REAL TIME ANALYSIS OF A PATIENT'S STATUS

Alfonso Del Gaizo

Department of Computer Science, University of Salerno, Via
Giovanni Paolo II 132, Fisciano (SA), Italy

SUMMARY

1.INTRODUCTION

The system that will be described in the following chapters has the purpose of monitoring a patient and detecting a sudden change of state, to then report this anomaly through an alarm. The idea came from the need of hospital companies to have a system capable of supporting the work of nurses.

The control system, described in this document, uses artificial intelligence and machine learning techniques which, through images obtained from the surveillance cameras of a hospital room where the patient rests, can recognize if the latter passes from a state of calm to a state of agitation or anomalous.

In this paper, there will be a first part dedicated to the theory of data science, IoT and machine learning. Subsequently ,all the technologies used for the implementation of the system and the motivation of certain operational choices will be illustrate, such as for example the efficient management of the video stream coming from video cameras. Finally the practical part of the project will be discussed starting from the details related to the libraries used up to the description of the rules and the functions defined for the identification of the changes of the status.

2 STATE OF ART

2.1 Object Detection Tools

There are many tools for object detection, they use different approaches based on their final purpose of use. These tools are used can be applied in many fields, so it is useful to know some of them to get an overview of their potential.

The first tool described is YOLO. It is an extremely fast real time multi object detection algorithm

The algorithm applies to a neural network to an entire image. The network divides the image into an S x S grid and comes up with bounding boxes, which are boxes drawn around images and predicted probabilities for each of these regions (Darren Eng, YOLO: Real Time Object Detection,2016 .https://github.com/pjreddie/darknet/wiki/YOLO:-Real-Time-Object-Detection) .The method used to up with these probabilities is logistic regression. The bounding boxes are weighted by the associated probabilities. For class prediction, independent logistic classifiers are used. Another tool is Google Cloud AutoML Vision Object Detection, it enables developers to train custom machine learning models that are capable of detecting individual objects in a given image along with its bounding box and label and includes the following features : Object Localization than detects multiple objects in an image and provides information about the object and where the object was found in the image , and API/UI that provides an API and custom user interface for importing your dataset from a Google Cloud Storage hosted CSV file and training images, for adding and removing annotations from imported images, for training and reviewing model evaluation metrics, and for using your model with online prediction (Google Cloud AutoML:custom machine learning models,2019.https://cloud.google.com/automl/?hl=it).

Another tool is Kairos, it integrates Face Recognition via our cloud API, or host Kairos on your own servers for ultimate control of data, security, and privacy. Kairos feature are : *Face Detection* finds human faces in photos and images, *Face*

Identification that searches for face matches, *Face Verification* that searches for someone's face, *Gender Detection* detects gender of each face found (female or male), *MultiFace Detection* that detects individuals, crowds, audiences and groups, *Age Detection* finds human faces in photos and images, *Facial Coordinates* that detects size (pitch, roll, yaw and key landmarks), *AntiSpoof Detection* ensures security by checking the liveness of faces, *Diversity Recognition* understand more about the diversity of human face (Kairos, Recognize People The Way You Want,2019.https://www.kairos.com).
Finally there is AWS Rekognition which will be deepened in the following chapters.

The tools described are different from each other. Among the various differences there are: the objects identified in the images, the recognition models used, training phases. Yolo is a tool that uses a pre-trained convolutional network capable of recognizing at least 20 objects (people, bikes, cars, dogs, cats, etc.), and also returns the boxes necessary to identify the recognized objects in the space. The main disadvantage of using YOLO is an expensive training phase because it requires a large dataset of images and many iterations only to have a reasonable confidence value for a new object to be recognized.To reduce training time, YOLO gives the possibility to activate training via NVIDIA GPU (a cumbersome alternative would be to use GPUs in the cloud if you do not have any performing components). As for the AWS Rekognition and Google AutoML tools, they are very similar. They allow the recognition of multiple objects in space, as the convolutional networks used are trained on millions of images. The only significant difference between the two tools lies in facial recognition, in fact Amazon Rekognition would seem much more precise in recognizing details related to emotions and the characters of the face. As for the AWS Rekognition and Google AutoML tools, they are very similar. They allow the recognition of multiple objects in space, as the convolutional networks used are trained on millions of images. The only significant difference between the two tools lies in facial recognition, in fact Amazon Rekognition would seem much more precise in recognizing details related to emotions and the characters of the face. Precisely for this important detail it was decided to use the Amazon recognition tool to perform the recognition of emotions and face within the space.

3. EDCAR

The framework, named Elements and Descriptors of Context and Action Representations (EDCAR), enables the representation of the relevant elements, general descriptors of the context, and actions that have been taken, including the definition of action compositions and sequences, in order to monitor and recognize abnormal situations. EDCAR and the associated system also support video summarization of relevant scenes, providing an inference engine to handle complex queries.

Video event understanding is the translation process of low-level video contents into high-level semantic concepts. In this context, the concept of video sequence represents the key point for classifying low-level contents into their semantic interpretation. Indeed, it might happen that contents appearing in different video sequences yield different semantic interpretations. Moreover, the definition of action in the context of human activity recognition provides the second key point in the event recognition process. Such a definition considers an action as a set of gestures, where the term "set" does not impose particular restrictions on how the action is interpreted, how its gestures are related, or how actions depend on each other.

A fundamental challenge in the real-time contexts is the capability to recognize actions when they occur. Moreover, in order to support the recognition in complex scenarios, contextualized into specific environments, there is the need of customizing what has to be detected. One way to do this is to enable the modeling of knowledge concerning target scenarios. However, general-purpose knowledge representation frameworks provide too basic mechanisms, which make it difficult to model complex target scenarios through simple rules. This led us to define a new knowledge representation framework, named EDCAR, which enables the modeling of objects or actors, context, and actions of the scenario to be detected by means of ECR (Elements of Context Representations), GCD (General Context Descriptors) , and AR (Action Representations), respectively. In particular, in order to define such scenarios, it has been necessary to address 235 the concept of an action composed of more elementary ones, and the definition of action sequences. A scene instance is described by relevant elements of context, which collect data about the points of interest in the recorded scene, including their correlations. GCDs can be related to ECRs and describe the whole environment, such as geographical, historical, emotional or biological information. By defining target scenarios through the EDCAR framework it is possible to understand current events through the stored knowledge, by reducing events to simple ones, and by instantiating elements, descriptors of context, and actions, with actual data (filling the slots); and reason about events and supply missing information in occurred events by making inferences on the instantiated data (Caruccio L., Polese G., Tortora G.,Iannone D. , A Knowledge Representation Framework to Enhance Automatic Video Surveillance, Department of Computer Science, University of Salerno).

The EDCAR framework is composed of seven layers as described in the following.

- *Environmental Layer*, collects data from a set of video surveillance devices, such as cameras, microphones, thermal sensors, and so forth;
- *Frame Layer*, analyze frames in order to identify elements within them, and to extract useful information instantiating ECR, GCD, and AR forms.
- *ECR Layer*, collects all relevant elements of context generated through the analysis on frames in the previous layer and the ECRs.
- *AR Layer*, collects all actions involving ECRs, based on the scene interpretation and the ARs.
- *GCD Layer*, collects all the contextual information according to the GCDs. Such information is generated by the analysis of both frames in the frame layer, which also involves ECRs, and the data collected through Information Layer, described next.
- *Information Layer*, collects all useful information to characterize the environment, such as statistical, geographical, and / or historical data, using external information sources, statistics, or sensor data.
- *Current Scenario Layer*, relates information about ARs and GCDs instantiated in the previous layers, and determines a current scenario in- position.

