.l.: s.n.], 2014. p. 50–65.

ZHANG, H.-B. et al. A comprehensive survey of vision-based human action recognition methods. **Sensors**, [S.l.], v. 19, n. 5, p. 1005, 2019.

ZHOU, B. et al. Places: a 10 million image database for scene recognition. **IEEE transactions on pattern analysis and machine intelligence**, [S.l.], v. 40, n. 6, p. 1452–1464, 2017.

ZHU, H. et al. Yotube: searching action proposal via recurrent and static regression networks. **IEEE Transactions on Image Processing**, [S.l.], v. 27, n. 6, p. 2609–2622, 2018.

ZHU, H.; HU, J.-F.; ZHENG, W.-S. Learning hierarchical context for action recognition in still images. In: PACIFIC RIM CONFERENCE ON MULTIMEDIA, 2018. **Anais...** [S.l.: s.n.], 2018. p. 67–77.

ZHU, H.; VIAL, R.; LU, S. Tornado: a spatio-temporal convolutional regression network for video action proposal. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2017. **Proceedings...** [S.l.: s.n.], 2017. p. 5813–5821.

ZHU, J. et al. Action machine: rethinking action recognition in trimmed videos. **arXiv preprint arXiv:1812.05770**, [S.l.], 2018.

ZHUANG, B. et al. Hcvrd: a benchmark for large-scale human-centered visual relationship detection. In: THIRTY-SECOND AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE, 2018. **Anais...** [S.l.: s.n.], 2018.