4. CONTEXT

This chapter describes the technologies and theory to support the realization of a patient status analysis project. This will be followed by the deepening of the terms IoT , machine learning and artificial intelligence to then complete the excursus with a brief introduction and description of Amazon Web Service.

4.1 Internet of Things (IoT)

The term IoT is used for the first time by Kevin Ashton, a researcher at MIT (Massachusetts Institute of Technology), where the standard for RFID and other sensor was found. But even if the term is new, we talk about these concepts for a long time, essentially from the birth of the internet and the semantic web.
With Internet of Things we mean a set of technologies that allow you to connect any type of device to the internet. The purpose of this type of solution is basically to monitor and control and transfer information and then perform subsequent actions(Margaret Rouse,What is internet of things,2016,https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT).
In the city environment, for example, a detector located in a street can check the street lights and indicate if the lamp works, but the same light could, if properly equipped, also report information on the quality of the air or on the presence of people. The Internet of Things finds more and more consensus and is increasingly an opportunity for development. The number of connected devices is increasing, and there is a strong trust in Italy in the more consolidated IoT technologies and resistance to try the most innovative Internet of Things.

The evolution of the Internet has extended the Internet itself to real objects and places which can now interact with the network and transfer data and information. The object interacts with the surrounding world, as it is endowed with "intelligence", that is, it retrieves and transfers information between the internet and the real world.

In this way an "electronic identity" can be given to everything that forms the world around us, through, for example, RFID (Radio frequency identification) and other technologies (such as the QR code).

The phase that can be defined as pre-Internet of Things is represented by “simple” sensors: devices capable of performing data collection in an increasingly precise and targeted manner according to specific application areas (equipment dedicated to detecting data related to the temperature of environments, to movement of vehicles, air quality, noise level of certain environments or the presence of certain substances. In this phase it is correct to speak of sensors that detect information and transform it into digital data. this pre-IoT phase the network connection: these are devices that in different forms and modes are interrogated “manually” or with an organization of the data collection not supported by a network. The transition from sensors to the Internet of Things is made up of a network connection. In order to work, the IoT needs to collect and store a large amount of data. But what good are all these data collected? To make the internet of things work properly so that it is really useful to us, it is important to process collecting and analyzing large volumes of real time data (for example from sensors, traffic lights, and any connected IoT device), and in the company to improve safety and productivity, both in any field and for any type of connected object. For this there is a need for integrated systems between big data, nosql database and IoT data.

The Internet of Things is one of the new frontiers, by now consolidated, of using the internet. No longer are people or "legal persons", businesses, recognizable on the Internet, but things can be. Things, objects, tools that acquire intelligence, or the ability to detect information and communicate it. The Internet of Things is a real New Internet precisely because it opens up prospects that were once unimaginable, in which objects take an active role thanks to being online and sending and receiving data on the network.

The main areas of application of the Internet of Things (both for end consumers like me and you, for companies and manufacturing) are represented by those contexts in which there are "things" that can "talk" and generate new information such as for example: smart home, home automation, smart buildings, industrial monitoring, robotics, automotive industry, automotive, self driving car smart health, healthcare, the biomedical world, all areas of telemetry, all areas of surveillance and security,smart city, smart mobility,new forms of digital payment through objects and other.

4.2 Amazon Web Service (AWS)

Amazon Web Service is one of the most complete and managed cloud platform , offering more than 165 comprehensive data center services globally. Millions of customers, including the fastest growing start ups and the largest companies and industry-leading government agencies, rely on AWS to power their facilities, become more agile and cost the costs.

AWS provides services for a wide range of applications including processing, storage, databases, networks, analysis, machine learning and artificial intelligence, Internet of Things (IoT) , security and application development, distribution and management.

In addition to the largest range of services , AWS also offers the deepest features within them. For example: Amazon EC2 offers more types and sizes of calculation instances than any other provider, including the most powerful GPU instances for machine learning workloads , also has twice the database services of anyone else, with 11 relational and non-relational databases. Furthermore has the highest number of container execution methods , with Amazon Container Service, Amazon Elastic Container Service and Amazon Web Service Fargate.

This large selection of services and deep features make the migration of existing apps to the cloud simpler, faster and cheaper, and allows you to design anything you can imagine.

AWS offers a large global footprint. No other cloud service provider offers so many operational regions , with 69 availability zones spread across 22 geographic regions worldwide, with another 9 availability zones and 4 additional regions already announced for the near future.

AWS regions provide multiple physically separate and isolated zones of availability that are connected via highly redundant, low latency and high throughput networks. These availability zones offer AWS customers an easier and effective way to design and execute applications and databases, increasing their availability, fault tolerance and scalability compared to infrastructures with a single traditional data center or multiple data centers. The AWS region model has been recognized by Gartner as the recommended approach to run business applications that require high availability.

4.3 Machine Learning

Machine Learning is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead,it is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. (Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. Artificial Intelligence in Design ,1996. Springer, Dordrecht. pp. 151–170).The different techniques of learning and developing algorithms give rise to as many possibilities of use that widen the field of application of machine learning making a specific definition difficult. However, we can say that when we talk about machine learning we talk about different mechanisms that allow an intelligent machine to improve its capabilities and performance over time. The machine, therefore, will be able to learn to perform certain tasks by improving, through experience, its own abilities, its own answers and functions. At the base of the automatic learning there are a series of different algorithms that, starting from primitive notions, will know how to take a specific decision rather than another or carry out actions learned over time.

Today we have different ways of learning, all effective, which differ not only in the algorithms used, but above all for the purpose for which the machines are made. Depending on the type of algorithm used to enable machine learning, that is, depending on the way the machine learns and accumulates data and information, three different machine learning systems can be divided: supervised learning , unsupervised learning and reinforcement learning.

The three learning models are used in different ways depending on the machine on which one must operate.

Supervised learning consists of providing the machine's computer system with a series of specific and coded notions, that is, models and examples that allow to build a real database of information and experiences. In this way, when the machine is faced with a problem, it will only have to draw on the experiences included in its system, analyze them, and decide which response to give based on already coded experiences. Unsupervised learning requires that the information inserted in the machine is not coded, the machine has the possibility to draw on certain information without having any example of its use and, therefore, without having knowledge of the results expected depending on the choice made. Reinforcement learning is the most complex learning system, which requires the machine to be equipped with systems and tools that improve learning and, above all, understand the characteristics of the surrounding environment.

One of the main characteristics of machine learning is its close correlation with other branches of computer science, statistics, optimization and many other areas of modern intelligent science.

Data mining is a form of learning, but limited to unsupervised learning only. In fact, data mining aims at extracting information and data to improve machine knowledge. Very often, even if the use of techniques can be similar, what differentiates the branches related to machine learning, artificial intelligence, data mining and other intelligent systems, is the purpose for which these systems were created it can be said that unsupervised learning is an integral part of both machine learning and data mining, the main difference is precisely in the aim: data mining aims exclusively to improve the machine through constantly new knowledge while machine learning has as its purpose that of an ever deeper learning, which takes into consideration not only the possibility of new knowledge, but also that of reproducing the knowledge carried out, with the aim of an ever more advanced improvement of the machine for ever more specific uses.

4.4 Artificial Intelligence

Artificial Intelligence is a branch of computer science that allows the programming and design of both hardware and software systems that allow to equip machines with certain characteristics that are typically considered human, such as, for example, visual, spatio-temporal and decision-making perceptions . That is to say, not only of intelligence understood as the capacity for calculation or knowledge of abstract data, but also and above all of all those different forms of intelligence that are recognized by Gardner's theory (Gilman, Lynn. "The Theory of Multiple Intelligences",2012. Indiana University) and that range from spatial intelligence to social intelligence, from the kinesthetic to the introspective one. An intelligent system, in fact, is created by trying to recreate one or more of these different forms of intelligence which, although often defined as simply human, can actually be traced back to particular behaviors reproducible by some machines. The use of neural networks and algorithms capable of reproducing the typical reasoning of human beings in different situations, have enabled intelligent systems to increasingly improve the different abilities of behavior. In order to achieve this, the research focused not only on the development of ever new algorithms, but above all on increasingly numerous algorithms, which could imitate different behaviors according to environmental stimuli. These complex algorithms, inserted into intelligent systems, are therefore able to "make decisions" that is to make choices according to the contexts in which they are inserted. In the case of algorithms connected to intelligent vehicle systems, for example, a car without a driver can decide, in case of danger, whether to steer or brake depending on the situation, that is, according to whether the information sent by the various sensors allow to calculate a higher percentage of safety for the driver and passengers by braking or turning.

Artificial intelligence has been able to progress more when specific algorithms have been created, able to improve the behavior of the machine (understood as the ability to act and make decisions) that can thus learn through experience, just like humans . Developing algorithms that are able to learn from their mistakes is essential for creating intelligent systems that operate in contexts for which programmers cannot a priori foresee all the possibilities of development and the contexts in which the system operates. Through machine learning, therefore, a machine is able to learn to perform a specific action even if such action has never been programmed among the possible actions.

4.4.1 Artificial Intelligence and Health Care

Artificial Intelligence is destined to revolutionize the healthcare world as well. From a study by Frost & Sullivan (Frost&Sullivan, Artificial Intelligence - The Cognitive Area,2019. .https://ww2.frost.com/research/visionary-innovation/artificial-intelligence-cognitive-era) the market of Artificial Intelligence for Healthcare should reach 6.6 billion dollars by 2021, with a growth rate of 40%. Clinical applications of the AI will strengthen the processes of medical imaging diagnosis and support patient care activities. Also according to the report, Artificial Intelligence has the potential to improve revenue in the health sector by 30-40% and, at the same time, to reduce treatment costs by up to 50%. The most obvious application of Artificial Intelligence to the healthcare sector is in data management. A whole series of data concerning the patient's medical history must be collected and stored: X-rays, laboratory results, reports on the patient's pathologies, family history and genome. Likewise, data relating to the health of the patient must be collected, stored and managed: health "diaries"; reports processed by the patient himself through a device; home diagnostics, data from relatives, service communities and third parties. It can be used to diagnose and define treatment plans. For example, IBM has launched a special program for oncologists that can provide specialists with evidence-based treatment options. This particular artificial intelligence-based system has an advanced ability to analyze the meaning and context of structured and unstructured data contained within clinical notes and reports, data that can become decisive for choosing the path of patient care. Then, by combining the characteristics obtained from the patient's file with the clinical expertise, but also with external research and other data, the program is able to identify the potential treatment plan. IBM has also launched an algorithm called Medical Sieve. It is an exploratory and long-term project to build the next generation of "cognitive assistants" with analytical and reasoning skills and with a wide range of medical knowledge. This system wants to help make decisions in the field of radiology and cardiology. It is in fact able to analyze radiographs to identify problems quickly and reliably. In the future radiologists will take care of the most complicated cases, where human supervision is necessary, while routine ones can be carried out by the AI.

Another example is a project of a medical startup in San Francisco has developed a virtual nurse. This is an avatar that helps doctors and patients monitor and manage their health in a better way. The avatar, who has a smiling face and a pleasant voice, is an interface that uses machine learning to support chronic patients between visits to the doctor and the other. A similar solution is provided by AiCure, an app supported by the National Institutes of Health, which uses the smartphone's webcam and AI to independently confirm that patients are following the doctor's instructions. It turned out to be useful for those seriously ill, those who tend to disregard the doctor's instructions and for those who participate in clinical trials.

4.5 Link Between Topics

Some specific topics have been introduced in order to have an overview of the context behind the development of the monitoring system. We can say that these arguments are linked for many reasons. As a first argument, the general IoT counting was discussed, clearly it is fundamental because we can use techniques for collecting frames deriving from the videos of surveillance images. The goal of the software is to make predictions on these images and it is precisely at this juncture that the concept of Machine Learning and Artificial Intelligence is connected. These concepts are fundamental because through them it has been possible to train convolutional models that allow to make predictions on the images obtained by surveillance cameras. Finally the AWS cloud platform was introduced (through a brief description), it is important as it allows the recognition of objects in space (more specifically the details of the face) by offering a tool called AWS Rekognition which precisely recalls the concepts of Artificial Intelligence described. In fact, this tool provides a convolutional network which, through machine learning algorithms, is able to recognize multiple objects within the images.

5 PREDICTIVE MODEL AWS REKOGNITION

In this chapter we describe an AWS service called Rekognition for the facial analysis of the subjects and then there will be an in-depth analysis on a specific prediction model used by the recognition tool based on neural networks.

It is important to introduce this argument in detail as this tool is the operational heart of the detection system. AWS Rekognition allows you to have all the necessary elements to be able to check the rules to identify a patient's change of status. This tool also interfaces with a second fundamental part: the management of the video stream. In fact it is thanks to the optimal management of the video stream that AWS Rekognition gets all the images needed to make predictions on faces.

5.1 AWS Rekognition

Amazon Rekognition is a service that facilitates the addition of effective visual analysis to applications. Rekognition Image allows you to easily create powerful applications to search, verify and organize millions of images. Rekognition Video allows you to extract the context from saved videos or live streams based on movement, helping you to analyze them.

Amazon Rekognition is based on the same deep, proven and highly scalable learning technology developed by Amazon computer vision experts, which enables billions of images and videos to be analyzed daily and requires no machine learning experience for use. Amazon Rekognition is a simple and easy-to-use API that can quickly analyze any image or video file stored on Amazon S3. Amazon Rekognition continuously learns from new data and we continue to add new labels and facial recognition features to the service (Amazon Rekognition, Intelligent Analysis of Images and Videos for applications, 2019. https://aws.amazon.com/it/rekognition/).

It can perform real-time analysis on Amazon Kinesis Video Streams videos and analyze the images that are uploaded to Amazon S3. For large processes, you can use AWS Batch to analyze thousands of images and videos at once. Face-based user verification can be easily integrated into new or existing applications. This is a

process that uses only one API , it also offers constant response times regardless of the volume of requests made. Application latency remains constant, even if the volume reaches tens of millions of requests.
Among the services offered there are: Detection of objects, scenes and activities, Facial recognition, Detection of unsafe content, Text in images.

5.2 Predictive Model

Amazon Rekognition uses deep learning models to perform face detection and search for faces in collections. It continues to improve the accuracy of its models based on customer feedback and advances in deep learning research. These improvements are provided as model updates. For example, with version 1.0 of the model, IndexFaces can index the 15 largest faces in an image. Later versions of the model allow IndexFaces to index the 100 largest faces in an image. When creating a new collection, it is associated with the most recent version of the template. To improve accuracy, the model is updated periodically.

Specifically, this service uses a convolutional network that will be analyzed in detail in the next paragraph, this because one needs to understand fully how this predictive model works and how it increases its performance and the accuracy of its predictions.

5.2.1 Convolutional Neural Network

Convoluted Neural Networks, or ConvNet (CNN) are one of the most widely used Deep Learning algorithms in computer vision today, and are used in many fields, from autonomous cars to drones, from medical diagnosis to support and treatment for the visually impaired.

What is a convolution? It’s a type of artificial feed-forward neural network in which the pattern of connectivity between neurons is inspired by the organization of the animal visual cortex, whose individual neurons are arranged in such a way as to respond to the overlapping regions that tile the visual field. The convolution, mathematically speaking, means "to slide" one function (blue) over another (red), effectively "mixing" them together. The result will be a function (green) that represents the product of the two functions.

Convolutional neural networks function like all neural networks: an input layer, one or more hidden layers, which perform calculations using activation functions, and an output layer with the result (Ronan Collobert e Jason Weston, A Unified

Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning, in Proceedings of the 25th International Conference on Machine Learning, New York, NY, USA, ACM, 2008.). Convolutions are the difference , each layer hosts what is called a "feature map", which is the specific feature that the nodes are looking for. In the example below, the first layer could be used to encode the diagonal lines, then "slide" a special filter on the image, and multiply it by the scalar product for the area below.
The result will be another matrix, slightly smaller than the original image, or of the same size if zero padding is used, called feature map. Each layer of neurons can possibly be applied more filters, thus generating more feature maps.Typically each convolutional layer is followed by one of Max-Pooling3, gradually reducing the size of the matrix, but increasing the level of "abstraction". It then passes from elementary filters, such as vertical and horizontal lines, to more and more sophisticated filters, which can for example recognize the headlights, the windscreen ; up to the last level where it is able to distinguish a car from a truck .
So in summary, convolution is a mathematical operation that describes a rule for mixing two functions or two distinct pieces of information. A starting image can be transformed obtaining a particular effect by applying to it a convolution mask, that is a series of numbers often represented by a matrix.

6 VIDEO FLOW MANAGEMENT

In this chapter we discuss the approach used to manage the stream analysis of live videos from recording devices. More specifically, we describe the approach used to break down the frames belonging to live recordings and how the Rekognition tool works to quickly analyze the input video stream frames in real time.

6.1 Video frame decomposition

Data flows are data generated continuously from thousands of data sources, which generally forward records of data simultaneously and in small doses (in the order of kilobytes). Data flows are composed of various types of data, such as log files generated by customers using Web applications or on mobile devices, purchases made on e-commerce sites, events within video games, information coming from social media networks, financial transaction data, geolocation and telemetry services related to connected devices , all this information must be processed sequentially and incrementally, record by record or based on dynamic time intervals, and used for a wide range of analysis operations, such as correlation, aggregation, filtering and sampling. Processing in streams is advantageous in most cases where new data is generated continuously and dynamically. It applies to most sectors and use cases related to Big Data. In general, companies start with very simple applications, such as collecting system logs and rudimentary calculations as minimax algorithms. Later on, these applications evolved to become sophisticated calculations almost in real time. Initially, they are used to process data streams so as to produce fairly essential reports and perform simple actions in response, for example to generate alarms when a given measurement exceeds a predefined threshold. Soon, however, the applications begin to deal with more sophisticated data analysis, for example for the application of machine learning algorithms and the extraction of in-depth information starting from the available data. Progressively complex algorithms on flows and events are applied, for example time intervals in which to identify the most recent successful films, making the analyzes more and more detailed.

In the patient status analysis project the real-time video is taken from the surveillance cameras installed in the hospital room.

There are many considerations to be made about the video to be analyzed in real time; first of all there is the need to understand how to break down the stream of video streaming to be able to then analyze the images frame by frame. During the development of the system, more decomposition parameters were tested, in order to have enough frames to analyze and a time latency that would allow the Rekognition tool to make its predictions with a subsequent and quick analysis of the data in output.

After many analyzes then we came to a conclusion in which the frames derived from the video stream had to be created every second, so as to be able to give all the time necessary for the predictions and for the control of the rules. The operation of this frame-by-frame division starts precisely from the acquisition of the video from the device specified in input, subsequently there is a phase of decomposition of the video in frame time by time, clearly it is not enough just the division into frames to have images that can be easily analyzed. In fact the next step goes to operate on the decomposed image to be able to resize it and make it usable for final processing.

Also in this step a very precise study was carried out on how the image had to be resized, in fact it was possible to notice that an image too large in terms of size did not allow a fluidity of analysis as there were problems of storage in the destination buffer , too large images are not essential for a correct prediction. On the contrary, an image with excessive scaling could not guarantee a correct prediction of both the subject in the space and of the details of the face for the detection of emotions. At the end of the resizing process there is therefore buffer management, which is also very delicate as each resized image is first encoded in a string of bytes and then saved inside the buffer. After loading, the image encoded in byte streams is processed by the Rekognition convolutional network to then generate the predictions necessary to be able to perform checks and set the rules.

7 ARCHITECTURE

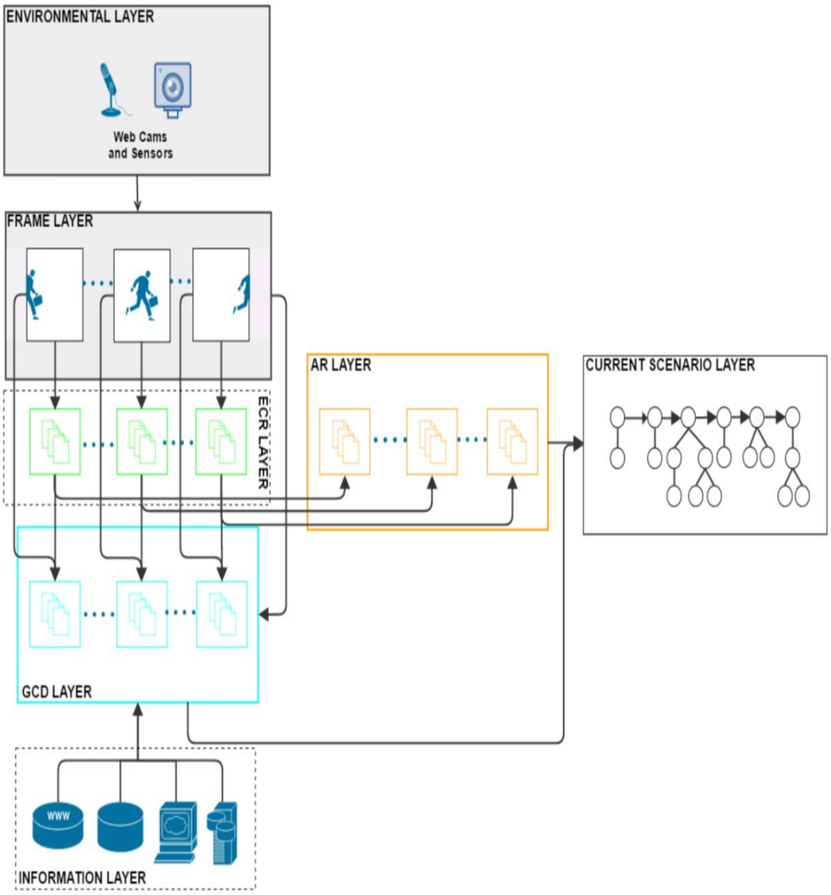

Fig.1 Framework Architecture

7.1 Framework Architecture

In this chapter the architecture framework will be analyzed in detail. There will be a description of the elements that are present within the analyzed context with the relative properties that characterize it and the role taken on. Subsequently the actions that the element carries out within the context will be described, with all the details relating to the type of action and the consequences. Finally, the reference scenario of the patient monitoring system with its architecture will be outlined.

7.1.1 *ECR (Elements of Context Representation)*

- *Type*. A scene may contain different kinds of elements. Each element of context may assume as value of the type feature one of the basic types Actor or Object, or a specialization of them. A specialization hierarchy can be constructed for both basic types.
- *Properties*. An element holds specific properties. The specification of properties in the model refers to technical information that can be caught through devices and feature extraction algorithms.
- *Entry Condition*. Some element have to satisfy an Entry Condition in order to be considered relevant in the scene to be distinguished from others.

PATIENT

Features	**Description**

Type	PERSON
Properties	Location : XValue , YValue Emotion: Value Status : Value
Entry Condition	(IS, Calm)

Table 1: An ECR describing the concept of PATIENT

In the table just illustrated there is the definition of the ECR which describes the element within the context. The name of the element is Patient (the only fundamental element of the system), after which the type of the element is specified, and in the case of the patient it is a PERSON. Each element must have some characteristic property, in the case of the PATIENT, it has a position within the analyzed space and an emotion that characterizes its state. This last is also an entry condition, in fact it is assumed that the patient at the beginning of the analysis is in the state of CALM.

7.1.2 *GCD (General Context Descriptor)*

- *Type*. Several context information can be collected according to sources managed in the information layer. For this reason, it is possible to collect several kinds of data, such as biometric, geographic, or statistic data.
- *Properties*. A context information holds specific properties. The specification of properties in the model refers to data that can be caught through tools or modules working on external sources and/or on the identified ECR.
- *Tool/Module.* Independently from its type, context information depends on tools or modules to extract it.
- *ECR Index*. A context information can be connected to at most one ECR.

FACIAL EXPRESSION

Features	**Description**
Type	Biometric Emotions
Properties	Classification : {Angry , Disgusted, Sad,Confused, Fear}
Tool/Module	AWS Rekognition
ECR Index	Actor

Table 2 : A GCD describing Facial Expression

The GCD just described summarizes all the features related to the FACIAL EXPRESSION. It is biometric since it is identified by facial features and therefore by the change in facial expression. Among its properties is the description of the various types of expression that can be classified, namely: angry, disgusted, sad, confused and fear. The module that takes care of the recognition is precisely AWS Rekognition. Finally, it is specified that the facial expression is linked to the actor.

PATIENT MOVEMENT

Features	Description
Type	Movement in Space
Properties	Bounding Box : {Left.Top.Width,Height}
Tool/Module	AWS Rekognition
ECR Index	Actor

Table 3 : A GCD describing Patient Movement

The GCD just described summarizes all the features related to the PATIENT MOVEMENT. It is a movement in space as it is identified by the bounding boxes that describe its trace in space. Among its properties there is the description of the size and position of the bounding boxes: left, top, width, height. The recognition module is always AWS Rekognition. Finally, it is specified that the facial expression is linked to the actor.

7.1.3 *AR (Action Representation)*

- *Type*. An action can be simple or composite. A simple action represents individual events that can be detected by analyzing the properties of elements occurring in it.
- *Elements*. An action involves one or more elements of context. For this reason, through the elements specification it is possible to define the elements on which the action applies.
- *Rule.*Independently from its type, the recognition of an action depends on the rules characterizing it.
- *Effect.* An action can produce an effect on the context. The effect yields the context update that must be carried out when the action occurs.

Action	**Rule**
CHANGE_POS	ActorObject.Location != (GET_POS, ActorObject)
CHANGE_EMOTION	ActorObject.Emotion != (GET_EMOTION, ActorObject)
CHANGE_STATUS	(CHANGE_POS) ^ (CHANGE_EMOTION)

Table 4 : AR Rules of Patient Analysis

CHANGE_POS

Features	Description
Type	Simple
Elements	ActorObject
Rule	ActorObject.Location != (GET_POS, ActorObject)
Effect	(SET_LOC, ActorObject)

Table 5: AR Rules of Movement Analysis

CHANGE_EMOTION

Features	Description
Type	Simple
Elements	ActorObject
Rule	ActorObject.Emotion != (GET_EMOTION, ActorObject)
Effect	(SET_EMOTION, ActorObject)

Table 6: AR Rules of Emotion Analysis

CHANGE_STATUS

Features	Description
Type	COMPOSITE
Elements	ActorObject
Rule	(CHANGE_POS) ^ (CHANGE_EMOTION)
Effect	(SET_STATUS, ActorObject)

Table 7 : AR Composite Rule of Status final Analysis

7.1.4 *Scenario*

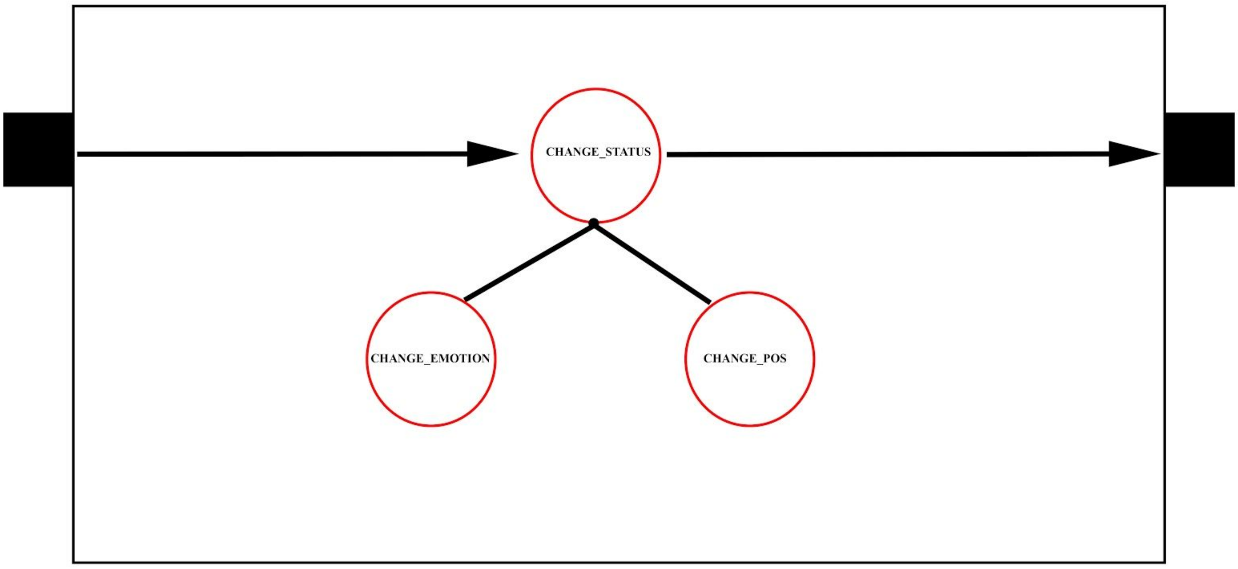

Fig.2 Change Status Analysis Scenario

7.2 Software Architecture

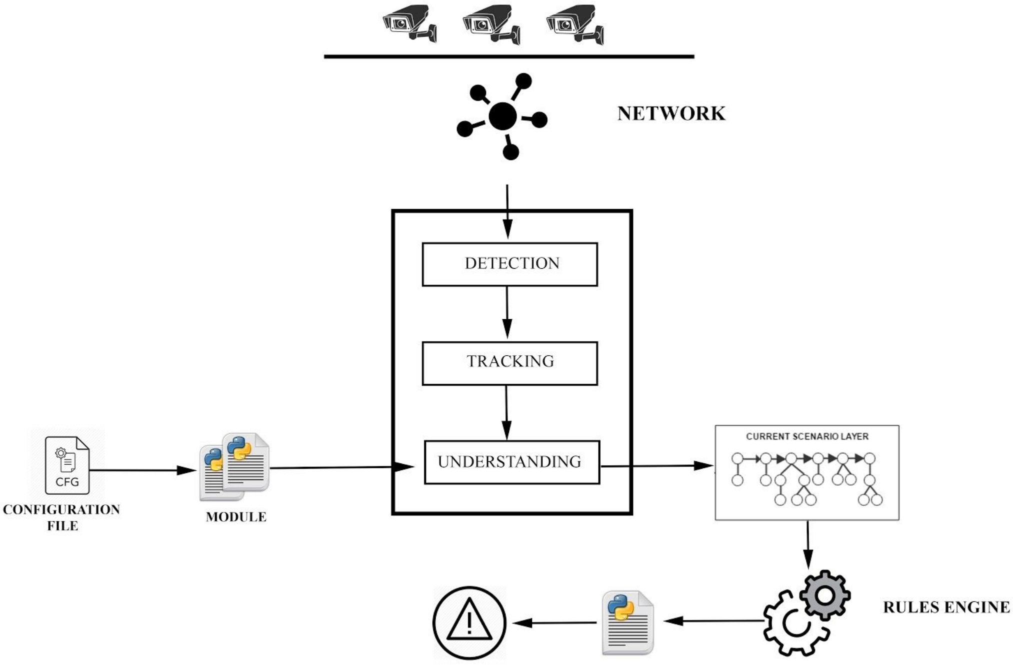

Fig.3 Software Architecture of Analysis System

The input module shown on the top represents a set of cameras and sensors, which can be used for transmitting input information through a network. The core of the system is composed of three main modules: Detection, Tracking, and Understanding.

- *Detection*. The goal of the detection module is to acquire the data, to detect "atomic" events, and to identify elements of context.
- *Tracking*. Object tracking is the process of tracking an object over time by locating its position in every frame of the system (Joshi, K. A., & Thakore, D. G. (2012). A survey on moving object detection and tracking in video surveillance system. International Journal of Soft Computing and Engineering, 2, 44–48). Thus, through the tracking module the system tracks moving elements of context in the images produced by the Detection module. The information on the tracked objects are synthesized by means of simple actions, which are given in output to the Understanding module.
- *Understanding*. The use of sequences/compositions of actions representing 575 relevant situations allows us to effectively analyze different video scenes, and to derive logic consequences to correlate them, also determining whether they contain target events. In particular, this module concerns the understanding at semantic level of the findings from the previous module, involving the elements of the context, through the instance modeling of the current scenario.

These models interact with other system modules that allow you to collect a series of useful information for using the system. In fact the modules are set through a configuration file that specifies all the features that the recognition system must have. In summary, the configuration file specifies the modules to be loaded.
The detection, tracking and understanding modules also interact with the current modeled scenario. Subsequently there is the engine rules that allows to classify the scenario as a target, the rule that allows the recognition of the scenario concerned is applied and finally there is a module that allows the sending of an alarm due to the recognition of the described rule.

8 IMPLEMENTATION DETAILS

In this paper we discussed the theory concerning the recognition and management of the streaming stream. We conclude the analysis in this chapter with the discussion of implementation details related to the project itself. The topics covered will be: the description of the technologies and libraries used, the rules underlying the recognition of a particular state of the monitored patient and finally the description of the functions and the main steps in terms of code for the correct recognition of the scene.

8.1 Used Tools and Libraries

The Python programming language and IDE PyCharm were used to implement the project.

The choice of this language is surprising, because Python is a multi-paradigm language whose main objectives are dynamism, simplicity and flexibility, and it is related to a significant increase in productivity and the development of higher quality and maintainability code. The logical immediacy of its structures, which make it one of the most rapidly learning languages, has ensured that it is defined by some as a clear language not only for machines but also for humans. While PyCharm is an integrated development environment that allows for easy development in python. It allows the management of projects and is integrated with the main versioning tools What is particularly striking about PyCharm is the speed with which it allows text editing. Although based on a JVM it does not suffer from the annoying lag of the gui of which other IDEs like Eclipse and above all NetBeans can "boast". It allows you to debug applications in a very convenient way by providing a snapshot of the application at the current breakpoint and including all the information that of the python VM.

The libraries used are: boto3, OpenCV and timer.

Boto3 is the Amazon Web Services (AWS) SDK for Python. Allows Python developers to create, configure and manage AWS services, such as EC2 and S3. Boto offers an easy-to-use object-oriented API and low-level access to AWS services. Inside code boto3 is used to call up the Rekognition prediction service; the generated instance serves to be able then to recall facial recognition functions and obtain parameters to then apply the established rules.(Boto3 Documentation, 2019. https://boto3.amazonaws.com/v1/documentation/api/latest/index.html)

The OpenCV library plays another fundamental role in that it allows the management of streaming video streams, the division of the component into frames and their subsequent resizing.
(OpenCV Documentation : About, 2019, https://opencv.org/about/)

Finally there is timer, a library that allows you to act on time precisely, more specifically this library is used for the definition of a timer that at a predetermined interval calls a function dedicated to other controls.

8.2 Rules

After having gathered all that is necessary for the correct identification of a particular state of the patient, the basic rules describing the alteration of a patient's condition. There are two main aspects to take into consideration, namely the movement of the head and the facial expression of the person. These two associated factors make it possible to identify status changes.

The purpose of the system is to detect a sudden change from a calm state of the person which leads to a state of agitation.

The first factor, that is the movement, is fundamental, in fact there is the need to identify in a given period of time multiple quite anomalous movements, such as for example several clicks of the head. Clearly analyzing the movement and identifying sudden anomalies already allows us to have a starting point which, associated with the analysis of emotions, allows us to recognize the change of state. The second factor is therefore emotion. The latter is fundamental because according to the established rules, if sudden movements occur then the facial expression is analyzed. Clearly the emotion must be negative, such as: sadness, disgust, fear. Finally, once a negative facial expression has been identified and associated with the movement, there will be the final composition that forms the rule that establishes the change of state.

8.3 Code

In this paragraph we go into the details of the implementation, analyzing the individual functions and controls that allow us to have a fairly precise scene recognition system.

8.3.1 Configuration File

In the phase before the execution the user must modify a configuration file that allows to specify all the functions that will be made available by the system during execution. The format of the configuration file is JSON type, the choice of using this is obvious (JSON is the most used format for information exchange), clearly the user will have to access it and will be able to specify both the various tools to be used and the set of features made available for that particular tool. For example, to use the status analysis function of a patient, specify in the configuration file the name of the tool (in this case AWS Rekognition) and then in the actions to be performed the name of the specific function.
Subsequently, when the startup script is executed, it will read and analyze the configuration file to load all the necessary modules for the actions with a relative interface for the user's choice.

8.3.2 Start Window and Main Panel

Then, after changing the configuration file, the user can run the initial script called Start.py.

When the latter is started, the main part of the script calls a function that loads the configuration file in JSON format and controls all the functions specified by the user (it is clear that if there are errors in the file no module will be loaded). After analyzing the specified functions the script start.py launches another script related to the specified actions configuration file. The script launched is called functionPanel.py.

The functionPanel.py script takes care of building a small frame in which there are buttons, which specify the action to be performed. The Tkinter library is used in the code which allows the creation of a very simple graphic interface.

If the user interacts with the interface and clicks on the button related to an action then the script functionPanel.py launches the execution of the script StatusAnalysis.py which allows the loading of the detection modules.

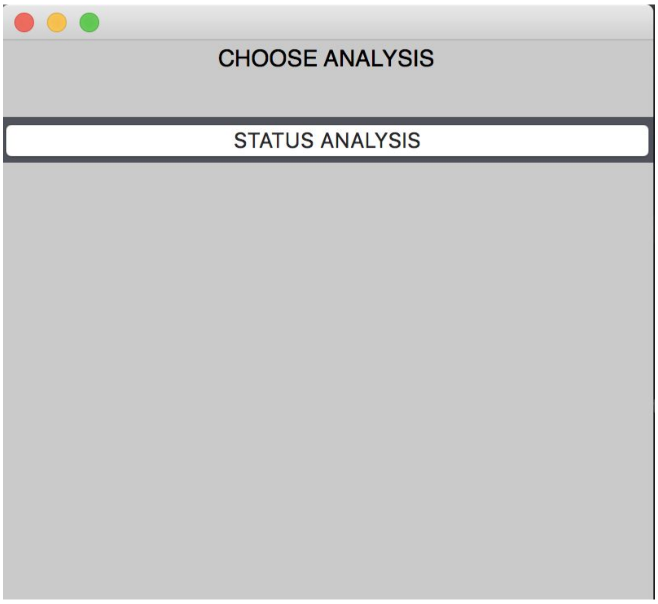

Fig.3 Action Panel

8.3.3 Detect Face Movement and Face Emotion

Going into the specific action of recognizing the state of the patient. After selecting the user through the interface, the script launches the detection process called StatusAnalysis.py. The latter is the core of the execution of the action as it starts both the motion detection module and the negative emotion detection module. Simultaneously with the start-up of the modules, it starts a 40-second timer in which it waits for the result of the two detection modules to carry out the final checks (specified in the following paragraph).

The first step in terms of code is to get the video stream from a real time device, in this project a simple webcam with a not too high definition was used.
Then the OpenCV library is used, more specifically the VideoCapture function with parameter 0; this function allows to obtain a real time video stream and based on the parameters it is possible to decide which device to take the data from. The parameter 0 specifies to take the stream from a webcam, clearly it is possible to replace this value with a URL of a live streaming such as in concrete terms the URL of the surveillance cameras or even a smartphone in live streaming.
Another function used is read, it is used for a first reading of the stream and to divide what has been read into a frame. Subsequently after the division in frame is called the resize function, very important, as it allows to resize the frames taken from the stream and then insert them into the buffer. After the resizing operation there is the call to the imencode function which allows to establish the frame format, in our case .jpg and the size.
Finally the imencode function always defines a buffer and converts the resized frame into a series of bytes to be inserted inside it, this because the images will then be read of the buffer by Rekognition which will deal with the predictions. A final function used is imshow , which allows to display a small window in the program run where

the live video stream will be inserted with the boxes related to the face to be analyzed.

The next step is to work on identifying the movement in the head and analyzing the subject's emotions. The first function called is detect_face, which has as its parameters the frames encoded in byte strings inside the buffer generated by the imencode function.

After the detect_faces call, a function provided by the Amazon Web Services API, the output will be precisely the identification of a face within the frame with many associated parameters that will be essential for the application of the basic rules of status change. In fact, after executing the recognition function, the first step is to obtain the bounding box that identifies the position of the face and its size within the frame.

Before going into details about the bounding box, it is necessary to specify that the system is designed for the identification of a single face and clearly the identification function provided by the API Rekognition identifies more than one face present within the scene. Therefore it is taken for granted that within the analysis scenario there is always only one individual. Clearly more people within the same scene would induce the system to generate anomalous results totally distant from reality. Therefore, the next step consists in obtaining information on the position and size of the head. The detect_faces function returns an array with multiple elements, through a loop it will be analyzed element by element (it is taken for granted that there will be only one element within the array since as already mentioned previously inside the frame there is only one individual, ie the patient).

All the coordinates of the bounding box are taken from the analyzed element, ie: left, top, height, width. After a careful analysis it was understood that to carry out checks on the bounding box a lot of time would be lost so the final solution was to define the center of the bounding box. This is because having two value for the center greatly simplifies movement-related controls through frame-by-frame analysis. The center of the bounding box is calculated using the following formulas:

$$centerX = (x + (width / 2))$$

$$centerY = (y + (height / 2))$$

Another important parameter for performing a complete analysis is the emotion defined by the patient's face. This last parameter is also provided by the detection function. The emotions identified are associated with a confidence percentage; it is obvious that with a low percentage of confidence the emotion must not be taken into consideration. After a careful study it was possible to notice that an emotion with a confidence greater than 56% had a correct response with the real expression of the analyzed face. Specifically, the emotions involved in the analysis are: sad, angry, confused, fear.

As mentioned before, the two detection modules launch a 20-second timer to determine then at its end whether fundamental changes have been detected for the analysis. If this happens then the two detection modules end otherwise another 20 second detection phase is launched.

8.3.3 Final Check

The last step is to understand if the two detection modules have detected the changes necessary to be able to detect a different status that can be reported. The last check is done in the script StatusAnalysis.py (the script that launches the detection modules). During the start-up of the modules, the latter are launched thanks to the use of two parallel processes. After launching the processes the main script starts a timer with a duration of 40 seconds. Knowing that the two modules end if they have detected changes in movement and emotions, then the last control is reduced to the analysis of a Boolean variable. In fact, if the two processes end in the analysis script, a variable called check is set to 1, which is checked at the end of the interval set by the timer. Clearly if the Boolean variable is equal to 1 after the interval of 40 seconds then it means that the two detection modules are terminated and the system can then launch a message on the console that signals the successful detection of the change of status of a patient .

8.4 Frame Example

In this paragraph, a typical scenario will be described in a visual way in which there is the monitoring of the patient with the detection of a change of state.

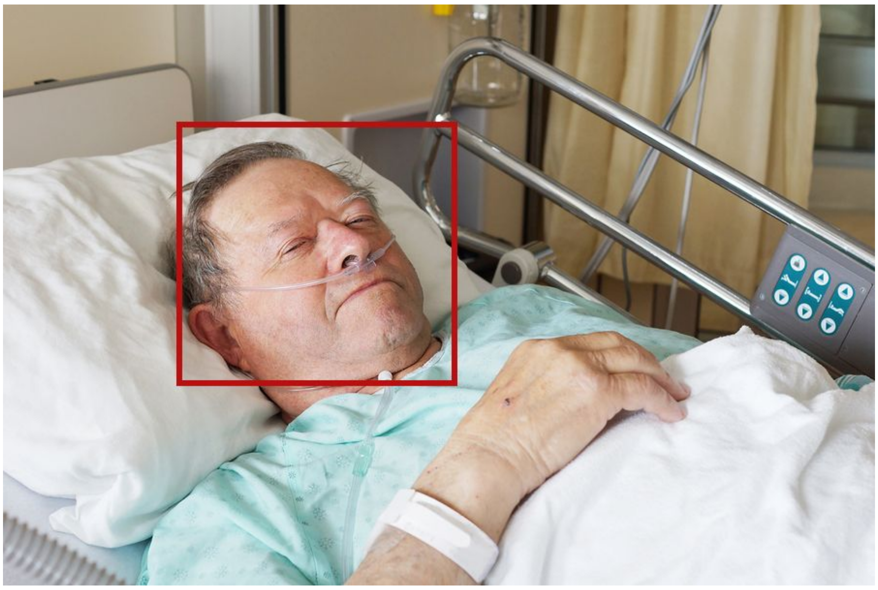

Fig.4 Patient Calm Status

Figure 4 describes the initial situation of the scenario. As described in detail in the previous chapters, there is the ECR Patient who is in an initial state of calm (as she is at rest). Also within the scenario, another patient property is described: the position in the analysis space, identified by the system developed through the graphic use of bounding boxes. Furthermore, the system also identifies the patient's state of calm through facial recognition, thus identifying the subject's entry conditions within the scene.

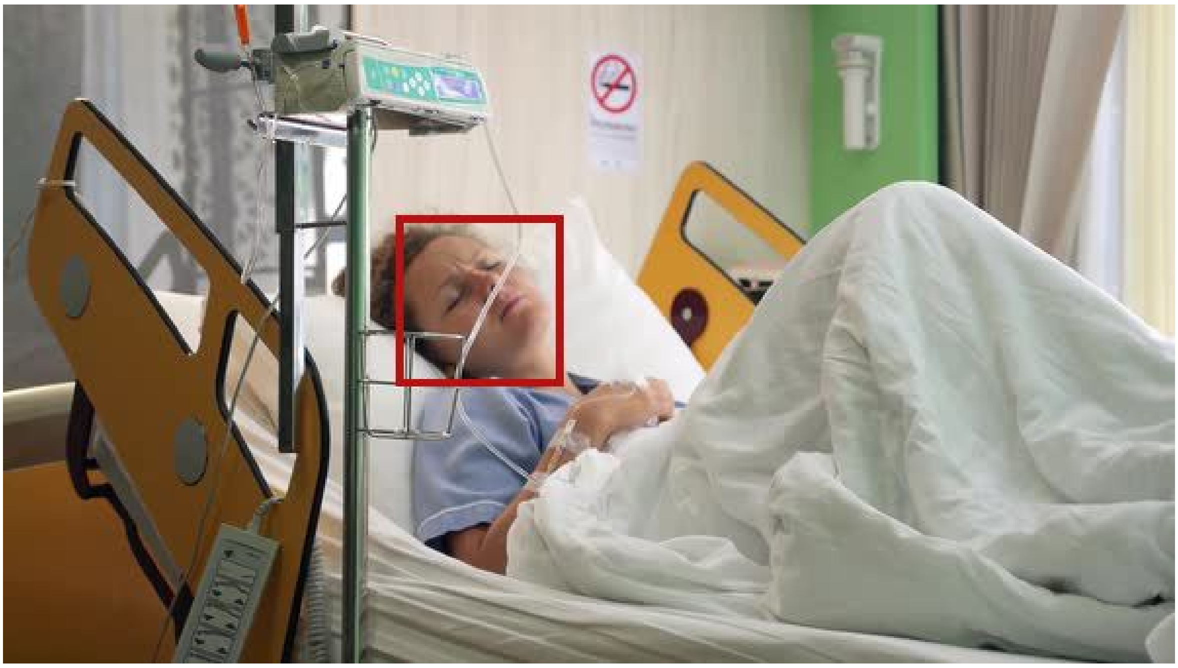

Fig.5 Patient Change Status

The objective of the system is to detect changes of state and in this figure we have an example of change. After an initial state of calm, the system monitors the subject concerned at regular time intervals. When an irregular and sudden movement is detected, the emotions are also checked. The system detects in sudden motion through the change of the position of the bounding boxes, speaking of architecture we can describe the detection of these two changes through the Action Representation (AR): CHANGE_POS and CHANGE_EMOTION.

The final check detects whether these two ARs are true. In the positive case another AR called CHANGE_STATUS is used, which in fact modifies the initial (calm) state of the patient in a state of agitation.

9 CONCLUSIONS

The implemented system has been carefully analyzed with the aim of being able to identify both negative and positive sides. First of all, one of the positive aspects is the scalability of the system, in fact thanks to the division of the detection functions and of the actions in modules it will be possible for future developments to add some other typology of analysis that can join with what has been implemented and therefore re-use the modules already developed.

Clearly there are many things to correct within the system. First of all, optimize the parallel management of the startup processes of the modules intended for detection. Furthermore it would be necessary to analyze the data coming from the video stream sources in order to make the final analysis of the chosen actions more accurate and the detection faster.

Finally, the system module that deals with changing the status of the patient should be tested on real cases, ie on streams of cameras specifically placed near the patient's face in a hospital room, so that all variables can be considered real environment and possibly make the necessary changes to better the system more and more.

BIBLIOGRAPHY

[1] Darren Eng, YOLO: Real Time Object Detection,2016. https://github.com/pjreddie/darknet/wiki/YOLO:-Real-Time-Object-Detection

[2] Kairos, Recognize People The Way You Want,2019. https://www.kairos.com

[3] Google Cloud AutoML:custom machine learning models,2019. https://cloud.google.com/automl/?hl=it

[4] Caruccio L., Polese G., Tortora G.,Iannone D. , A Knowledge Representation Framework to Enhance Automatic Video Surveillance, Department of Computer Science, University of Salerno.

[5] Margaret Rouse,What is internet of things,2016. ,https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

[6] Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. Artificial Intelligence in Design ,1996. Springer, Dordrecht. pp. 151–170

[7] Gilman, Lynn. "The Theory of Multiple Intelligences",2012. Indiana University

[8] Frost&Sullivan, Artificial Intelligence - The Cognitive Area,2019. https://ww2.frost.com/research/visionary-innovation/artificial-intelligence-cognitive-era

[9] Amazon Rekognition, Intelligent Analysis of Images and Videos for applications, 2019. https://aws.amazon.com/it/rekognition/

[10] Ronan Collobert e Jason Weston, A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning, in Proceedings of the 25th International Conference on Machine Learning, New York, NY, USA, ACM, 2008.

[11] Joshi, K. A., & Thakore, D. G. (2012). A survey on moving object detection and tracking in video surveillance system. International Journal of Soft Computing and Engineering, 2, 44–48.

[12] Boto3 Documentation, 2019. https://boto3.amazonaws.com/v1/documentation/api/latest/index.html)

[13] OpenCV Documentation : About, 2019, https://opencv.org/about/